Studying the Clustering Paradox and Scalability of Search in Highly Distributed Environments

Article *in* ACM Transactions on Information Systems · April 2013

DOI: 10.1145/2457465.2457468

Weimao Ke
Drexel University

Studying the Clustering Paradox and Scalability of Search in Highly Distributed Environments

WEIMAO KE, Drexel University
JAVED MOSTAFA, University of North Carolina at Chapel Hill

With the ubiquitous production, distribution and consumption of information, today's digital environments such as the Web are increasingly large and decentralized. It is hardly possible to obtain central control over information collections and systems in these environments. Searching for information in these information spaces has brought about problems beyond traditional boundaries of information retrieval (IR) research. This article addresses one important aspect of scalability challenges facing information retrieval models and investigates a decentralized, organic view of information systems pertaining to search in large-scale networks. Drawing on observations from earlier studies, we conduct a series of experiments on decentralized searches in large-scale networked information spaces. Results show that how distributed systems interconnect is crucial to retrieval performance and scalability of searching. Particularly, in various experimental settings and retrieval tasks, we find a consistent phenomenon, namely the *Clustering Paradox*, in which the level of network clustering (semantic overlay) imposes a scalability limit. Scalable searches are well supported by a specific, balanced level of network clustering emerging from local system interconnectivity. Departure from that level, either stronger or weaker clustering, leads to search performance degradation, which is dramatic in large-scale networks.

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

General Terms: Algorithms, Performance, Theory, Experimentation

Additional Key Words and Phrases: information network, information retrieval, large-scale distributed system, decentralized search, self-organization, loose coupling, network clustering, efficiency, scalability

ACM Reference Format:
Weimao Ke and Javed Mostafa, 2013. Studying the Clustering Paradox and Scalability of Search in Highly Distributed Environments. *ACM Trans. Inf. Syst.* 31, 2, Article 3 (April 2013), 37 pages.
DOI:http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

The growing magnitude, dynamics, and heterogeneity of today's digital environments such as the Web pose great challenges for finding information in them. While classic information retrieval systems conduct search operations by collecting and indexing information in advance, this centralized model has suffered from the increasing decentralization of information and systems [Baeza-Yates et al. 2007].

This work was supported by Weimao Ke's startup fund in the College of Information Science and Technology at Drexel University. The research was partially supported through a grant from the National Center for Research Resources, Grant UL1RR025747, and is now at the National Center for Advancing Translational Sciences, Grant UL1TR000083.

Authors' addresses: W. Ke, College of Information Science and Technology, Drexel University, Philadelphia, PA 19104, U.S.A., email: wk@drexel.edu; J. Mostafa, School of Information and Library Science, University of North Carolina, Chapel Hill, NC 27599, U.S.A., email: jm@unc.edu.

DOI:http://dx.doi.org/10.1145/0000000.0000000

We live in a distributed networked space, where information and intelligence are highly distributed. In reality, people have different expertise, share information with one another, and ask trusted peers for advice/opinions on various issues. The World Wide Web is a good example of information distribution, where web sites serve narrow information topics and tend to form communities through hyperlink connections [Gibson et al. 1998; Flake et al. 2002; Menczer 2004]. Likewise, individual digital libraries maintain independent document collections and none claims to be all encompassing or comprehensive. There is no single global information repository.

Because of the distributed nature of information and its size, dynamics, and heterogeneity, it is extremely challenging, if not impossible, to collect, store, and process all information in one place for retrieval purposes. Centralized solutions will suffer from its vulnerability to scalability demands [Baeza-Yates et al. 2007]. It has become critical to investigate alternative models beyond the state-of-the-art retrieval systems, particularly for searching a large, highly decentralized environment such as the Web. A potential candidate is to take advantage of existing distributed computing powers and design a new search architecture in which all systems can participate to help one another find information.

This research studies the general problem of search and retrieval in a fully distributed/decentralized environment, where no global information is available nor can a centralized index be built. Specifically, it focuses on query routing for decentralized search and addresses the scalability challenge by integrating perspectives from information retrieval as well as complex network research. One aim is to understand the influence of system interconnectivity and the emergent network structure on decentralized search performance, which is crucial to the design of network overlays for scalable IR operations in these environments.

2. PROBLEM STATEMENT

It is a great challenge to perform effective and efficient retrieval operations in large, dynamic, and heterogeneous information networks. Collection of information in advance and centralization of IR operations are hardly possible because systems are dynamic (e.g., in the deep web) and information is distributed. A fully distributed architecture is desirable and, due to many additional constraints, is sometimes the only choice. What is potentially useful in such an information space is that individual systems (e.g., peers, sites, or agents) are connected to one another and collectively form some structure (e.g., the Web graph of hyperlinks, peer-to-peer networks, and interconnected services and agents in the Semantic Web). Many of these structures, according to studies on complex networks, are small world networks in which there is a small degree of separation between any two systems/nodes [Albert et al. 1999; Albert and Barabási 2002; Barabási 2009].

While an information need may arise from anywhere in the space – for example, a query can be issued to a Web system, raised by an information agent, or sent from a connected peer – relevant information may exist in certain segments. There requires a mechanism to help the two – the query and the resource – meet each other by either delivering relevant information to the one who needs it or routing a query (representative of the need) where information can be retrieved. Potentially, intelligent algorithms can be designed to help one traverse a *short path* to another in the networked space.

One might question why there has to be so much trouble to find information through a network. A simple solution would be to connect a system to all other systems and choose the relevant system from a full list. However, no one can manage to have a complete list of all others and afford to maintain the list given the size of such a space. The Web, for example, has more than hundreds of millions of sites and trillions of documents. It is hardly possible to implement and maintain a system that integrates all.

2.1. Research Questions

Now let's review the problem in its basic form. Let $G(A, E)$ denote the graph of a networked space, in which A is the set of all $agents$[1] (nodes, sites, or peers) and E is the set of all edges or connections among the agents. Agents have individual information collections, know how to communicate with their direct (connected) neighbors, and are willing to share information with them. Some agents' information collections are partially known. Many agents, given their dynamic nature, only provide some information when properly queried – that their information cannot be collected in advance without receiving a properly formulated query.

Depending on task contexts, agents may represent information seekers as well as providers and mediators. Imagine an agent in the network, say, A_u, receives an information request, i.e., a query representation of an information need. Suppose another agent A_v, somewhere in the network, has relevant information for the need. Assume that A_u is not directly connected to and might not even know the existence of A_v. However, we reasonably assume that the network is a small world and there are short paths from A_u to A_v. Now the basic question is:

PROBLEM 1. *Can agents that are directly and/or indirectly known (connected) to A_u help identify A_v such that A_u's query can be submitted to A_v who in turn provides relevant information back to A_u?*

One can reasonably propose a simple solution to the problem above through flooding or breadth first search. However, flooding may achieve retrieval effectiveness at the cost of coverage – it will reach a significant proportion of all agents in the network for a single query. This type of solutions will not scale. A constraint here is that network traffics should be minimized for each query. We should therefore consider efficiency:

PROBLEM 2. *Efficiency: Given A_v is findable for A_u in a network, can the number of agents involved in the search process be relatively small compared to the network size so that each query only engages a very small part of the network?*

More critically, the question about search scalability should be asked as well:

PROBLEM 3. *Scalability: Can the number of agents involved in each query remain small (on a relatively constant scale) regardless of the scale of network size? And how?*

3. RELATED WORK

Related challenges for distributed search have been studied in areas of distributed (federated) information retrieval, peer-to-peer networks, multi-agent systems, and complex networks [Callan 2002; Crespo and Garcia-Molina 2005; Yu and Singh 2003; Kleinberg 2006b; Meng and Yu 2010]. Recent distributed IR research has focused on distributed database content (and characteristics) discovery [Si and Callan 2003a], database selection [French et al. 1998; French et al. 1999; Hawking and Thomas 2005; Shokouhi and Zobel 2007], and result fusion [Aslam and Montague 2001; Baumgarten 2000; Manmatha et al. 2001; Si and Callan 2005; Lillis et al. 2006]. Research has studied the effectiveness of database selection and result fusion given a relatively small number of distributed, persistent information collections [Meng et al. 2002; Shokouhi

[1] An $agent$ can be seen as a participating distributed system, which provides information and/or serves as an intermediary.

and Si 2011]. Their scalability to larger, unstable environments remains an important question.

A peer-to-peer network often involves more than thousands, sometimes millions, of distributed peers who dynamically join and leave the community. Distributed hashing tables (DHTs) have been used in structured P2P environments for unique identifier lookup. Some studies applied DHTs for partitioning an indexing space across redundant peers for efficient location of popular information items [Luu et al. 2006; Skobeltsyn et al. 2007]. Others proposed the use of this technique for search in the presence of information overlap [Bender et al. 2005].

While DHT-based techniques are applicable in structured P2P environments, their resilience to transient populations and adaptability to content and topology changes remain open questions [Lua et al. 2005]. More important, when diverse information needs are to be served, it is extremely challenging for such techniques to create and maintain (update) a distributed index structure in a space- and traffic-efficient manner. Flooding is often the technique employed for maintaining indexing currency, which has received critiques for its computational costs.

DHTs, based on document-key-level index partitioning, are not the ideal technique for information retrieval in distributed environments, particularly in unstructured peer-to-peer networks. Alternative methods based on peer-level segmentation can be used to support search efficiency and effectiveness [Bawa et al. 2003; Liu et al. 2006; Lu and Callan 2006]. Reorganization of collections around content clusters/topics was proposed to improve distributed retrieval effectiveness [Xu and Croft 1999].

Semantic overlay networks (SONs), based on peer segmentation, have been widely used for distributed IR operations in unstructured networks [Tang et al. 2003; Crespo and Garcia-Molina 2005; Cooper and Garcia-Molina 2005; Doulkeridis et al. 2008]. In SONs, peers with semantically similar content are clustered together, which in turn form a global (hierarchical) structure for efficient query routing [Crespo and Garcia-Molina 2005]. These techniques were shown in experiments to improve retrieval performance.

One popular approach to SONs was to form a hierarchical network structure through re-organization of distributed systems/peers, in which super-peers assumed greater responsibilities for bridging/mediating across segments. However, some questioned the reliability of such an architecture as attacks on super peers (nodes or agents) can lead to a large disconnected structure [Albert and Barabási 2002; Lua et al. 2005]. In addition, as Lu [2007a] observed, updating super-peers for changes in distributed collections is traffic intensive and may cause problems in environments where bandwidth is limited.

Regardless of various approaches to peer re-organization and segmentation, the underlying network structure appears to play an important role in conducting effective and efficient retrieval operations in distributed settings. As individual systems interconnect to form a global structure, finding relevant information in decentralized environments transforms into a problem concerning not only information retrieval but also complex networks. Understanding network structure or system interconnectivity will provide guidance on how decentralized search and retrieval methods can function in these information spaces.

In a variety of large interconnected environments, it is well known that any pair of individual nodes are separated by a very small number of others. In other words, small diameters are a common feature of many naturally, socially, or technically developed communities – a phenomenon known as *small world* or *six degrees of separation* [Milgram 1967; Watts 2003]. The small world phenomenon also appears in various types of large-scale digital information networks such as the World Wide Web [Albert et al. 1999; Albert and Barabási 2002] and the network for email communications [Dodds et al. 2003]. The small degree of separation shows promises on efficient traversal of such a network to reach any desired targets. Nonetheless, it remains challenging to identify shortcuts to *relevant* targets in the information retrieval context, where relevant as well as non-relevant information collections are all within short distances.

Network clustering represents one approach to understanding how network characteristics can be taken advantage of for efficient traversal of relevant paths. One level of clustering, in P2P research, is the identification of similar peers and segmentation of them based on topical relevance. As we discussed, semantic overlay networks (SONs) have been widely used for retrieval effectiveness and efficiency [Bawa et al. 2003; Crespo and Garcia-Molina 2005; Lu 2007b; Doulkeridis et al. 2008]. Clustering enables similar peers to connect to each other and sometimes allows super peers to coordinate local reconstruction and to update remote connections for efficient query routing. Query propagation in local segments often leads to improved recall [Bawa et al. 2003; Lu 2007b].

Research on complex networks studies the problem in its basic form and shows promises as well. Particularly, studies showed that, with local intelligence and basic information about targets, members of a very large network are able to find very short paths (if not the shortest) to destinations collectively [Milgram 1967; Kleinberg 2000; Watts et al. 2002; Dodds et al. 2003; Liben-Nowell et al. 2005; Boguñá et al. 2009]. The implication in IR is that relevant information, in various networked environments, is not only a few degrees (connections) away from the one who needs it but potentially findable. This provides the potential for distributed algorithms to traverse such a network to find relevant information efficiently.

In this respect, Kleinberg [2000] conducted one of the key studies on decentralized search in small world networks. The research, based on an abstract lattice model and a clustering exponent α to control network clustering, discovered that some critical value of α enables optimal search efficiency. Particularly, in a d dimensional space, search time (or the number of hops required to reach a target) is bounded by $c \log^2 N$ only when $\alpha = d$, where N is network size. When α becomes either larger or smaller, search performance is greatly degraded. In other words, neither weak clustering nor strong clustering is desirable. A specific, balanced level of clustering must be maintained for search efficiency. Related studies on complex network search provided results consistent to this finding [Watts et al. 2002; Dodds et al. 2003; Liben-Nowell et al. 2005; Kleinberg 2006a; Boguñá et al. 2009]. In distributed IR settings, nonetheless, this remains a phenomenon to be scrutinized and understood.

In short, network structure, i.e., how distributed system interconnect, matters to search efficiency and scalability. While structural properties such as *network clustering* are manifestations of underlying dynamics of a network, network formation can be guided for related properties to be optimized to improve retrieval efficiency. Research needs to understand the optimal network structure for search so that mechanisms such as *semantic overlying* can be better designed to obtain the desired structural properties. For a better understanding of the impact in the large-scale IR context, it requires in-depth experimental examination of various IR tasks, proper control and isolation of important variables, and further analysis of retrieval performance at a larger scale using evaluation methods pertinent to individual tasks. This article reports on important findings from a series of large-scale distributed IR experiments.

4. RESEARCH ANGLE AND HYPOTHESES

Finding relevant information in distributed environments is a problem concerning complex networks and information retrieval. We know from the small world phenomenon, common in many real networks, that every piece of information is within a short radius from any location in a network. However, relevant information is only a tiny fraction of all densely packed information in the “small world.”

If we allow queries to traverse the edges of a network to find relevant information, there has to be some association between the network space and the relevance space in order to orient searches. Random networks could never provide such guidance because edges are so independent of content that they have little semantic meaning. Fortunately, research has discovered that development of a wide range of networks follows not a random process but some preferential mechanism that captures “meanings.”

Surely, these networks, even with a good departure from randomness, do not automatically ensure efficient findability of relevant information. To optimize such a network for search, mechanisms should be designed to enable more meaningful semantic overlay on top of physical connections. In peer-to-peer information retrieval research, such techniques as semantic overlay networks have been widely used. A better understanding of the impact of structural properties on distributed search is critical to further development of these techniques.

4.1. Information Network and Semantic Overlay

Let us refer to the type of networks in this research as information networks to emphasize the focus on finding relevant information. Practically, information networks include, but are not limited to, peer-to-peer networks for information sharing, the Web where many sites/systems/databases reside, and networks formed by information agents. Close examination of these networks reveals some common characteristics illustrated in Figure 1.

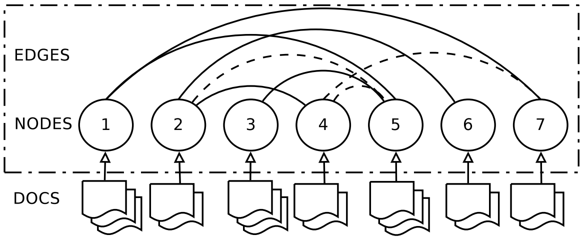
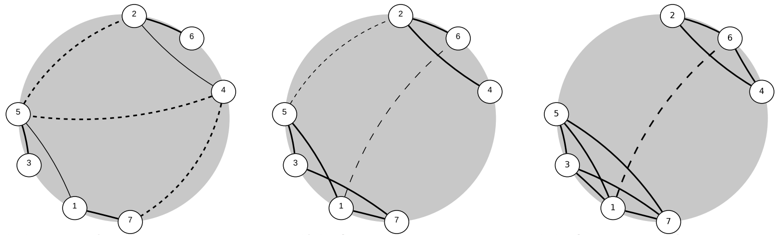


Fig. 1. Information Network

As shown in Figure 1, an information network is formed by nodes (e.g., peers, web sites, or agents) through edges, e.g., by means of network communication/interaction/links. A node has a set of information items or documents, which in turn can be used to define its topicality. If we can discover the content of each node and layout the nodes in terms of their topicality, then the information network in Figure 1 may be visualized in the form of Figure 2 (a).

Figure 2 (a) shows a circle representation of the topical (semantic) space, in which there are two topical clusters of nodes, i.e., cluster 1-3-5-7 and cluster 2-4-6 (visually separated on the topical circle space). Connection-wise, there are local edges (solid lines) within each cluster and long-range ones (dashed lines) between the clusters.

Within-group local connections are useful because they bring “close” (topically similar) nodes together to form segments, which is consistent to their topical separation. This establishes an important association between the topological (network) space and the topical (search) space that potentially guides searches. In terms of Granovetter [1973], these are *strong ties*.



(a) Weak clustering (b) Stronger. . . (c) Strong clustering

Fig. 2. Evolving Semantic Overlay

Long-distance connections, shown as dashed lines in Figure 2, bring randomness to the network. When there are many long-distance connections, the topological (network) space tells little about the topical (search) space – we can hardly rely on topically non-relevant edges in the search for topical relevance. Nonetheless, between-group connections, or weak ties, often serve as bridges and are critical for efficient diffusion of information [Granovetter 1973].

While the initial network, shown in Figure 2 (a), might not be good enough for decentralized search, some overlay can be built upon the physical layer to bring more semantics to the network space. Due to no global control over such an information network, mechanisms should be designed to guide individual adaptation and network evolution for this purpose. Over the course of network development shown in Figures 2 (a), (b), and (c), semantic overlay is strengthened through the reinforcement of strong ties and reestablishment of some weak ties. Note that semantic overlay is a logical (soft) layer of interconnectivity – even if two nodes are physically connected, semantic overlay may maintain a probability function that keeps them from contacting each other for search.

4.2. Clustering Paradox

Here we elaborate on the *Clustering Paradox* mentioned earlier. Semantic overlay illustrated above is essentially a type of *clustering*, which is the process of bringing similar items together [Berry 2004]. Research has found *clustering* at various levels useful for information retrieval. The *Cluster Hypothesis* states that relevant documents are more similar to one another than to non-relevant documents and therefore closely related documents tend to be relevant to the same requests [van Rijsbergen and Sparck-Jones 1973]. Traditional IR research utilized document-level clustering to support exploratory searching and to improve retrieval effectiveness [Hearst and Pedersen 1996; Fischer and Nurzenski 2005; Ke et al. 2009].

Distributed information retrieval, particularly unstructured peer-to-peer IR, relied on peer-level clustering for better decentralized search efficiency. Topical segmentation based techniques such as semantic overlay networks (SONs) have been widely used for efficient query propagation and high recall [Bawa et al. 2003; Crespo and Garcia-Molina 2005; Lu and Callan 2006; Doulkeridis et al. 2008]. Hence, overall, clustering was often regarded as beneficial whereas the potential *negative* impact of clustering (or over-clustering) on retrieval has often been overlooked.

Research on complex networks has found that a proper level of network clustering with some presence of remote connections has to be maintained for efficient searches [Kleinberg 2000; Watts et al. 2002; Liben-Nowell et al. 2005; Simsek and Jensen 2008; Boguñá et al. 2009]. Clustering reduces the number of “irrelevant” links and aids in creating topical segments useful for orienting searches. Without sufficient clustering, the network has too much randomness to guide efficient traversals because *weak ties* dominate. While searches may jump quickly from one place to another (hops) in the network space, there is no "gradient" to lead them toward targets. With very strong clustering, on the other hand, a network tends to be fragmented into local communities with abundant *strong ties* but few *weak ties* to bridge remote parts [Granovetter 1973; Singh et al. 2001]. Although searches might be able to move gradually toward targets, necessary "hops" become unavailable.

As discussed earlier, we refer to this phenomenon as the *Clustering Paradox*, in which neither strong clustering nor weak clustering is desirable. In other words, trade-off is required between *strong ties* for search orientation and *weak ties* for efficient traversal. In Granovetter's terms, whereas *strong ties* deal with local connections within small, well-defined groups, *weak ties* capture between-group relations and serve as bridges of social segments [Granovetter 1973]. The *Clustering Paradox*, seen in light of strong ties and weak ties, has received attention in complex network research but requires close scrutiny in a decentralized IR context.

4.3. Function of Clustering Exponent α

One key parameter widely used in complex network research for studying the impact of clustering is the *clustering exponent* α. Kleinberg [2000] studied decentralized search in small world using a two dimensional model, in which peers had rich connections with immediate neighbors and sparse associations with remote ones. The probability p_r of connecting to a neighbor beyond the immediate neighborhood was proportional to $r^{-\alpha}$, where r was the search distance between the two in the dimensional space and α a constant called *clustering exponent*[2]. It was shown that only when *clustering exponent* $\alpha = 2$, search time (i.e., search path length) was optimal and bounded by $c(\log N)^2$, where N was the network size and c was some constant [Kleinberg 2006b].

The *clustering exponent* α, as shown in Figure 3, describes a correlation between the network (topological) space and the search (topical) space [Kleinberg 2000; Boguñá et al. 2009]. When α is small, interconnectivity has little dependence on topical closeness – local segments become less visible as the network is built on increased randomness. As shown in Figure 4 (a), 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 [Granovetter 1973]. The network becomes highly segmented. As shown in Figure 4 (c), when $\alpha \to \infty$, 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 4 (b), the network becomes a narrowly defined *small world*, in which both local and remote connections are present.

In this way, the *clustering exponent* α influences the formation of local clusters and overall network clustering. The impact of $\alpha \in [0, \infty)$ on network clustering is similar to that of a rewiring probability $p \in [1, 0]$ in Watts and Strogatz [1998]. However, α additionally defines the association of interconnectivity and topical distance. It was further discovered that optimal value of α for search, in many synthetic networks previously studied, depends on the dimensionality of the search space. Specifically, when $\alpha = d$ on a d-dimension space, decentralized search is optimal. Further studies conducted by various research groups have shown consistent results [Watts et al. 2002; Liben-Nowell et al. 2005; Simsek and Jensen 2008; Boguñá et al. 2009]. The results were primarily obtained in research on low dimensional synthetic spaces using highly abstract models. Its implications in IR settings have yet to be scrutinized.

[2] The *clustering exponent* α is also known as the *homophily exponent* [Watts et al. 2002; Simsek and Jensen 2008].

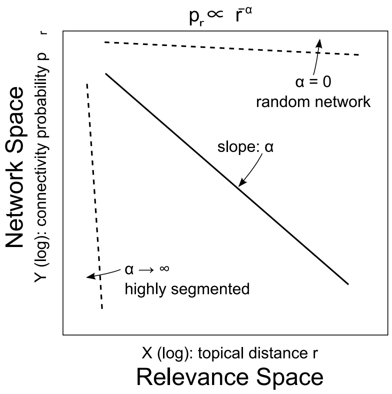

$p_r \propto r^{-\alpha}$
Network Space
Y (log): connectivity probability p_r
α = 0
random network
slope: α
α → ∞
highly segmented
X (log): topical distance r
Relevance Space

Fig. 3. Network Clustering: Function of Clustering Exponent α

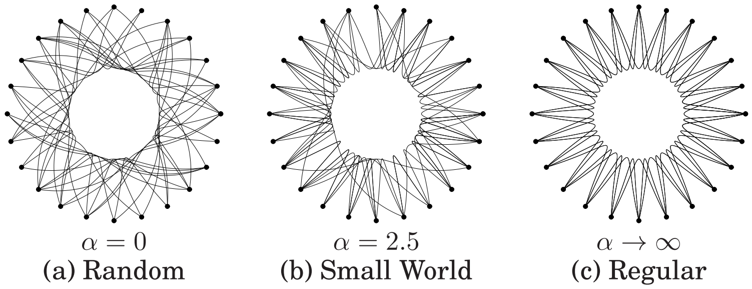

$\alpha = 0$
(a) Random
$\alpha = 2.5$
(b) Small World
$\alpha \to \infty$
(c) Regular

Fig. 4. Network Clustering: Impact of Clustering Exponent α. Compare to Watts and Strogatz [1998]. (a) a random network, provided no association between interconnectivity and topical distance at $\alpha = 0$, (b) a small world network when a moderate α value allows the presence of both local and remote connections, and (c) a regular network where nodes only connect to local neighbors at $\alpha \to \infty$ (simulated given $\alpha = 1000$). The figures were produced by simulations based on $n = 24$ nodes and $k = 4$ neighbors for each. Topical distance is measured by the angel between two nodes (vectors from the origin/center) in the 1-sphere (circle) representation.

5. HYPOTHESES

Earlier discussions provide evidence for potential hypotheses. In sections 4.2 and 4.3, we discussed previous research on the impact of network clustering on decentralized search and our observation of the *Clustering Paradox*, which appears to suggest the following hypothesis.

HYPOTHESIS 1. *Given local constraints*[3] *of a network, there exists some balance of network clustering that enables optimal search performance in an IR context.*

Given the balance or optimization, we further conjecture that some local search algorithm without global information is scalable to very large network sizes. In other words, search performance should remain more or less stable (with no dramatic change) even when the network grows dramatically. This leads to the second hypothesis.

[3] Local constraints refer to limited capacities of individual agents/peers, e.g., the number of connections an agent can manage.

HYPOTHESIS 2. *With optimal network clustering, search efficiency*[4] *is explained by a poly-logarithmic function of network size.*

We have known that scale-free properties such as power-law degree distribution appear in many real networks, in which research has found good scalability and robustness [Albert and Barabási 2002]. Although degree distribution may interact with network clustering on search performance, we tend to believe that such networks, regardless of their differences, support scalable decentralized search operations. In other words,

HYPOTHESIS 3. *Power-law degree distributions have an impact on network optimization for search – that is, different distributions may require different network clustering levels for optimal search. However, Hypotheses 1 and 2 remain true with different degree distributions.*

While most search methods rely on topical relevance, research has also found degree-based methods effective in power-law networks in which hubs have rich connectivity [Adamic et al. 2001; Boguñá et al. 2009]. We therefore conjecture that:

HYPOTHESIS 4. *In large-scale information networks, search (neighbor selection) methods that utilize information about neighbors' degrees and relevance (similarity to a query) are among scalable algorithms stated in Hypotheses 1 and 2.*

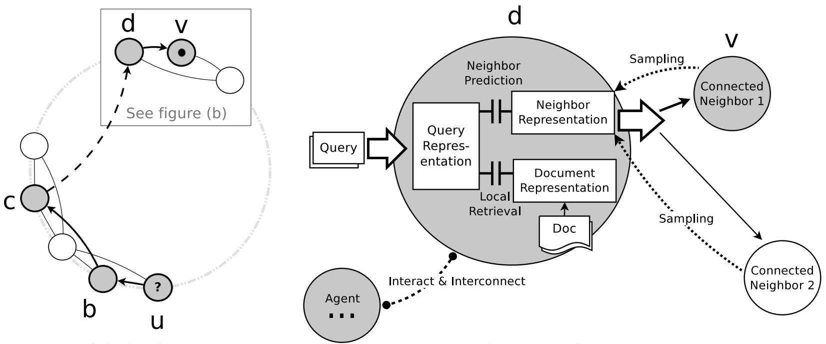
6. SIMULATION FRAMEWORK OVERVIEW

In this research, we proposed to use multi-agent systems for a bottom-up investigation of decentralized IR functions. According to [Jennings and Wooldridge 1998; Huhns 1998], an *agent* is an active computational entity situated in some environment, and that is capable of *autonomous action* in this environment in order to meet its design objectives. While single-agent systems focus on the individual agent as the functional unit, multi-agent systems emphasize the societal view of agents and their collective capability. Multi-agent systems provide a new paradigm in which a complex system – a decentralized information retrieval system in this research – can be naturally decomposed into autonomous, heterogeneous, and cooperative components to cope with the complexity and unpredictability of the environment [Jennings 2001].

Based on multi-agent systems, we have developed a decentralized search platform named *TranSeen* for finding relevant information distributed in networked environments. Each agent represents an IR system, which has its document collection and can connect to others to route queries. We emphasize the societal view of agents who have local intelligence and can collaborate with one another to perform global search tasks. We illustrate the conceptual model in Figure 5 (a) and elaborate on major components shown in Figure 5 (b).

Figure 5 (a) visualizes a 2D circle (1-sphere) representation of the information space. Let agent A_u be the one who has an information need 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. When an agent receives a query, it first runs a local search operation to identify potential relevant information from its individual document collection. If local results are unsatisfactory, the agent will send the query to neighbors based on a predicting function using the query representation and information about neighbors. In Figure 5 (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

[4] Efficiency, or search time, will be measured by search path length in tasks performed by best search algorithms.



(a) Global View (b) Local View

Fig. 5. Conceptual Framework. (a) Global view of agents work together to route a query in the network space. (b) Local view of how components function within an agent's neighborhood.

a remote connection (weak tie) for the query to "jump." The entire network topology is self-organized by agents using an interconnectivity probability function supervised by clustering exponent α, which we discuss in Section 7.

7. ALGORITHMS

In the previous section, we described the *TranSeen* multi-agent framework for decentralized information retrieval experiments. Figure 5 (b) illustrates how various components work together within each agent or system. The TranSeen system was implemented 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 [Bellifemine et al. 2007], and 2) Lucene, a high-performance library for full-text search [Hatcher et al. 2010].

This section elaborates on specific algorithms implemented in the TranSeen framework and used in the research. Section 7.1.1 presents the *TF*IDF* weighting scheme for information representation (to represent documents and queries) while section 7.1.2 discusses a similar method we refer to as *DF*INF* for neighbor (agent) representation. Section 7.1.3 discusses the coefficient for measuring the similarity of two information items. Section 7.2 describes five search (neighbor selection) algorithms based on neighbor relevance (similarity) and/or connectivity. Section 7.3 elaborates on the function for agent interconnectivity (clustering) based on *clustering exponent* α and *degree exponent* γ.

7.1. Basic Functions

*7.1.1. TF*IDF Information Representation.* We use the Vector-Space Model (VSM) for information (document and query) representation [Baeza-Yates and Ribeiro-Neto 2004]. Given that information is highly distributed, a global term space is not assumed. Instead, each agent processes information it individually has and produces a local term space, which is used to represent each information item using the TF*IDF (Term Frequency * Inverse Document Frequency) weighting scheme. An information item is then converted to a numerical vector where a term 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 are computed within the information space of an agent given no global information.

*7.1.2. DF*INF Agent Representation.* For neighbor (agent) representation, we use a similar mechanism. Specifically, we assume agents are able to collect their direct neighbors' document frequency (DF) information and use it to represent each neighbor as a meta-document of terms. Distributed IR research has shown DF information useful for collection selection [Callan et al. 1995; Callan and Connell 2001; Powell and French 2003]. Treating each meta-document as a normal document, it becomes straightforward to calculate *neighbor frequency* (NF) values of terms, i.e., the number of meta-documents (neighbors) that contains a particular term. A meta-document (neighbor) is then represented as a vector where term t is computed by:

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

where $df'(t)$ is the frequency of the term t of the term space in the meta-document, N' is the total number of an agent's neighbors (meta-documents), 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.

7.1.3. Similarity Scoring Function. Based on the term vectors produced by the ***TF*IDF*** (or ***DF*INF***) representation scheme described above, pair-wise similarity values can be computed. Given a query q, the similarity score of a document d matching the query is computed by :

$$\sum_{t \in q} W(t) \cdot coord(q, d) \cdot queryNorm(q) \tag{3}$$

where $W(t)$ is the weight of term t given by equation 1 or 2, $coord(q, d)$ 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 adopted in Lucene [Baeza-Yates and Ribeiro-Neto 2004; Hatcher et al. 2010]. Given a query, an agent will use this scoring function to rank its local documents and determine whether it has relevant information. In addition, when an agent has to contact a neighbor for the query, similarity-based neighbor selection methods will use this to evaluate how similar/relevant a neighbor is to a query.

7.1.4. Retrieval Federation/Fusion Method. In some of the search tasks (e.g., Relevance Search and Authority Search tasks described in Section 8.3), search results will contain a rank list of *relevant* documents from multiple distributed systems. Result fusion/federation has been an important research topic in distributed IR. Drawing on basic ideas from classic federation models such as CORI and GlOSS [Gravano et al. 1994; Callan et al. 1995; French et al. 1999], the following method is used in our experiments.

First, when a search is finished (i.e., a query finishes traversing a network for relevant documents), the method will select top n_s (5 in the experiments) systems whose meta-documents are most relevant/similar to the query (based on the DF*INF and similarity scoring functions described above). Each of the selected systems is queried again to provide a list of top n_d (e.g., 20) most relevant documents. Given similarity score S_d of document d from a system with a meta-document similarity score S_m, the document's similarity score is then normalized to:

$$S'_d = S_d \cdot S_m \tag{4}$$

All the $n_s \cdot n_d$ documents are sorted in terms of their normalized scores S'_d. Only top n_T (a predefined parameter in each experiment, 10 in relevance and authority searches, and 3 in TREC web track tasks) documents will be retrieved as search results. Results will then be evaluated using normalized discounted cumulative gain (nDCG) at position n_T described in section 8.5 [Jarvelin and Kekalainen 2002].

7.2. Neighbor Selection Strategies (Search Algorithms)

The similarity scoring function in Equation 3 can produce output about each neighbor's similarity/relevance to a query. Based on this output, we further propose the following strategies to decide which neighbors should be contacted for the query. Each search will keep track of all agents on the search path. All strategies below will ignore neighbors who have been contacted for a query to avoid loops. These strategies will be tested and compared in experiments.

7.2.1. Random Walk (RW): Effectiveness Lower-bound. 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* will serve as the search performance lower-bound.

7.2.2. SIM Search: Similarity-based Greedy Routing. Let k be the number of neighbors an agent has and $S = [s_1, .., s_k]$ be the similarity vector about each neighbor's similarity/relevance to a query. The ***SIM*** method sorts the vector and forwards the query to the neighbor with the highest score. With greedy routing, only one instance of the query will be forwarded from one agent to another until relevant information is found or some predefined conditions are met (e.g., the maximum search path length or Time to Live (TTL) is reached).

To obtain the similarity vector given a query, neighbors should be represented to reflect document collections they have. Query-based sampling techniques can be used to obtain this information. In order to simplify the process and focus on major retrieval challenges, we assume that agents are cooperative – that is, they share with one another document frequency (DF) values of key terms in their collections, based on which a meta-document can be created as representative of a neighbor's topical area. A query is then compared with each meta-document, represented by ***DF*INF*** (see Equation 2), to generate the cosine similarity vector S.

7.2.3. DEG Search: Degree-based Greedy Routing. In the degree-based strategy, we further assume that information about neighbors' degrees, i.e., their numbers of neighbors, is 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. Related degree-based methods were found to be useful for decentralized search in power-law networks [Adamic et al. 2001; Adamic and Adar 2005].

*7.2.4. SimDeg: Similarity*Degree Greedy Routing.* The ***SimDeg*** method is to combine information about neighbors' relevance to a query and their degrees. Simsek and Jensen [2008] 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 uses 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 will be forwarded to the neighbor with the highest sd value. Simsek and Jensen [2008] showed that this combined method is sensitive to the ratio of values between two neighbors, not the actual values that might not be accurately measured.

7.2.5. SimPGS: Size-based Database Selection. Collection size (the number of documents) is an important characteristic for resource selection in federated search [French et al. 1999; Thomas and Hawking 2009]. Si and Callan [2003b] introduced a database selection method in which the estimated relevance of a database given a query q is computed by $\sum p(rel|d)p(d|C)N_c$, where $p(rel|d)$ is the probability of a document d being relevant and N_c the collection size to be estimated. In this work, we assume collection sizes are known to neighbors to avoid further complexity of database sampling and size estimation. Replacing $\sum p(rel|d)p(d|C)$ with an overall collection (meta-document) relevance/similarity score S_c, ***SimPGS*** is computed by $S_c \cdot N_c$. We use this method in the TREC'09 Web track tasks described in section 8.2.

7.3. System Interconnectivity and Network Clustering

For network clustering, the first step is to determine how many links (degree d_u) each distributed system u should have. Once the degree is determined, the system will interact with a large number of other systems (from a random pool) and select only d_u systems as neighbors based on a connectivity probability function guided by the clustering exponent α.

In experiments on the ClueWeb09B collection, we collect information about each web site (treated as an agent/system) incoming hyperlinks and normalize the in-degrees as their d_u values. We control the range of degree distribution $[d_{min}, d_{max}]$ for the normalization and study its impact on search performance. Given the number of incoming hyperlinks d'_u of system u, the normalized degree is computed by:

$$d_u = d_{min} + \frac{(d_{max} - d_{min}) \cdot (d'_u - d'_{min})}{d'_{max} - d'_{min}} \tag{5}$$

where d'_{max} is the maximum degree value in the hyperlink in-degree distribution and d'_{min} the minimum value in the same distribution. Once degree d_u is determined from the degree distribution, a number of random systems/agents will be added to its neighborhood pool such that the total number of neighbors $\hat{d}_u \gg d_u$, e.g., $\hat{d}_u = 1,000$ given $d_u = 30$. Then, the agent in question (u) queries each of the $\hat{d}_u$ neighbors (v) to determine their topical distance r_{uv}. Finally, the following connection probability function is used by system u to decide who should remain as neighbors (to build the interconnectivity overlay):

$$p_{uv} \propto r_{uv}^{-\alpha} \tag{6}$$

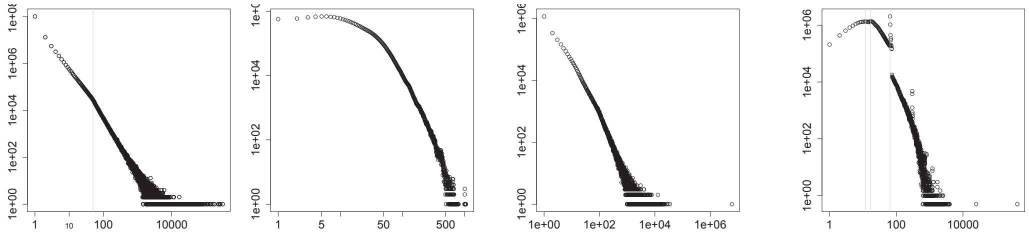
where α is the *clustering exponent* and r_{uv} the pairwise topical (search) distance. The finalized neighborhood size will be the expected number of neighbors, i.e., d_u. With a positive α value, the larger the topical distance, the less likely two systems/agents will connect. As illustrated in Figure 4, large α values lead to highly clustered networks while small values produce random networks with many topically remote connections or weak ties.

8. EXPERIMENTAL DESIGN

8.1. Data Collection

We rely on the ClueWeb09 Category B collection created by the Language Technologies Institute at Carnegie Mellon University for IR experiments. The ClueWeb09 collection contains roughly 1 billion web pages and 8 billion outlinks crawled during January - February 2009. The Category B is a smaller subset containing the first crawl of 50 million English pages from 3 million sites with 454 million outlinks. The ClueWeb09 dataset has been adopted by several TREC tracks including Web track and Million Query track [Clarke et al. 2009a]. Additional details about the ClueWeb09 collection can be found at http://boston.lti.cs.cmu.edu/Data/clueweb09/.

A hyperlink graph is provided for the entire collection and the Category B subset. In the Category B subset, there are 428,136,613 nodes and 454,075,604 edges (hyperlinks). Nodes include the first crawl of 50 million pages and additional pages that were linked to. Only 18,607,029 nodes are the sources (starting pages) of the edges (average 24 outlinks per node) whereas 409,529,584 nodes do not have outgoing links captured in the subset. Analysis of the Category B hyperlink graph produces Figures 6 (a) in-degree frequency distribution and (b) out-degree distribution (on log/log coordinates).



(a) In-degree (b) Out-degree (c) Site size (# pages) (d) Page size (in KB)

Fig. 6. ClueWeb09 Category B: Statistical Distributions. X denotes each variable: (a) in-degree, (b) out-degree, (c) site size, and (d) page size. Y represents the frequency (# occurrences) of the X value.

Based on $50,221,776$ pages extracted from $2,777,321$ unique domains (treated as sites) in the Category B subset, we have also analyzed # pages per web site distributions. The mean number of pages per site is 18. The distribution of the number of pages per site is shown on log/log coordinates in Figure 6 (c). Figure 6 (d) shows page size (text length) frequency distribution on log/log coordinates. There are a couple of visible high points on the graph – that is, many web pages have a content length of roughly 12 KB, 17 KB, or 65 KB. The mean size is $1,109$ KB while the median is 622 KB.

8.2. Network Model

Each agent represents an IR system serving a collection of pages/documents. We assume that there is no global information about all document collections. Nor is there centralized control over individual agents. Agents have to represent themselves using local information they have and evaluate relevance based on that. Using the ClueWeb09 collection, we treat a web site as an agent and use hyperlinks between sites to construct the initial network. Network clustering will then be performed using the method described in Section 7.3.

8.3. Task Levels

Given the size of the web (and likewise the ClueWeb09 collection), it is nearly impossible to manually judge the relevance of every document and establish a complete relevance base. Hence, we primarily rely on existing evidence in data to do automatic relevance judgment. We use documents (with title and content/abstract) as queries to simulate decentralized search on three task levels, each of which involves some arbitrary mechanism to determine whether a document is relevant to a query. In addition, we use TREC'09 web track ad hoc topics and relevance judgments for experiments on short web queries. We elaborate on the four levels below.

8.3.1. Task Level 1: Threshold-based Relevance Search.

The first level involves finding documents with relevant information. Relevant documents are considered few, if not rare, given a particular information need. For evaluation purposes, we will first perform centralized IR operations on the entire collection and treat top-ranked documents (e.g., top 100 of 50 million) as the relevant set, which will then be used in decentralized IR experiments for relevance judgment. The approach is potentially biased by the centralized IR system employed and is therefore not entirely objective. However, this will establish an evaluation baseline and provide basic ideas about how well search methods work. This approach has been used commonly in research such as [Bawa et al. 2003; Lu and Callan 2006].

8.3.2. Task Level 2: Co-citation-based Authority Search.

The second task level involves finding agents that are best "regarded" as relevant to a query (i.e., a web page). On this level, we define relevant documents as those that are frequently *cited together* (linked to) with the given query document. On the web, citation-based (link-based) techniques have been shown to effectively identify authority evidence [Page et al. 1998; Kleinberg et al. 1999]. More importantly, research showed co-citation techniques are very accurate at discovering similar, important (web) documents [Dean and Henzinger 1999]. This task level, relying on co-citation patterns as relevance/authority judgment, is potentially more objective and more challenging than the first level. It can also be seen as popular[5] item search because a web document receives many in-links (and co-citations) only when it has achieved some popularity level.

8.3.3. Task Level 3: Rare Known-Item Search (Exact Match).

The third task level, presumably most challenging, is to find the source of a given document (query). Specifically, when a query document is assigned to an agent, the task involves finding the site or author who created it and therefore hosts it. In other words, in order to satisfy a query, an agent must have the *exact* document in its local collection. The strength of this task is that relevance judgment is well established provided the relative objectiveness and unambiguity of creatorship or a "hosting" relationship. Among the 50 million pages in the ClueWeb09 collection, for example, there are likely only a few copies of a document being searched for. The extreme rarity will pose a great challenge on the proposed decentralized search methods.

8.3.4. Task Level 4: TREC'09 Web Track Ad Hoc Topics.

Lastly, we use ad hoc topics and relevance judgments from TREC'09 web track for experiments on web searches with short queries. The collection provides 50 topics and $13,118$ relevance judgment records for submissions based on ClueWeb09B [Clarke et al. 2009b]. We use free-text terms without detailed descriptions for query representation and anchor texts (extracted from links pointing to $8,307,747$ pages in ClueWeb09B) for document representation. Web sites (domains) with at least one document being judged relevant or non-relevant are included in the distributed network, resulting in $1,916$ sites/agents with $635,332$ pages. Although the size of the network is not very large (e.g., compared to the $100,000$-system experiments we conduct in this research), this task level provides another well justifiable baseline in the web search context.

[5] *Popularity* here is in terms of the frequency of an item being cited, rather than the number of copies that have been duplicated, e.g., in peer-to-peer networks.

8.4. Additional Independent Variables

8.4.1. Degree Distribution: d_{min} and d_{max}. We will use the degree (in-degree) distribution of the ClueWeb09B hyperlink graph and normalize the distribution to fall in a range $[d_{min}, d_{max}]$. With different d_{min} and d_{max} values, the degree distribution will continue to follow a pattern similar to Figure 6 but is with a different degree distribution exponent γ because the slope on log-log changes. We use various degree ranges, e.g., $[30, 30]$, $[30, 60]$, and $[30, 120]$, to examine the impact of degree distribution on decentralized searches. With the range $[30, 30]$, all agents/systems share one common degree, i.e., 30.

8.4.2. Network Clustering: Clustering Exponent α. Based on a degree d_u picked from a distribution, the clustering exponent α controls the probability of topically relevant or irrelevant agents connecting to each other (see Section 7.3 for details). One central question in this study is about the impact of $\alpha \in [0, \infty)$ on search performance. As shown in Figures 3 and 4, when $\alpha = 0$, the network becomes a random network as interconnectivity is independent of topical relevance. When $\alpha \to \infty$, the network is extremely clustered, in which agents only connect to very close (topically relevant or similar) neighbors. To establish a reasonable range of α, we replicated experimental simulations of Kleinberg [2000] on various network size scales and conducted pilot experiments in the IR context as well. We identified $\alpha \in [0, 1, 2, 3, 4, 5]$ as a good range for experiments in this study.

8.4.3. Maximum Search Path Length L_{max}. Provided the importance of achieving overall network utility and scalability of search, we propose the use of a parameter, namely, the maximum search path length L_{max}, which defines the longest path each search allowed for query traversal. If a search reaches the maximum value, even when the query has not been answered, the task will be terminated and returned to its originator.

8.5. Evaluation: Dependent Variables

Given diverse users in an open, dynamic environment, some queries are likely to be narrowly defined. The study focuses on how relevant information can be found and scalability of decentralized searches. We emphasize the finding of highly relevant information in large distributed environments and propose the use of the following evaluation measures.

8.5.1. Effectiveness: Classic IR Metrics. We use traditional IR effectiveness metrics such as precision, recall, F, and discounted cumulative gain (DCG) for effectiveness evaluation. 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 $F_1 = 2 \cdot P \cdot R/(P + R)$. These measures will be used to evaluating results from *exact match* searches. For each query, recall is 1 when an exact match is found; recall is 0 if otherwise.

For ranked retrieval results, Jarvelin and Kekalainen [2002] proposed several cumulative gain metrics. Specifically, given a ranked list of retrieval results, the discounted cumulative gain at a rank position p is defined as:

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2 i} \tag{7}$$

where rel_i is the relevance value of the item at position i. Because search results (and rank list length) vary on queries, a normalized DCG function was also proposed for values to be compared and aggregated across multiple queries. Given an ideal DCG at position p (DCG achieved based on sorted relevance) $iDCG$, the normalized DCG is computed by:

$$nDCG_p = \frac{DCG_p}{iDCG_p} \tag{8}$$

Normalized discounted cumulative gain at position 10 ($nDCG_{10}$) will be used in *relevance search* and *authority search* experiments, where a federated ranked list of documents gets retrieved for each query. We use $nDCG_3$ for TREC'09 Web track evaluation given limited relevance judgments and the focus on very top results in web searches.

8.5.2. Efficiency. For efficiency, the maximum search path length L_{max} (or the max number of hops allowed) will be controlled in each experiment whereas the actual search path length will be recorded. The average search length of all tasks can therefore be calculated to measure efficiency: $\bar{L} = \sum_{i=1}^{N_q} L_i/N_q$, where L_i is the search path length of the i_{th} query and N_q the total number of queries. With shorter path lengths, the entire distributed system is considered more efficient given fewer agents involved in searches.

8.5.3. Scalability Analysis. One important objective of this research is to understand how decentralized IR systems can function and scale in large, heterogeneous, and dynamic network environments. Findings are useless if they are only based on small network sizes. For scalability, we will run experiments on different network size scales. Effectiveness vs. efficiency patterns will be compared to discover how search methods work on the size scales. Best results in terms of efficiency and effectiveness will also be compared and plotted against network size. Their functional relationships with network size will be analyzed.

8.6. Parameter Settings

Table I summarizes some of the major independent variables discussed above and presents combinations of parameters to be tested in the proposed experiments. Under each experimental setting, each of four proposed search methods will be employed to conduct searches. Effectiveness and efficiency results will be recorded automatically for later analysis. Parameter values in the table have been chosen based on pilot experiments conducted earlier.

Table I. Major Experimental Settings. Symbols: N denotes network size, i.e., the number of distributed system in the network; L_{max} denotes maximum search path length allowed in each experiment; α is clustering exponent. Main experiments will be focused on Exact Match searches in networks of a degree range $d \in [30, 60]$.

N	(L_{max})	Task Level	α	Degree Range	Search Method
10^2	(20)	Relevance Search	0	$[30, 30]$	Random Walk (RW)
			1		
10^3	(100)		2		Similarity (SIM) Search
		Authority Search	3	**[30,60]**	
10^4	(500)		4		Degree (DEG) Search
			5		
10^5	(2500)	**Exact Match**	..	$[30, 120]$	Similarity+Degree (SimDeg)
$1,916$	(100)	TREC'09 Web	0-5	$[4, 8]$	SIM, DEG, SimDeg, PGS, SimPGS

8.7. Simulation Procedures

Experiments are conducted on a Linux cluster of 5 PC nodes, each has Dual Intel Xeon E5620 (2.4 Ghz) Quad Core Processors (8 processors), 24 GB fully buffered system memory, and an REHL 6 installation. The nodes are connected internally through a dedicated 1Gb network switch. The agents (distributed IR systems) are equally distributed among the 40 processors, each of which loads an agent container in Java, reserves 2GB memory, and communicates to each other. The Java Runtime Environment version for this study is OpenJDK $1.6.0\_22$. We provide the pseudo code about the experimental procedures in Algorithm 1. For relevance and authority searches, a threshold of document matching score 1.0 based on Equation 3 is used to determine whether a potentially relevant system has been reached. The score is normalized by document length and query terms, and can be higher than 1.0.

ALGORITHM 1: Simulation Experiments

```
for each Network Size $N \in [10^2, 10^3, 10^4, 10^5]$ do
    for each Task Level $\in [Relevant, Authority, ExactMatch]$ do
        for each Clustering Exponent $\alpha \in [0, 1, 2, 3, 4, 5]$ do
            rewire the network using $\alpha$
            for each Search Method $\in [SIM, SimDeg, DEG, RW]$ do
                for each Query do
                    repeat
                        forward a query from one agent/system to another
                    until relevant found OR search path length $L \geq L_{max}$
                    if relevant found, i.e., similarity scores surpass a threshold then
                        if task is Relevant Search OR Authority Search then
                            query additional neighbors for more relevant documents
                        else if task is Exact Match Search then
                            retrieve the most similar/relevant document
                        end if
                        send the results back to the first agent/system
                        merge and rank all retrieved documents
                    else
                        send message back about failure
                    end if
                end for
                measure search effectiveness: precision, recall, $F_1$, nDCG$_{10}$
                measure search efficiency: search path length $L$ and search time $\tau$
            end for
        end for
    end for
end for
```

9. EXPERIMENTAL RESULTS

In the presentation of experimental results, we first discuss *rare known-item (exact match) searches* in Section 9.1. We report on detailed results in Section 9.1 and analyze the *clustering paradox* in Section 9.2. We evaluate scalability of searches in Section 9.3 and scalability of network clustering in Section 9.4. Section 9.5 presents results on search performances when degree distribution varies. Section 9.6 discusses additional results from *relevance search* and Section 9.7 on *authority search* .

9.1. Rare Known-Item (Exact Match) Search

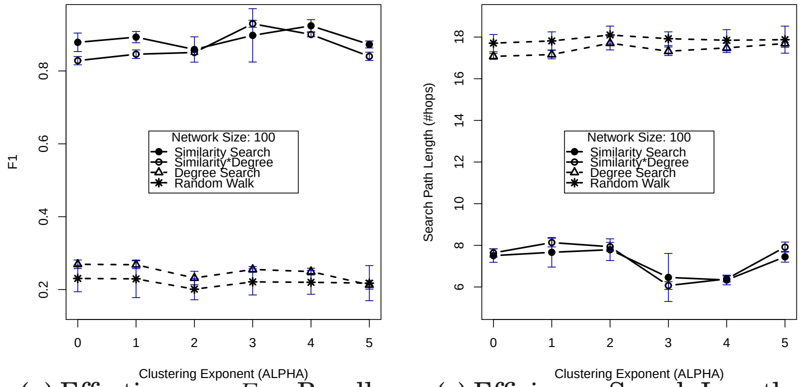
For main experiments, we identified 85 documents (web pages with title and content) from 100 most highly connected (popular) web domains (systems) by random sampling and manual selection[6]. These 85 web documents were used as queries in most of our decentralized search experiments. Main experiments were focused on finding exact match documents (rare known items) because this task level, challenging in a distributed environment, can be objectively evaluated.

We sorted all Web domains in the ClueWeb09B collection by connectivity/popularity and started with the 100 most highly connected web domains for experiments on the 100-system network. Then we extended the network to include more systems on the sorted list for larger network sizes $N \in [10^2, 10^3, 10^4, 10^5]$. We set the max search path length L_{max} to $[20, 100, 500, 2500]$ for the different network sizes respectively. Table II shows the number of web documents in each network thus constructed.

Table II. Network Sizes and Total Numbers of Docs

Task	Exact Match/Relevance/Authority				TREC Web
Network Size N	100	$1,000$	$10,000$	$100,000$	$1,916$
Number of Docs N_D	0.5 million	1.7 million	4.4 million	10.5 million	0.6 million

With each network size, we varied the clustering exponent α for network construction and tested each of the four proposed search methods, namely, Random Walk (RW), Similarity Search (SIM), Degree Search (DEG), and Similarity*Degree Search (SimDeg). To determine the number of links (degree) each system should have, we utilized the Web graph of the ClueWeb09B collection and normalized the degree distribution to the range of $[30, 60]$[7]. In all experiments, no document identification information was used for indexing or searching. This section presents main experimental results on rare known-item (exact-match) searches.



(a) Effectiveness: F_1 = Recall (c) Efficiency: Search Length

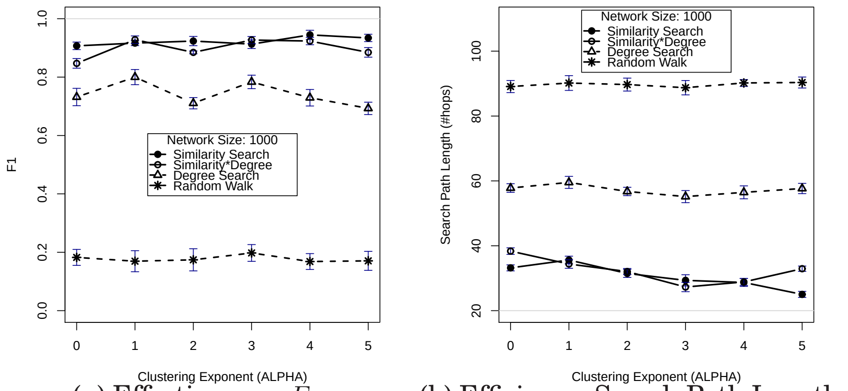
Fig. 7. Effectiveness and Efficiency on 100-System Network

9.1.1. 100- and 1000-System Networks. In all rare known-item searches, precision was maintained at 1.0 because a document was retrieved only when it exactly matched

[6] We manually removed from queries web documents with either very little or very common text content, e.g., a web site's terms of use page.

[7] The majority had a degree of 30 while very few had 60 connections. Degree ranges $[30, 30]$ and $[30, 120]$ were used in additional experiments.

a query document. In 100-system networks (shown in Figures 7), similarity search (SIM) and similarity*degree (SimDeg) methods performed very well in terms of effectiveness and efficiency, showing a very large advantage in F_1 (and recall) over degree (DEG) search and random-walk (RW) methods. For example, in the 100-system network, SIM and SimDeg searches achieved above 0.9 F_1 at a network clustering level ($\alpha = 3$) while DEG and RW searches only had F_1 values around 0.2. Whereas SIM and SimDeg methods only involved 5 systems (5%) and took less than 150 milliseconds to reach F_1 0.9 at $\alpha = 3$, RW and DEG searches traversed $17 - 18$ systems (and more than 400 milliseconds) for a roughly 0.2 F_1. The differences are large and statistically significant[8].



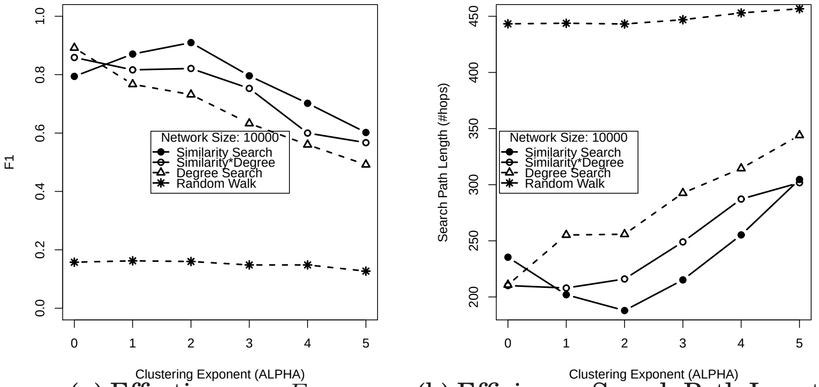
(a) Effectiveness: F_1 (b) Efficiency: Search Path Length

Fig. 8. Performance on 1,000-System Network

In $1,000$-system networks (Figure 8), SIM and SimDeg search methods continued to show large advantages on search performance. SIM search achieved its best performance higher than 0.9 F_1 by only traversing less than 30 systems (or 3%) in the network. The RW method, as a baseline, involved roughly 90 systems to reach 0.2 F_1. In 100- and 1000-system networks, the impact of network clustering (guided by α) on search performance is not clearly shown. As discussed in Section 4.2, among others, network structure is increasingly relevant in larger networks, where it becomes important to find a balance between strong ties for search guidance and weak ties for "jumps." In the following sections, we will discuss results and plots from experiments on the $10,000$- and $100,000$-system networks, and present initial evidence, which appears to support the *Clustering Paradox*.

9.1.2. 10,000-System Network. When the network was extended to $10,000$ systems, some interesting patterns on search performances began to emerge. As shown in Figure 9 (a) and (b), while SIM and SimDeg searches continued to dominate search performance both in effectiveness (F_1) and efficiency (search path length), some network clustering levels appeared to produce better results than others. For example, SIM search achieved best effectiveness (highest F_1 score) and efficiency (smallest search path length) at $\alpha = 2$. Reducing α (weaker clustering) or increasing α (stronger clustering) led to degraded performances. The plots provide visual evidence about the *Clustering*

[8] In this article, reported differences are statistically significant based on (generalized) linear regression unless stated otherwise.



(a) Effectiveness: F_1 (b) Efficiency: Search Path Length

Fig. 9. Performance on 10,000-System Networks

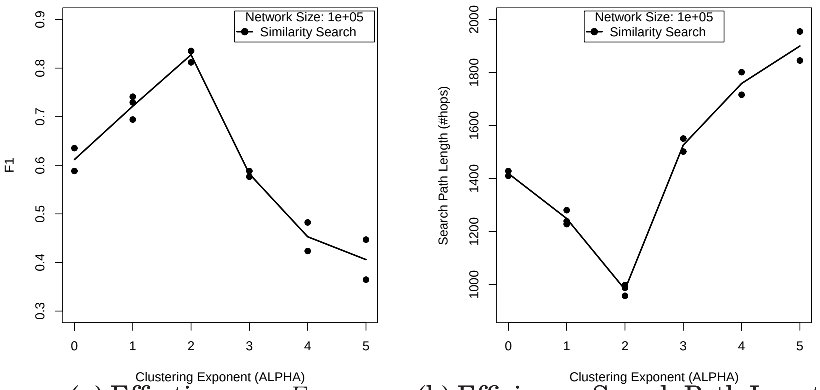
Paradox in IR, in which neither under- nor over-clustering is desirable. Section 9.2 presents an in-depth statistical analysis of this phenomenon.

DEG search performances over clustering levels $\alpha \in [0, 1, .., 5]$ follow a very different pattern. Interestingly, DEG search achieved its best performance at $\alpha = 0$, i.e., with no clustering in a random network. The best result of DEG search result surpassed the performance of SimDeg search but was outperformed by SIM search (at $\alpha = 2$) in terms of efficiency. The SimDeg method, which combines similarity and degree information, appears to have mixed the performances of SIM and DEG methods in Figure 9 (a) and (b).

It is important to point out that strong performances of DEG and SimDeg methods were due, in part, to the particular "popular" queries used in the experiments. As described early, the queries were pages identified from most highly connected (linked-to) domains in the ClueWeb09B collection. In the 100-system network, DEG performed poorly because most of the systems were highly connected domains and it is difficult to differentiate one from another based on degree values. DEG-related methods performed better in larger network because of the degree-based discriminative power is useful in directing queries to popular targets. We shall see in section 9.8 that the patterns of DEG and SimDeg performances were not replicated in TREC'09 web track tasks.

9.1.3. 100,000-System Network. Because SIM search produced superior results in the $[10^2, 10^3, 10^4]$-system networks, we concentrated on SIM searches for experiments on the largest network in the study, i.e., the network of $100,000$ systems. Another reason for not conducting experiments on the other search methods was because of time constraints – other methods such as RW were much less efficient and would have taken a very long time to finish with the large network size 10^5.

A similar pattern on SIM search performance continued to appear in the $100,000$-system network, where more than 10 million documents were served in a distributed manner. As shown in Figures 10 (a) and (b), SIM search continued to achieve its best effectiveness and efficiency at $\alpha = 2$. Smaller α (weaker clustering) or larger α values (stronger clustering) led to performance degradation. The inflection at $\alpha = 2$ looks much sharper in the $100,000$-system network than in the $10,000$-system network, suggesting a potentially stronger impact of network clustering on search performance. We conducted further analysis and relied on statistical tests to better understand the impact of connectivity, to predict the scalability of search, and to answer related research questions. We discuss these tests and findings in the following Sections 9.2 and 9.3.

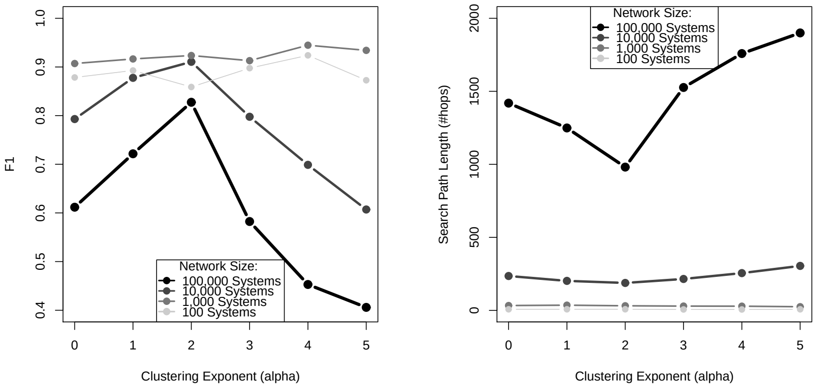


(a) Effectiveness: F_1 (b) Efficiency: Search Path Length

Fig. 10. Performance on 100,000-System Network. Line is the average of individual data points at each α level.

9.2. Clustering Paradox

Given that the Similarity (SIM) search method was shown to perform much better than the other methods, we focus on SIM search in the discussion about the impact of network clustering on search performance.



(a) Effectiveness: F_1 (b) Efficiency: Search Path Length

Fig. 11. Performance on All Network Sizes

Figure 11 shows SIM search performances over network clustering levels $\alpha \in [0, 1, 2, 3, 4, 5]$ for network sizes $N \in [10^2, 10^3, 10^4, 10^5]$ in terms of (a) effectiveness and (b) efficiency. Both sub-figures demonstrate that network structure (clustering) had an important impact on decentralized information retrieval performance, particularly in larger networks. Some level of network clustering (i.e., $\alpha = 2$ in the experiments) supported best search performance. Effectiveness and efficiency degraded when there was stronger or weaker clustering.

While search efficiency (search path length) under different clustering conditions only differed slightly or moderately in the 100-, $1,000$-, and $10,000$-system networks, the difference was more dramatic in the network of $100,000$ systems (Figure 11 (b)). For example, when α increased from $2 \rightarrow 3$ in the $10,000$-system network, search path length increased from about 190 to 220, roughly a 30 hops (or 15%) increase. The same degree of network clustering change, however, resulted in a much larger increase of search path length roughly from 1000 to 1550, by 550 hops (or 55%) in the $100,000$-system network.

Statistical tests indicated that SIM search achieved significantly better results with the balanced level of network clustering (i.e., at $\alpha = 2$) than with over- or under-clustering. Significant differences appeared in both the $10,000$-system and $100,000$-system networks. Results from the statistical analysis are shown in Tables III, IV ($10,000$-system network) and Tables V, VI ($100,000$-system network). We elaborate on the results below[9].

Table III. SIM Search: Network Clustering on Effectiveness in Network 10,000

Comparison	Difference in F_1	Error	t value	$Pr(>\|t\|)$	R^2
$\alpha : 0 \to 1$	0.08471	0.01299	6.519	0.00018 ***	0.842
$\alpha : 1 \to 2$	0.03294	0.01065	3.092	0.015 *	0.544
$\alpha : 2 \to 3$	-0.1129	0.009843	-11.47	0.000003 ***	0.943
$\alpha : 3 \to 4$	-0.09882	0.006444	-15.34	0.00000032 ***	0.967
$\alpha : 4 \to 5$	-0.09176	0.01299	-7.062	0.00011 ***	0.862

Table III compares SIM search effectiveness scores (F_1) between every two consecutive levels of clustering (α) on the $10,000$-system network. It shows that when clustering exponent α increased from $0 \to 1 \to 2$, i.e., from random/no clustering to some level of clustering, search effectiveness improved. When α continued to increase from $2 \to 3 \to 4 \to 5$, search effectiveness degraded.

Table IV. SIM Search: Network Clustering on Efficiency in Network 10,000

Comparison	Difference in Search Length	Error	t value	$Pr(>\|t\|)$	R^2
$\alpha : 0 \to 1$	-33.39	5.177	-6.45	0.0002 ***	0.839
$\alpha : 1 \to 2$	-14.1	4.422	-3.188	0.013 *	0.56
$\alpha : 2 \to 3$	27.28	3.27	8.341	0.000032 ***	0.897
$\alpha : 3 \to 4$	40.09	4.195	9.557	0.000012 ***	0.919
$\alpha : 4 \to 5$	49.34	3.972	12.42	0.0000016 ***	0.951

Similar patterns also appear in Table IV on SIM search efficiency in the $10,000$-system network. When α increased from $0 \to 5$, the general trend was that search performance first improved (to smaller search path lengths) and then degraded (to longer search path lengths). The inflection point appeared at $\alpha = 2$, where SIM search performed at its best.

Table V. SIM Search: Network Clustering on Effectiveness in Network 100,000

Comparison	Difference in F_1	Error	t value	$Pr(>\|t\|)$	R^2
$\alpha : 0 \to 1$	0.1098	0.02531	4.338	0.023 *	0.862
$\alpha : 1 \to 2$	0.1059	0.01617	6.548	0.0028 **	0.915
$\alpha : 2 \to 3$	-0.2451	0.01103	-22.21	0.0002 ***	0.994
$\alpha : 3 \to 4$	-0.1294	0.02999	-4.315	0.05 *	0.903
$\alpha : 4 \to 5$	-0.04706	0.0506	-0.93	0.45	0.302

Table V shows consistent results on the $100,000$-system network, in which best search effectiveness and efficiency were also found at $\alpha = 2$. As compared to the $10,000$-system network, the impact of network clustering of $100,000$ systems on search performance appeared to be stronger. For example, in the 10^4 network, changing α from $1 \to 2$ resulted in an F_1 increase of 0.03 and 14 hops shorter in search path length.

[9] Significance codes: *** p < 0.001, ** p < 0.01, * p < 0.05, . p < 0.1, based on linear regression.

The same degree of network clustering change led to a 0.11 increase in F_1 and a search path shortened by 268 in the 10^5-system network.

Table VI. SIM Search: Network Clustering on Efficiency in Network 100,000

Comparison	Difference in Search Length	Error	t value	$Pr(>\|t\|)$		R^2
$\alpha : 0 \rightarrow 1$	-170.1	21.8	-7.801	0.0044	**	0.953
$\alpha : 1 \rightarrow 2$	-267.9	20.11	-13.33	0.00018	***	0.978
$\alpha : 2 \rightarrow 3$	545.1	24.08	22.64	0.00019	***	0.994
$\alpha : 3 \rightarrow 4$	232.3	49.13	4.729	0.042	*	0.918
$\alpha : 4 \rightarrow 5$	141.3	69.37	2.037	0.18		0.675

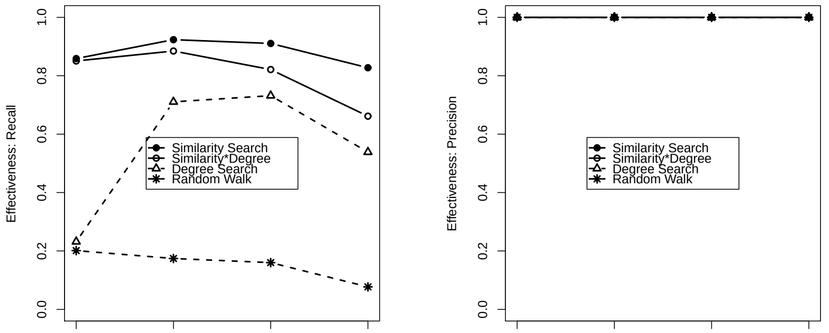
Overclustering also had a stronger impact in the 10^5 network than in the 10^4 network. When α increased from $2 \rightarrow 3$ in the 10^4 network, F_1 had a 0.11 loss while search path length increased by 27. The same degree of change in the 10^5 network resulted in much more dramatic performance loss – a 0.25 loss in F_1 and a 545 increase in search path length.

These tests support our first hypothesis about the *Clustering Paradox*, that there does exist a level of network clustering ($\alpha = 2$ in our experiments), below and above which search perform degrades. In other words, that specific level of clustering supports best search performance in terms of both effectiveness and efficiency.

One additional important finding is that the *clustering paradox* appears to have a scaling effect on search performances. The negative impact of under- or over-clustering on search effectiveness and efficiency is much greater in larger networks. Small performance degradation in a small network may lead to a much greater disadvantage when the network grows in magnitude. This scaling effect requires closer examination.

9.3. Scalability of Search

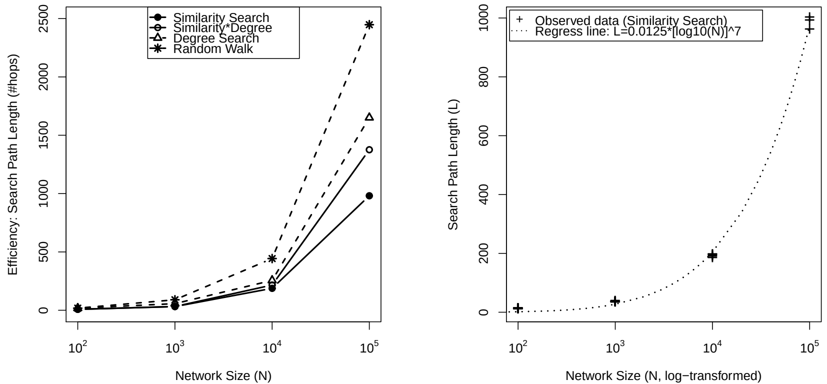
For each network size, we identified network clustering conditions under which superior performance was observed (i.e., at $\alpha = 2$ in the experiments). We plotted recall and precision vs. network size at $\alpha = 2$ in Figure 12. As discussed earlier, SIM and SimDeg searches consistently achieved very high recall and precision across the various network sizes, much better than DEG and RW methods. DEG search tended to perform better in larger networks than in smaller ones given the *popular* nature of queries we used.



(a) Recall vs. Network Size (log) (b) Precision vs. Network Size (log)

Fig. 12. Scalability of Search Effectiveness at $\alpha = 2$

Figure 13 (a) shows average search path length (efficiency) vs. network size at $\alpha = 2$. Search path length for RW and DEG increased dramatically in larger networks while the increases for SIM and SimDeg were relatively moderate. SIM and SimDeg methods appeared to be much more scalable than RW and DEG methods.



(a) Search Length vs. Network Size (b) Scalability of SIM Search

Fig. 13. Scalability of Search Efficiency at $\alpha = 2$. X (network size) is log-transformed.

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. 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/system in the network and no effort is needed to search further. The best fit for search path length L was produced by the model in Table VII, in which $L = 0.0125 \cdot (\log_{10} N)^7$ has a nearly perfect $R^2 = 0.999$.

Table VII. SIM Search: Search Path length vs. Network size

Search Path Length: $L \sim 0 + \beta(\log_{10} N)^7$, where N is network size.				
	Coefficient Estimate	Standard Error	t value	$Pr(> \|t\|)$
β	0.0125	$7.04e - 05$	177	$5e - 52$ ***
$R^2 = 0.999$ (adj. 0.999), $F = 31457$ on 1 and 34 DF				

Figure 13 (b) shows actual data points on search path length L vs. network size N, together with values (dotted line) predicted by the regression model $L = 0.0125 \cdot (\log_{10} N)^7$. Overall, the scalability analysis supports search time as a poly-logarithmic function of network size (hypothesis 2) – so that when an information network continues to grow in magnitude, it is still promising to conduct effective search operations within a manageable time limit. This poly-logarithmic scalability was supported by a particular network clustering level, i.e., $\alpha = 2$ in the experiments. Although we found the order of the poly-logarithmic relationship to be roughly 7 in this study, a smaller exponent can be expected when other factors on network structure and search methods can be optimized.

9.4. Scalability of Network Clustering

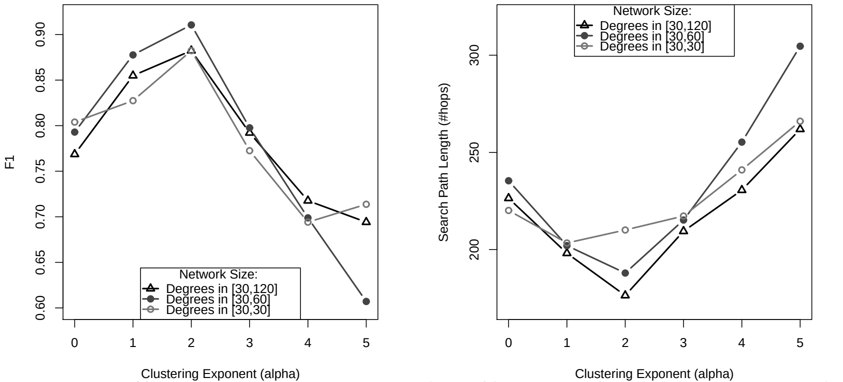
It is important to understand how much effort is needed to construct and maintain a network structure for effective and efficient search functions. Our search methods relied on local indexes and a network structure self-organized by distributed systems in the network. Without global information or centralized control, network clustering was performed locally – distributed systems formed a network based on their limited opportunities to interact, individual preferences, and budgets.

This local mechanism for clustering demonstrated a high level of scalability. In our experiments, average clustering time τ_c remained relatively constant, < 1 sec, across all network size scales $N \in [10^2, 10^3, 10^4, 10^5]$ (see also Ke and Mostafa [2010]). This constant time characteristic was due to the fact that clustering was performed locally and simultaneously. Each system relied on its own computing resources and established its connectivity independently. The number of neighbors one had to communicate to in the clustering process was also a constant across the different network sizes.

This highly scalable, local clustering mechanism is particularly useful for coping with dynamics and heterogeneity in the distributed environment. Even when changes occur in the network (e.g., with system arrival/departure and/or new content), the clustering mechanism does not require the entire community to respond to the changes. Instead, only neighbor systems directly connected to changed nodes will need to receive updates. Yet as shown by experimental results, this very local mechanism supports effective and efficient discovery of relevant information in the global space.

9.5. Impact of Degree Distribution

The main experiments discussed in earlier sections were conducted on a degree (d_u, number of neighbors/connections per system) distribution normalized to $d_u \in [30, .., 60]$. For example, in experiments on the $10,000$-system network, we obtained the number of incoming hyperlinks each of the 10^4 systems (web sites) received from the entire ClueWeb09B collection and established the original degree distribution. We normalized all degrees to fit in the range of $[30, 60]$ using Equation 5 described in Section 7.3. These degrees were then used in experiments to determine how many neighbors a system was supposed to have for network construction and clustering.



(a) Effectiveness: F_1 (b) Efficiency: Search Path Length

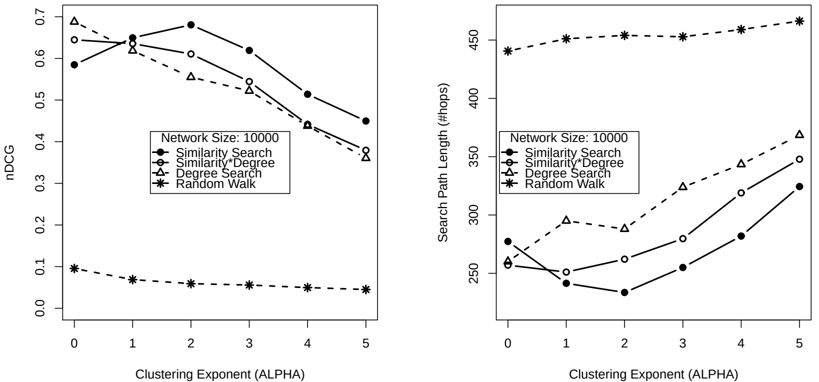
Fig. 14. SIM Search Performance with Varied Degree Ranges

We varied the range of degrees and studied the impact of degree distribution on search performance. In addition to range $[30, 60]$, we also used $[30, 30]$ and $[30, 120]$ for experiments on the network of $10,000$ systems. With range $[30, 30]$, all systems had a uniform degree, i.e., 30. With range $[30, 120]$, degree values spread over larger values as compared to those $\in [30, 60]$.

Experimental results with different degree ranges $[30, 30]$ and $[30, 120]$, in addition to main experiments on range $[30, 60]$, are shown in Figures 14 (a) and (b). Overall, best performances on the three different degree distributions $[30, 30]$, $[30, 60]$, and $[30, 120]$ all appeared around $\alpha = 2$. Statistical tests on the degree ranges produced consistent results on the impact of network clustering. While search performance changes when degree distribution varies[10], evidence continues to support the existence of the *Clustering Paradox*.

9.6. Relevance Search

At the task level of *Relevance Search*, the goal was not (only) to find exact matches but to find documents that were *relevant* (similar) to each query. Because the ClueWeb09B was a very new, large web collection, there was not a complete human judged relevance base for evaluation. To establish a relevance base automatically, we followed the following arbitrary mechanism, which has been widely used by IR researchers for evaluation of large scale distributed system performance [Bawa et al. 2003; Lu 2007b].



(a) Effectiveness: nDCG at 10 (b) Efficiency: Search Path Length

Fig. 15. Relevance Search Performance on 1,000-System Network

First we built a centralized IR system using the core search engine function of our distributed systems and indexed 4.4 million documents that appeared in the $10,000$-system network. Then, we issued each query to the centralized IR system and retrieved top 100 documents. We treated the 100 documents as the *only* relevant documents among all 4.4 million pages for each query and used similarity scores produced by the centralized system as their relevance to the query. Finally, queries were issued to the $10,000$-system network to obtain a federated rank list of 10 documents. The results were compared to the *gold standard* produced by the centralized system and were evaluated using *normalized discounted cumulative gain* (nDCG) at position 10 (see Section 8.5).

Figure 15 shows experimental data from relevance searches in the $10,000$-system network. Results are consistent with those from *exact match* searches. While RW search continued to be a lower-bound baseline, SIM search performed relatively well, with its best performance at $\alpha = 2$. DEG search achieved superior search performances

[10] The inflection point with degree range $[30, 30]$ appeared to be slightly below 2.

with random/no clustering, i.e., at $\alpha = 0$, and degraded when there was stronger clustering.

Table VIII. SIM Search: Network Clustering on Relevance Search Effectiveness

Comparison	Difference in $nDCG_{10}$	Error	t value	$Pr(>\|t\|)$		R^2
$\alpha : 0 \to 1$	0.06469	0.02042	3.168	0.019	*	0.626
$\alpha : 1 \to 2$	0.03113	0.01309	2.379	0.041	*	0.386
$\alpha : 2 \to 3$	-0.06141	0.01218	-5.04	0.0007	***	0.738
$\alpha : 3 \to 4$	-0.1069	0.00716	-14.93	0.0000057	***	0.974
$\alpha : 4 \to 5$	-0.04658	0.01358	-3.429	0.014	*	0.662

Table IX. SIM Search: Network Clustering on Relevance Search Efficiency

Comparison	Difference in Search Length	Error	t value	$Pr(>\|t\|)$		R^2
$\alpha : 0 \to 1$	-35.9	3.239	-11.08	0.000032	***	0.953
$\alpha : 1 \to 2$	-7.863	2.712	-2.9	0.018	*	0.483
$\alpha : 2 \to 3$	21.44	4.654	4.608	0.0013	**	0.702
$\alpha : 3 \to 4$	25.79	7.07	3.648	0.011	*	0.689
$\alpha : 4 \to 5$	40.41	7.287	5.546	0.0015	**	0.837

We analyzed SIM search performances over different values of $\alpha \in [0, 1, 2, 3, 4, 5]$. Table VIII compares SIM search effectiveness scores ($nDCG_{10}$) between every two consecutive levels of clustering (α) on the $10,000$-system network. It shows that when clustering exponent α increased from $0 \to 1 \to 2$, i.e., from random/no clustering to some level of clustering, search effectiveness improved. When α continued to increase from $2 \to 3 \to 4 \to 5$, search effectiveness degraded. This trend resembles how F_1 changed over α values in exact match searches (compare to Table III).

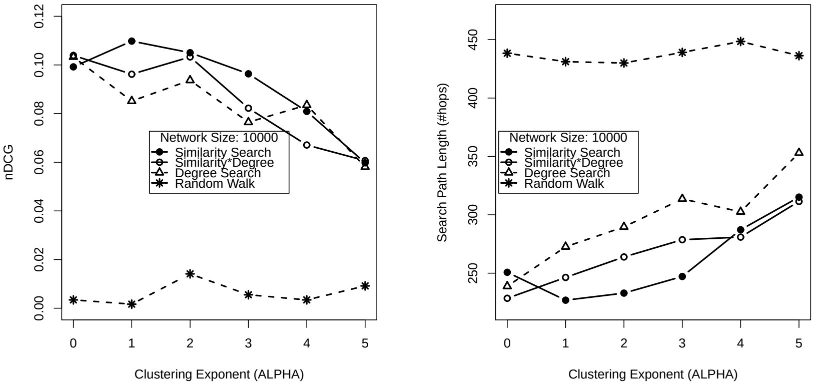
Similar patterns also appear in Table IX on SIM search efficiency in the $10,000$-system network. When α increased from $0 \to 5$, the general trend was that search performance first improved (to smaller search path lengths) and then degraded (to longer search path lengths). The inflection point appeared at $\alpha = 2$, where SIM search performed at its best (compare to Table IV). This provides further evidence that the *Clustering Paradox* mattered in the relevance searches (hypothesis 1).

9.7. Authority Search

Experiments on authority searches were conducted in a manner nearly identical to relevance searches, except for how results were evaluated. In *relevance searches*, decentralized search results from a network were compared to a *gold standard* produced by a centralized search system. In *authority searches*, we relied on co-citation information from the ClueWeb09B web graph to establish a *gold standard* on *relevant authority pages*.

For each of the 85 query documents used in *exact match* and *relevance search* tasks, we identified pages among the 4.4 million in the $10,000$-system network that were co-cited (being linked together) for at least 5 times. The number of citations of each page with the query was then normalized by the total number of citations (in-links) the page received to produce an authority score. We selected 100 web documents/pages with the highest authority scores as the relevance base (gold standard) for each query. Only 38 queries remained because the other queries did not have sufficient co-cited pages. Results from distributed searches in the $10,000$-system network were then compared to the gold standard. We continued to use *normalized discounted cumulative gain* (nDCG) at 10 to evaluate retrieval effectiveness.

Figure 16 presents results from authority search experiments on the $10,000$-system network. As shown in Figure 16 (a), search effectiveness was low in general – SIM,



(a) Effectiveness: nDCG at 10 (b) Efficiency: Search Path Length

Fig. 16. Authority Search Performance on 10,000-System Network

SimDeg, and DEG searches only achieved $nDCG_{10}$ scores slightly higher than 0.1. RW search effectiveness was well below $nDCG$ 0.02. The major reason for the low $nDCG$ scores was because the retrieval fusion (federation) method used in distributed searches only relied on topical similarity scores for ranking retrieved documents. The authority gold standard might have disregarded many content-wise similar pages if they did not have enough co-citations. Nonetheless, this task level provides additional evidence on how system connectivity affects search performance.

In Figures 16 (a) and (b), SIM search effectiveness and efficiency results look consistent with those from relevance searches. For SIM searches, $\alpha = 1$ seemed to support its best performance. Increasing or decreasing α value from 1 degraded both effectiveness and efficiency.

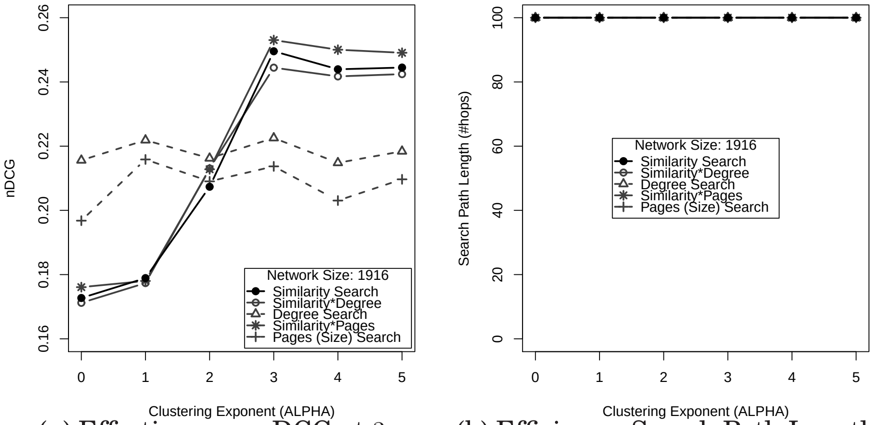
Statistical analysis showed that search performance improved when α increased from $0 \rightarrow 1$ and degraded when α changed from $2 \rightarrow 3 \rightarrow 4 \rightarrow 5$. We found no significant difference between performances at $\alpha = 1$ and at $\alpha = 2$. It is likely that the inflection point is at an α value between 1 and 2. Regardless of the actual network clustering level for best authority search performance, analysis here further supports the existence of *clustering paradox* in the IR context.

9.8. TREC'09 Web Track

As discussed, ad hoc search queries from TREC'09 Web track were performed in a 1916-system network based on anchor text document representation. Each query was forced to traverse 100 hops (search path length) before best results were identified using the result fusion method described in section 7.1.4. We limited each agent's connectivity to a small number of neighbors ($d \in [4, 8]$ neighbors), chosen from the entire community of 1915 (i.e. $1916 - 1$) systems.

We use $nDCG$ at position 3 for effectiveness evaluation because of limited numbers of relevant records in the judgments and the emphasis on top results in web retrieval. Figure 17 shows efficiency and effectiveness results. In Figure 17 (b), search path length (efficiency) is always 100 because this was the number of hops each query was set to traverse in the web track task. Figure 17 (a) shows retrieval effectiveness in terms of $nDCG_3$ over various clustering levels $\alpha \in [0, 5]$.

Degree-based DEG search and collection-size-based PGS search achieved roughly the same level of effectiveness at different network clustering levels. This *flat* pattern of DEG search is very different from what we observed earlier in exact match (section 9.1), relevance search (section 9.6), and authority search (section 9.7). In those



(a) Effectiveness: nDCG at 3 (b) Efficiency: Search Path Length

Fig. 17. TREC'09 Web Search Performance (a 1916-system network)

tasks, DEG and SimDeg searches achieved their best performances at $\alpha = 0$ (with no clustering in a random network) and their performances degraded with stronger clustering. We reason that this was because queries in the earlier tasks were full web documents from highly connected (popular) sites. Degree information was critical to finding those relevant, well-connected sites and weak clustering creates opportunities for sites of various degrees to interconnect, increasing the likelihood for searches to “jump” to *popular* targets.

Results on SIM, SimDEG, and SimPGS methods show similar, strong performances. There is a consistent inflecting pattern in web track searches, where best search performance was achieved at around $\alpha = 3$. SimPGS search, which combined similarity and collection size information, appeared to perform competitively with optimal network clustering (though not significantly better than SIM search). Figure 17 (a) also shows that weak-clustering appeared to have a greater impact on search effectiveness than over-clustering did. Over-clustering (e.g., increasing α from 3 to 5) did not significantly degrade search performance. This was likely due to the relatively small size of the network (1916 systems) in which clusters were not very far apart even with strong segmentation/clustering.

Different optimal α values have been reported in our research. In Ke and Mostafa [2010; Ke [2012], optimal α was found to be around 10 in a set of exact-match search experiments, in which neighbors were selected from a much smaller pool (of 150 random candidates). Among other factors, the pool size does have an important impact on network clustering. Even with the same α value, a larger pool (1000 candidates or more in this study) offers more choices and enables the alpha-based connectivity function to be more *selective*, leading to stronger network clustering.

We reason that the larger pool size we used in this research makes the α parameter a closer counterpart of the clustering exponent in Kleinberg [2000], which functions at the global level (i.e., pool size = network size). If what we observe here about α (roughly 2 to 3) is close to its *true* optimal value, then it suggests that the search space in this investigation is a space of 2 to 3 dimensions according to Kleinberg [2000] and Boguñá et al. [2009]. This is a very surprising finding because information retrieval often involves high dimensionality (of a feature space) and it is not intuitive to understand how two to three dimensions can be sufficient to represent various information collections in the space. Finding a good explanation and understanding the implication of this require closer scrutiny in future research.

10. CONCLUSION

We conducted experiments on decentralized IR operations on various scales of information networks and analyzed effectiveness, efficiency, and scalability of proposed search methods. Experimental results discussed in the above sections well support all of the four hypotheses on the *Clustering Paradox*, scalability of search, influence of degree distribution, and efficient search methods. We summarize major findings.

10.1. Clustering Paradox

Results provided evidence about the *Clustering Paradox* in the IR context and showed network structure was crucial to retrieval performance. In experiments on the ClueWeb09B collection, we found SIM search, one of the proposed methods that relied on similarity clues, achieved its best performance only at clustering exponent $\alpha = 2$ in larger-scale networks of $10,000$ and $100,000$ distributed systems. This level of network clustering appears to have supported a balance between strong ties and weak ties. While strong ties aid in creating local segments useful to guide searches, weak ties provide opportunities for searches to jump from one segment to another. Increasing or decreasing the level of network clustering shifts the balance and degrades search performance in effectiveness and efficiency.

This phenomenon of *Clustering Paradox* appeared in all of the experimented tasks, namely, relevance search, authority search, exact match (rare known-item search), and TREC'09 Web track tasks. It also appeared in networks of varied degree distributions. Statistical analyses in this study, extending findings from our earlier research[11] on different retrieval scenarios, add to the confidence about the generalization of this phenomenon in a broad range of retrieval contexts.

Our research has reported different optimal alpha values, which were in part due to varied random pool sizes we employed for neighbor selection and other factors related to differences in data and search tasks. The much larger pool size used in this study emplified the network clustering effect and made the α parameter a closer counterpart of the clustering exponent in Kleinberg [2000].

The discovery of optimal α at around 2 in major experiments of this study is a surprising, important finding. According to Kleinberg [2000; Kleinberg [2006a; Boguñá et al. [2009], decentralized searches in networks are optimal when clustering exponent α is equal to the dimensionality of the search space. An optimal α of 2 suggests low dimensionality of the search space in this study. However, IR systems have traditionally been associated with high dimensionality (e.g., of the feature space). The implication of this low dimensionality in large-scale IR remains to be further studied and understood.

10.2. Scalability of Findability

Examining the *Clustering Paradox* is crucial to understanding how search methods can scale in large information networks. We have found that search time can be well explained by a poly-logarithmic relation to network size at a specific level of network clustering. This poly-log relationship suggests a high scalability potential for searching in a continuously growing information space.

In our *exact match* (rare known-item search) experiments, search path length L (a surrogate for search time) was found to be proportional to $(\log N)^7$, where N is the number of systems in the network. The poly-logarithmic function was modeled on a wide range of network size scales $N \in [1, 10^2, 10^3, 10^4, 10^5]$ and showed a very large goodness of fit $R^2 = 0.999$. The exponent of the poly-log function was found to be 7,

[11] Our previous studies of different search tasks and on smaller scales can be found in Ke and Mostafa [2009; Ke and Mostafa [2010]. Ke [2012] provides an overview of the earlier results.

larger than 2 discovered in search experiments on non-IR tasks. We mainly focused on network clustering for search performance in the experiments and believe that a smaller exponent can be expected when other variables on decentralized searches are taken into account.

10.3. Scalability of Network Clustering

In addition to the scalability of decentralized searches, the network clustering function that supported very high effectiveness and efficiency of IR operations in large networks was found to be scalable as well. Clustering only involved local self-organization and required no global control – clustering time remained roughly constant, < 1 second, across the various network sizes $N \in [10^2, 10^3, 10^4, 10^5]$.

The clustering function required no "hard engineering" of the entire network but provided an organic way for systems to participate and interconnect given their opportunities and preferences. This highly scalable, local clustering mechanism is a useful feature to cope with dynamics in the distributed environment. When systems and contents evolve, the clustering mechanism can respond to changes locally without global propagation. That is, only neighbor systems directly connected to changed nodes will need to receive updates, eliminating a huge amount of network traffic that is otherwise needed. As shown by experimental results, this very local mechanism supports effective and efficient discovery of relevant information in the global space.

11. IMPLICATION AND FUTURE WORK

The exponential growth of digital information in large, dynamic environments poses great scalability challenges to today's information retrieval systems. To be able to find information effectively and efficiently in these environments such as the Web is a problem fundamental to numerous applications. The state-of-the-art Web search engines are not scalable because the data centers they rely on do not match and cannot grow proportionally to the growing world of information. This research presents an important direction toward alternative information retrieval architectures that have the potential to function better and scale more gracefully in the future. In the reported experiments, we focused on the impact of network structure on search performance and investigated a phenomenon we refer to as the *Clustering Paradox*, in which the topology of interconnected systems imposes a scalability limit. The findings show promises on decentralized search performance and offer insight into optimal network overlay for large-scale distributed IR.

This research represents an important step toward the development of decentralized retrieval systems in large, highly interconnected information spaces. Current experiments, however, were conducted on relatively stable environments. Further research on search in growing, dynamic (changing) settings is needed to show how such a search architecture can adapt and scale in real-world environments. In addition, the decentralized search architecture relies on individual distributed systems, which, in reality, exercise autonomy and self-interests. An information network cannot be assumed to be cooperative; besides the core challenge of search, issues such as incentive and trust should be examined [Kleinberg 2006b]. Systems are diverse in their information collections and retrieval functionality. Variations in search effectiveness of individual nodes will affect the overall utility of the decentralized retrieval architecture.

In the future, we plan to study system adaptation amid dynamic changes, retrieval effectiveness from real user perspectives, and additional system variables such as collection representation, ranking, and fusion functions in the context of large-scale distributed systems for information retrieval. We believe further investigation of additional structural factors as well as improved neighbor representation and search methods will further improve search scalability. Experiments at a million-distributed-system level (and higher) remain to be conducted in order to understand and demonstrate search efficiency on the web scale.

ACKNOWLEDGMENTS

The authors appreciate valuable discussions with Drs. Munindar P. Singh, José R. Pérez-Agüera, Michael Atwood, and Larry Alexander. We would like to thank anonymous ACM TOIS reviewers and associate editor Dr. Luo Si for highly constructive comments.

REFERENCES

Lada Adamic and Eytan Adar. 2005. How to search a social network. *Social Networks* 27, 3 (2005), 187 – 203. DOI:http://dx.doi.org/DOI:10.1016/j.socnet.2005.01.007

Lada A. Adamic, Rajan M. Lukose, Amit R. Puniyani, and Bernardo A. Huberman. 2001. Search in power-law networks. *Physical Review E* 64, 4 (Sep 2001), 046135. DOI:http://dx.doi.org/10.1103/PhysRevE.64.046135

Réka Albert and Albert-Lászlo Barabási. 2002. Statistical mechanics of complex networks. *Reviews of Modern Physics* 74, 1 (2002), 47–97. DOI:http://dx.doi.org/abs/cond-mat/0106096

Réka Albert, Hawoong Jeong, and Albert-Laszlo Barabási. 1999. Internet: Diameter of the World-Wide Web. *Nature* 401, 6749 (Sep 1999), 130–131. http://dx.doi.org/10.1038/43601

Javed A. Aslam and Mark Montague. 2001. Models for metasearch. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, New York, NY, USA, 276–284. DOI:http://dx.doi.org/10.1145/383952.384007

R. Baeza-Yates, C. Castillo, F. Junqueira, V. Plachouras, and F. Silvestri. 2007. Challenges on Distributed Web Retrieval. *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on* (April 2007), 6–20. DOI:http://dx.doi.org/10.1109/ICDE.2007.367846

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

Albert-Lászlo Barabási. 2009. Scale-Free Networks: A Decade and Beyond. *Science* 325 (July 24 2009), 412–413. http://www.sciencemag.org/cgi/content/full/325/5939/412

Christoph Baumgarten. 2000. Retrieving Information from a Distributed Heterogeneous Document Collection. *Information Retrieval* 3, 3 (2000), 253–271. DOI:http://dx.doi.org/10.1023/A:1026572910743

Mayank Bawa, Gurmeet Singh Manku, and Prabhakar Raghavan. 2003. SETS: search enhanced by topic segmentation. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. ACM, New York, NY, USA, 306–313. DOI:http://dx.doi.org/10.1145/860435.860491

Fabio Luigi Bellifemine, Giovanni Caire, and Dominic Greenwood. 2007. *Developing Multi-Agent Systems with JADE (Wiley Series in Agent Technology)*. John Wiley & Sons.

Matthias Bender, Sebastian Michel, Peter Triantafillou, Gerhard Weikum, and Christian Zimmer. 2005. Improving collection selection with overlap awareness in P2P search engines. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, New York, NY, USA, 67–74. DOI:http://dx.doi.org/10.1145/1076034.1076049

Michael W. Berry. 2004. *Survey of text mining: clustering, classification, and retrieval*. Springer.

Marián Boguñá, Dmitri Krioukov, and K. C. Claffy. 2009. Navigability of complex networks. *Nature Physics* 5, 1 (2009), 74 –80. DOI:http://dx.doi.org/10.1038/nphys1130

Jamie Callan. 2002. Distributed Information Retrieval. In *Advances in Information Retrieval*, W.Bruce Croft (Ed.). The Information Retrieval Series, Vol. 7. Springer US, 127–150. DOI:http://dx.doi.org/10.1007/0-306-47019-5_5

Jamie Callan and Margaret Connell. 2001. Query-based sampling of text databases. *ACM Transactions on Information Systems* 19, 2 (2001), 97–130. DOI:http://dx.doi.org/10.1145/382979.383040

James P. Callan, Zhihong Lu, and W. Bruce Croft. 1995. Searching distributed collections with inference networks. In *SIGIR '95: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, New York, NY, USA, 21–28. DOI:http://dx.doi.org/10.1145/215206.215328

Charles L. A. Clarke, Nick Craswell, and Ian Soboroff. 2009a. Overview of the TREC 2009 Web Track. In *Proc. of TREC-2009*. http://trec.nist.gov/pubs/trec18/papers/WEB09.OVERVIEW.pdf

Charles L. A. Clarke, Nick Craswell, and Ian Soboroff. 2009b. Overview of the TREC 2009 Web Track. In *The Eighteenth Text Retrieval Conference: TREC 2009*. Gaithersburg, MD. http://trec.nist.gov/pubs/trec18/papers/WEB09.OVERVIEW.pdf National Institute of Standards and Technology.

Brian F. Cooper and Hector Garcia-Molina. 2005. Ad Hoc, self-supervising peer-to-peer search networks. *ACM Transactions on Information Systems* 23, 2 (2005), 169–200. DOI:http://dx.doi.org/10.1145/1059981.1059983

Arturo Crespo and Hector Garcia-Molina. 2005. Semantic Overlay Networks for P2P Systems. In *Agents and Peer-to-Peer Computing*, Gianluca Moro, Sonia Bergamaschi, and Karl Aberer (Eds.). Lecture Notes in Computer Science, Vol. 3601. Springer Berlin Heidelberg, 1–13. DOI:http://dx.doi.org/10.1007/11574781_1

J. Dean and M. R. Henzinger. 1999. Finding Related Pages on the World Wide Web. *Computer Networks* 31, 11–16 (1999), 1467–1479.

Peter Sheridan Dodds, Roby Muhamad, and Duncan J. Watts. 2003. An Experimental Study of Search in Global Social Networks. *Science* 301, 5634 (2003), 827–829. DOI:http://dx.doi.org/10.1126/science.1081058

Christos Doulkeridis, Kjetil Norvag, and Michalis Vazirgiannis. 2008. Peer-to-peer similarity search over widely distributed document collections. In *LSDS-IR '08: Proceeding of the 2008 ACM workshop on Large-Scale distributed systems for information retrieval*. ACM, New York, NY, USA, 35–42. DOI:http://dx.doi.org/10.1145/1458469.1458477

Gudrun Fischer and André Nurzenski. 2005. Towards scatter/gather browsing in a hierarchical peer-to-peer network. In *P2PIR '05: Proceedings of the 2005 ACM workshop on Information retrieval in peer-to-peer networks*. ACM, New York, NY, USA, 25–32. DOI:http://dx.doi.org/10.1145/1096952.1096958

Gary William Flake, Steve Lawrence, C. Lee Giles, and Frans M. Coetzee. 2002. Self-Organization and Identification of Web Communities. *IEEE Computer* 35, 3 (2002), 66–71.

James C. French, Allison L. Powell, Jamie Callan, Charles L. Viles, Travis Emmitt, Kevin J. Prey, and Yun Mou. 1999. Comparing the performance of database selection algorithms. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, New York, NY, USA, 238–245. DOI:http://dx.doi.org/10.1145/312624.312684

James C. French, Allison L. Powell, Charles L. Viles, Travis Emmitt, and Kevin J. Prey. 1998. Evaluating database selection techniques: a testbed and experiment. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, New York, NY, USA, 121–129. DOI:http://dx.doi.org/10.1145/290941.290976

David Gibson, Jon Kleinberg, and Prabhakar Raghavan. 1998. Inferring Web communities from link topology. In *HYPERTEXT '98: Proceedings of the ninth ACM conference on Hypertext and hypermedia : links, objects, time and space—structure in hypermedia systems*. ACM, New York, NY, USA, 225–234. DOI:http://dx.doi.org/10.1145/276627.276652

Mark S. Granovetter. 1973. The Strength of Weak Ties. *Amer. J. Sociology* 78, 6 (May 1973), 1360–1380. DOI:http://dx.doi.org/10.1086/225469

Luis Gravano, Héctor García-Molina, and Anthony Tomasic. 1994. The effectiveness of GIOSS for the text database discovery problem. In *SIGMOD '94: Proceedings of the 1994 ACM SIGMOD international conference on Management of data*. ACM, New York, NY, USA, 126–137. DOI:http://dx.doi.org/10.1145/191839.191869

Erik Hatcher, Otis Gospodnetić, , and Michael McCandless. 2010. *Lucene in Action* (second edition ed.). Manning Publications. 475 pages. DOI:http://dx.doi.org/1933988177

David Hawking and Paul Thomas. 2005. Server selection methods in hybrid portal search. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, New York, NY, USA, 75–82. DOI:http://dx.doi.org/10.1145/1076034.1076050

Marti A. Hearst and Jan O. Pedersen. 1996. Reexamining the cluster hypothesis: Scatter/Gather on retrieval results. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM Press, New York, NY, USA, 76–84. DOI:http://dx.doi.org/10.1145/243199.243216

Michael N. Huhns. 1998. Agent Foundations for Cooperative Information Systems. In *In: Proc. s of the Third International Conference on the Practical Applications of Intelligent Agents and Multi-Agent Technology*, H.S. Nwana and D.T. Ndumu (Eds.). London.

Kalervo Jarvelin and Jaana Kekalainen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems* 20, 4 (2002), 422–446.

Nicholas R. Jennings. 2001. An agent-based approach for building complex software systems. *Commun. ACM* 44, 4 (2001), 35–41. DOI:http://dx.doi.org/10.1145/367211.367250

Nicholas R. Jennings and Michael J. Wooldridge. 1998. Applications of Intelligent Agents. In *Agent technology: foundations, applications, and markets*, Nicholas R. Jennings and Michael J. Wooldridge (Eds.). Springer-Verlag, Secaucus, NJ, USA, 3–28.

Weimao Ke. 2012. Decentralized Search and the Clustering Paradox in Large Scale Information Networks. In *Next Generation Search Engines: Advanced Models for Information Retrieval*, C. Jouis, I. Biskri, J.G. Ganascia, and M. Roux (Eds.). IGI Global, 29–46.

Weimao Ke and Javed Mostafa. 2009. Strong Ties vs. Weak Ties: Studying the Clustering Paradox for Decentralized Search. In *Proceedings of the 7th Workshop on Large-Scale Distributed Systems for Information Retrieval, co-located with ACM SIGIR 2009*. Boston, USA, 49–56.

Weimao Ke and Javed Mostafa. 2010. Scalability of findability: effective and efficient IR operations in large information networks. In *SIGIR '10: Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, New York, NY, USA, 74–81.

Weimao Ke, Cassidy R. Sugimoto, and Javed Mostafa. 2009. Dynamicity vs. Effectiveness: Studying Online Clustering for Scatter/Gather. In *SIGIR '09: Proceedings of the 32th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, New York, NY, USA, 19–26.

Jon Kleinberg. 2006a. Complex networks and decentralized search algorithms. In *In Proceedings of the International Congress of Mathematicians (ICM)*.

Jon Kleinberg. 2006b. Social networks, incentives, and search. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, New York, NY, USA, 210–211. DOI:http://dx.doi.org/10.1145/1148170.1148172

Jon M. Kleinberg. 2000. Navigation in a small world. *Nature* 406, 6798 (August 2000). http://dx.doi.org/10.1038/35022643

Jon M. Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew S. Tomkins. 1999. The Web as a Graph: Measurements, Models and Methods. *Lecture Notes in Computer Science* 1627 (1999), 1–17. citeseer.ist.psu.edu/kleinberg99web.html

David Liben-Nowell, Jasmine Novak, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. 2005. Geographic routing in social networks. *Proceedings of the National Academy of Sciences of the United States of America* 102, 33 (2005), 11623–11628. DOI:http://dx.doi.org/10.1073/pnas.0503018102

David Lillis, Fergus Toolan, Rem Collier, and John Dunnion. 2006. ProbFuse: a probabilistic approach to data fusion. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, New York, NY, USA, 139–146. DOI:http://dx.doi.org/10.1145/1148170.1148197

Jie Liu, Liang Feng, and Chao He. 2006. Semantic link based top-K join queries in P2P networks. In *Proceedings of the 15th international conference on World Wide Web (WWW '06)*. ACM, New York, NY, USA, 1005–1006. DOI:http://dx.doi.org/10.1145/1135777.1135987

Jie Lu. 2007a. *Full-text federated search in peer-to-peer networks*. Ph.D. Dissertation. Language Technologies Institute, Carnegie Mellon University. http://www.cs.cmu.edu/~jielu/Papers/thesis_jielu.pdf

Jie Lu. 2007b. Full-text federated search in peer-to-peer networks. *SIGIR Forum* 41, 1 (2007), 121–121. DOI:http://dx.doi.org/10.1145/1273221.1273233

Jie Lu and Jamie Callan. 2006. User modeling for full-text federated search in peer-to-peer networks. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, New York, NY, USA, 332–339. DOI:http://dx.doi.org/10.1145/1148170.1148229

Eng Keong Lua, Jon Crowcroft, Marcelo Pias, Ravi Sharma, and Steven Lim. 2005. A Survey and Comparison of Peer-to-Peer Overlay Network Schemes. *IEEE Communications Surveys and Tutorials* 7 (2005), 72–93.

Toan Luu, Fabius Klemm, Ivana Podnar, Martin Rajman, and Karl Aberer. 2006. ALVIS peers: a scalable full-text peer-to-peer retrieval engine. In *P2PIR '06: Proceedings of the international workshop on Information retrieval in peer-to-peer networks*. ACM, New York, NY, USA, 41–48. DOI:http://dx.doi.org/10.1145/1183579.1183588

R. Manmatha, T. Rath, and F. Feng. 2001. Modeling score distributions for combining the outputs of search engines. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, New York, NY, USA, 267–275. DOI:http://dx.doi.org/10.1145/383952.384005

Filippo Menczer. 2004. Lexical and semantic clustering by web links. *Journal of the American Society for Information Science and Technology* 55, 14 (2004), 1261–1269. DOI:http://dx.doi.org/10.1002/asi.20081

Weiyi Meng, Clement Yu, and King-Lup Liu. 2002. Building efficient and effective metasearch engines. *Comput. Surveys* 34, 1 (2002), 48–89. DOI:http://dx.doi.org/10.1145/505282.505284

Weiyi Meng and Clement T. Yu. 2010. *Advanced Metasearch Engine Technology*. Morgan & Claypool Publishers.

Stanley Milgram. 1967. Small-world Problem. *Psychology Today* 1, 1 (1967), 61–67.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1998. *The PageRank Citation Ranking: Bringing Order to the Web*. Technical Report. Stanford Digital Library Technologies Project. citeseer.ist.psu.edu/page98pagerank.html

Allison L. Powell and James C. French. 2003. Comparing the performance of collection selection algorithms. *ACM Transactions on Information Systems* 21, 4 (2003), 412–456. DOI:http://dx.doi.org/10.1145/944012.944016

Milad Shokouhi and Luo Si. 2011. Federated Search. *Foundations and Trends in Information Retrieval* 5, 1 (2011), 1–102.

Milad Shokouhi and Justin Zobel. 2007. Federated text retrieval from uncooperative overlapped collections. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, New York, NY, USA, 495–502. DOI:http://dx.doi.org/10.1145/1277741.1277827

Luo Si and Jamie Callan. 2003a. Relevant document distribution estimation method for resource selection. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. ACM, New York, NY, USA, 298–305. DOI:http://dx.doi.org/10.1145/860435.860490

Luo Si and Jamie Callan. 2003b. A semisupervised learning method to merge search engine results. *ACM Transactions on Information Systems* 21, 4 (2003), 457–491. DOI:http://dx.doi.org/10.1145/944012.944017

Luo Si and Jamie Callan. 2005. Modeling search engine effectiveness for federated search. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, New York, NY, USA, 83–90. DOI:http://dx.doi.org/10.1145/1076034.1076051

Özgürand Simsek and David Jensen. 2008. Navigating networks by using homophily and degree. *Proceedings of the National Academy of Sciences* 105, 35 (2008), 12758–12762. DOI:http://dx.doi.org/10.1073/pnas.0800497105

Munindar P. Singh, Bin Yu, and Mahadevan Venkatraman. 2001. Community-based service location. *Commun. ACM* 44, 4 (2001), 49–54. DOI:http://dx.doi.org/10.1145/367211.367255

Gleb Skobeltsyn, Toan Luu, Ivana Podnar Zarko, Martin Rajman, and Karl Aberer. 2007. Web text retrieval with a P2P query-driven index. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, New York, NY, USA, 679–686. DOI:http://dx.doi.org/10.1145/1277741.1277857

Chunqiang Tang, Zhichen Xu, and Sandhya Dwarkadas. 2003. Peer-to-peer information retrieval using self-organizing semantic overlay networks. In *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, New York, NY, USA, 175–186. DOI:http://dx.doi.org/10.1145/863955.863976

Paul Thomas and David Hawking. 2009. Server selection methods in personal metasearch: a comparative empirical study. *Information Retrieval* 12 (2009), 581–604. Issue 5. DOI:http://dx.doi.org/10.1007/s10791-009-9094-z

C. J. van Rijsbergen and K. Sparck-Jones. 1973. A test for the separation of relevant and non-relevant documents in experimental retrieval collections. *Journal of Documentation* 29, 3 (1973), 251–257.

Duncan Watts. 2003. *Six Degrees: The Science of a Connected Age*. W.W. Norton, New York.

Duncan J. Watts, Peter Sheridan Dodds, and M. E. J. Newman. 2002. Identity and Search in Social Networks. *Science* 296, 5571 (2002), 1302–1305. DOI:http://dx.doi.org/10.1126/science.1070120

Duncan J. Watts and Steven H. Strogatz. 1998. Collective dynamics of 'small-world' networks. *Nature* 393, 6684 (June 1998). http://dx.doi.org/10.1038/30918

Jinxi Xu and W. Bruce Croft. 1999. Cluster-based language models for distributed retrieval. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, New York, NY, USA, 254–261. DOI:http://dx.doi.org/10.1145/312624.312687

Bin Yu and Munindar P. Singh. 2003. Searching social networks. In *AAMAS '03: Proceedings of the second international joint conference on Autonomous agents and multiagent systems*. ACM, New York, NY, USA, 65–72. DOI:http://dx.doi.org/10.1145/860575.860587

Received November 2011; revised August, December 2012; accepted January 2013