Scalability of Findability: Decentralized Search and Retrieval in Large Information Networks

by
Weimao Ke

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the School of Information and Library Science.

Chapel Hill
2010

Approved by:

Dr. Javed Mostafa, Advisor

Dr. Diane Kelly, Reader

Dr. Gary Marchionini, Reader

Dr. Jeffrey Pomerantz, Reader

Dr. Munindar P. Singh, Reader

Abstract

WEIMAO KE: Scalability of Findability: Decentralized Search and Retrieval in Large Information Networks.
(Under the direction of Dr. Javed Mostafa.)

Amid the rapid growth of information today is the increasing challenge for people to survive and navigate its magnitude. Dynamics and heterogeneity of large information spaces such as the Web challenge information retrieval in these environments. Collection of information in advance and centralization of IR operations are hardly possible because systems are dynamic and information is distributed.

While monolithic search systems continue to struggle with scalability problems of today, the future of search likely requires a decentralized architecture where many information systems can participate. As individual systems interconnect to form a global structure, finding relevant information in distributed environments transforms into a problem concerning not only information retrieval but also complex networks. Understanding network connectivity will provide guidance on how decentralized search and retrieval methods can function in these information spaces.

The dissertation studies one aspect of scalability challenges facing classic information retrieval models and presents a decentralized, organic view of information systems pertaining to search in large scale networks. It focuses on the impact of *network structure* on search performance and investigates a phenomenon we refer to as the *Clustering Paradox*, in which the topology of interconnected systems imposes a scalability limit.

Experiments involving large scale benchmark collections provide evidence on the *Clustering Paradox* in the IR context. In an increasingly large, distributed environment, decentralized searches for relevant information can continue to function well only when

systems interconnect in certain ways. Relying on partial indexes of distributed systems, some level of network clustering enables very efficient and effective discovery of relevant information in large scale networks. Increasing or reducing network clustering degrades search performances. Given this specific level of network clustering, search time is well explained by a poly-logarithmic relation to network size, indicating a high scalability potential for searching in a continuously growing information space.

To Carrie and Lucy, with love

To the loving memory of my grandma

Acknowledgments

Serendipity is part of the journey of life. I came to the U.S. for a two-year master but found my passion for research after joining a walk with Dr. Javed Mostafa, now my advisor, who have guided me into a beautiful field known as Information Retrieval (IR). I cannot thank Dr. Mostafa enough for his constant guidance, support, encouragement, inspiration, and kindness over the years.

After an enjoyable transition from IT professional to IR researcher at Indiana University, I was very fortunate to join the doctoral program at SILS UNC and to have opportunities to interact with great researchers here. I would like to thank my committee members, Drs. Gary Marchionini, Diane Kelly, Jeffrey Pomerantz at SILS, and Dr. Munindar P. Singh at NC State University's Computer Science, who offered valuable guidance and important perspectives to help me develop as a scientist.

I would like to give special thanks to Dr. Katy Börner at Indiana University for her friendship, support, and guidance in areas related to information visualization and complex networks. I appreciate valuable help from faculty members and great support of the staff at SILS. I especially thank Dr. Paul Solomon for making my transition to UNC much easier.

I would like to thank many fellow students and friends in Indiana and in North Carolina for their friendship, company, and support, and for chances to come together and share ideas. A special thank you to Lilian and Ernest Laszlo for being always hospitable and encouraging. Thanks also to dear people and Dominican priests at

the St. Paul Catholic Newman Center in Bloomington for wisdom, guidance, and friendship.

I thank my parents for their support and patience during the years of my graduate study. Especially, I thank my mother for her unconditional love and trust. I thank my sisters for their care and support, in various ways. Thanks also go to my in-laws, especially my mother in law, for being here with my family.

I thank my dear late grandma, whose love endures so many years, for having shaped my personality and lived, in humble ways, best examples of integrity and diligence.

Finally, I owe tremendous gratitude to my loving family. My life has been so much more enjoyable and meaningful with the constant love of my wife Carrie and our sweet young lady Lucy. They are my source of energy in all of the work.

For all these, I thank God!

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

> An information retrieval system will tend *not* to be used whenever it is more painful and troublesome for a customer to have information than for him not to have it. – Mooers 1959 (see also Mooers, 1996)

Although often taken out of context, Mooers' law does relate to common frustrations with information. Amid the rapid growth of information today is the increasing challenge for people to survive and navigate in its magnitude. Having lots of information at hand is not necessarily helpful but often painful because it likely brings more overload than reward (Farhoomand and Drury, 2002). These problems have motivated research on intelligent *information retrieval*, automatic *information filtering*, and autonomous *agents* to help process large amounts of information and reduce a person's work (Belkin and Croft, 1992; Maes, 1994; Baeza-Yates and Ribeiro-Neto, 2004).

Traditional information retrieval (IR) systems operate in a centralized manner. They assume that information is on one side and the user on the other; and the problem is to match one against the other. As Marchionini (1995) recognized, retrieval implies an information object must have been "known" and those who "knew" it must have organized it for later being retrieved by themselves or others. However, figuring out who has what information is not straightforward as we are all dynamically involved in

the consumption and creation of information. It is widely observed that information is vastly distributed – before matching and ranking operations lays the question of where relevant information collections are (Gravano et al., 1999; Callan, 2000; Bhavnani, 2005; Morville, 2005).

We live in a distributed networked environment, 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 (Paepcke et al., 1998). There is no single global information repository.

Advances in computing technologies have enabled efficient collection (e.g., crawling), storage, and organization of information from distributed sources. However, there is a growing space on the Web where information is difficult to aggregate and make available to public. Research has observed that much valuable information was not published online for reasons such as privacy, copyright, and unwillingness to share to the public (Kautz et al., 1997b; Yu and Singh, 2003; Mostafa, 2005). More critically, five hundred times larger than the indexable Web is some hidden space called *deep web* where information is publicly available but cannot be easily crawled (Mostafa, 2005; He et al., 2007). Sites on the *deep web* often have large databases behind their interfaces and provide information only when properly queried. Sometimes, information is so *fresh* that storing it for later being found is useless – it might become outdated hours, if not seconds, after being produced, e.g., for information about stock prices or current weather conditions.

The *deep web* represents a large portion of the entire web that requires various levels of intelligent interactions, challenging for search engines to penetrate. Research has been done on the problem but solutions remain ad hoc. Researchers rely on existing search terms and/or visible contents to guess what keywords can be used to activate hidden information in *deep web* databases. However, this is not a general solution. For any database behind the scene, there are simply too many possibilities to guess – not to mention the fact that there are at least half million different databases/sites and more than one million interfaces[1] on the *deep web* (He et al., 2007)[2]. Moreover, the problem goes beyond what query terms should be used – you also need to "speak" in ways deep web systems understand. For example, orbitz.com[3] will not take your query if you simply enter "I need a flight from New York to London on Tuesday." Instead, you will need to speak in Orbitz's language – to specify the different elements in an acceptable query structure and provide the values. The variety of languages is an immense challenge and "learning them all" is not an option. And given the evolutionary nature of the Web, it is unrealistic for one to implement communication channels to all.

Because of the distributed nature of information and the size, dynamics, and heterogeneity of the Web, it is extremely challenging, if not impossible, to collect, store, and process all information in one place for retrieval operations. Centralized solutions will hardly survive – they are are vulnerable to scalability demands (Baeza-Yates et al., 2007). No matter how much can be invested, it will remain a mission impossible to

[1]One site or database can have multiple interfaces. For example, some offer both free text search and "advanced" search options while others use various facets for their search interfaces, e.g., to find a car by "region" and "price" or by "make" and "model."

[2]The numbers of deep web databases and interfaces have been growing over the years.

[3]Orbitz is a commercial web site for travel scheduling, e.g., to book flights and hotels.

replicate and index the entire Web for search. The *deep web*, hidden from the indexable surface, further challenges existing search systems. For the search service market, barriers to entry are so high that competition is only among the few. Are today's search engine giants good enough to serve our information needs? Before this could be answered, how current models for search would survive the continuous growth of the Web is another legitimate question.

As the Web continues to evolve and grow, Baeza-Yates et al. (2007) reasoned that centralized IR systems are likely to become inefficient and fully distributed architectures are needed. Even when one has sufficient investment to provide a "one for all" search service on the Web, the architecture will never remain centralized – it will be forced to break down into distributed and/or parallel computing machines given that no single machine can possibly host the entire collection. For example, it was estimated that today' search engine giant Google[4] had about a half million computers behind its services (Markoff and Hansell, 2006), a relatively significant proportion to the 60 million stable Internet-accessible computers projected by Heidemann et al. (2008). In another word, for every hundred stable Internet-accessible computers in the Internet, there is one Google machine[5]. Baeza-Yates et al. (2007) estimated that, by 2010, a Web search engine will need more than one million computers to survive. Even so, how to manage them in a distributed manner for efficiency will remain a huge challenge.

More importantly, however, we have to know potential alternative techniques and better methods to support searches in a less costly way. A potential candidate is to take advantage of the existing computing infrastructure of the Internet and invent

[4]Twelve years from now, it might become less relevant, if not irrelevant, to talk about Google – just as it has become less relevant to talk about Alta Vista now than it was a dozen years ago. But for the sake of discussions in today's context, Google will continue to be used as a well recognized search engine example.

[5]Note that not all Google machines were Internet-accessible and they were not necessarily a subset of the 60 million. Neither is it likely that Google used all the half million for search services

new strategies for them to work together and help each other search. Recent years have witnessed the large increase of personal and organizational storage in response to the fast growth of information. Yet the distributed network of computing machines (i.e., the Internet), with an increasing capacity collectively, have not been sufficiently utilized to facilitate search. Using distributed nodes to share computational burdens and to collaborate in retrieval operations appears to be reasonable.

Research on complex networks shows promises as well. It has been discovered that small diameters, or short paths between members of a networked structure, were a common feature of many naturally, socially, or technically developed communities – a phenomenon often known as *small world* or *six degrees of separation* (Watts, 2003). Early studies showed that there were roughly six social connections between any two persons in the U.S. (Milgram, 1967). 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).

In addition, 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, 2000b; 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 very likely a few degrees (connections) away from the one who needs it and is potentially findable. This provides potentials for distributed algorithms to traverse such a network to find it efficiently. However, this is never an easy task because not only desired information items or documents are a few degrees away but so are all documents. The question is how people, or intelligent information systems on behalf of them, can learn to follow shortcuts to relevant information without being lost in the hugeness of a networked

environment (e.g., the Web).

Dynamics and characteristics of a network manifest the way it has been formed by members with individual objectives, capacities, and constraints (Amaral et al., 2000). All this is a display of how members of a society have survived and will continue to scale collectively. To take advantage of a network is to potentiate a capacity potentially far beyond the linear sum of all as the (communicative) value of a network is said to grow proportionately to the square of its size in terms of Metcalfe's law (Ross, 2003). These networks, developed under constraints, were also found to demonstrate useful substructures and some topical gradient that can be used to guide efficient searches (Kleinberg et al., 1999; Watts et al., 2002; Kleinberg, 2006a).

1.1 Problem Statement

Dynamics and heterogeneity of a large networked information space (e.g., the Web) challenge information retrieval in such an environment. Collection of information in advance and centralization of IR operations are hardly possible because systems are dynamic 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).

While an information need may arise from anywhere in the space (from an agent or a connected peer), relevant information may exist in certain segments but there requires a mechanism to help the two 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 travel 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 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 millions of sites and trillions of documents, either visibly or invisibly. And considering the dynamics and heterogeneity, it is impossible to implement and maintain communication channels to all – that is why *deep web* remains a problem unsolved.

1.1.1 Scalability of Findability

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*[6] (nodes or peers) and E is the set of all edges or connections among the agents. On behalf of their principals, 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 a query being properly formulated and submitted. Still, some provide information that is time sensitive and therefore useless to be collected beforehand.

Being information providers, agents also represent information seekers. Imagine an agent in the network, say, A_u, has an information need (i.e., receives a request from a user) and formulates a query for it. Suppose another agent A_v, somewhere in

[6]For the discussion here, an *agent* is seen as a computer program or system that either provides or seeks information, on behalf of its human or organizational principal. The term will be defined more formally in Section 4.

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 question is:

Problem 1 *Findability: Can agents 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?*

A constraint here is that the network should not be troubled too much for each query. One can reasonably propose a simple solution to the problem above through flooding or breadth first search. However, flooding may achieve findability at the cost of coverage – it will reach a significant proportion of all agents in the network for a single query. Even if each agent issues one query a day, there will be too much traffic in the network and huge burden on other agents. This type of solutions will not scale[7]. We should therefore seek a balance between findability and efficiency:

Problem 2 *Efficiency of Findability: 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,

Problem 3 *Scalability of Findability: 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?*

[7]Here is a simple calculation of flooding scalability. In a network of 10 agents, if each agent submits a query that reaches half of the network, then every agent will have to process 5 queries on average. If the network size increases to one million, then every agent will have to take half million queries under flooding.

Small world networks such as the World Wide Web, as research has found, usually have a small diameter[8] on a logarithmic scale of network size (Albert et al., 1999). Experimental simulations on abstract models for network navigation, for example, achieved findablity through short path lengths bounded by $c(\log N)^2$, where c is a constant and N the network size (Kleinberg, 2000a). A goal of the literature review is to (hopefully) find an IR research direction for a logarithmic function of information findability.

Another related goal is to develop improved distributed IR systems by analyzing the impact of network characteristics on findability of information. The broad aim is to clarify the relationship of critical IR functions and components to characteristics of distributed environments, identify related challenges, and point to some potential solutions. The survey will draw upon research in information retrieval and filtering, peer-to-peer search and retrieval, complex networks, and multi-agent systems as the core literature.

1.2 Significance

Shapiro and Varian (1999) discussed the value of information to different consumers and reasoned that information is costly to create and assemble "but cheap to reproduce" (p. 21). In addition, finding relevant information to be replicated or used is likewise costly. Without a global repository, it is difficult to know about where specific information is. Quickly locating relevant information in a distributed networked environment is critical in the information age.

From a communication perspective, Metcalfe asserted that the value of a network grows proportionately to the square of its size, or the number of users connected to it

[8]A network diameter refers to the longest of all shortest pairwise path lengths.

(Shapiro and Varian, 1999; Ross, 2003). Searching distributed collections of information through collective intelligence of networked agents inherits the "squared" potential and has important implications in IR as well as in Information Science. Applications of information findability in networks include, but are not limited to, search and retrieval in peer-to-peer networks, intelligent discovery of (deep) web services, distributed desktop search, focused crawling on the Web, agent-assisted web surfing, and expert finding in networked settings.

Finding relevant information through a peer-to-peer (P2P) or online social network (e.g., facebook.com) is an obvious application. Another type of application, in the Semantic Web, is to build information agents through which queries can be directed efficiently to relevant services and databases. For example, one who needs to book an air ticket but does not know the existence of Orbitz can activate his software agent to send the query to connected others, who collectively carry the query forward to and results back from Orbitz through all intermediaries. We can also implement intelligent web browser assistants to help navigate through hyperlinks to find relevant web sites and/or pages.

From the perspective of search and discovery on the Web, efficient navigation in networks for information retrieval carries challenges as well as opportunities. A brief discussion follows.

A Broadened Searchable Horizon

In the past decade, we have seen the increased popularity of information retrieval systems, particularly web search engines, as useful tools in people's daily information seeking tasks. Although many enjoy, and some boast, the boosted findability on the Web, there is a significant portion of it too "hidden" or too "deep" to be found. An ideal distributed networked retrieval system, nonetheless, will allow deep sites to be reached

and hidden information to be found through efficient collective routing of queries by intermediary peers/agents.

Despite taking a different view on the problem of search, a distributed approach to information retrieval should not be seen as a replacement of current search systems such as Google. It can become part of a current system, e.g., for Google to deal with large collections distributed internally. In this way, a distributed architecture is an approach to scalability for current IR systems. On the other hand, a traditional system can also be seen as part of the distributed architecture, where Google, for instance, is a super-node/agent. With the integration of both search paradigms, the entire system will provide a broadened horizon for search on the Web.

Finding Information Alive

"Information is like an oyster: it has its greatest value when fresh." (Shapiro and Varian, 1999, p. 56) If crawler-based search systems can be seen as museums, which make copies of (and obviously not every piece of) information on the Web, then it will be desirable for people to go to the wild of the Web to find information alive. The idea of going to the wild is to chase information out to catch it – just like how we chase butterflies – which retrieval systems such as Google were not born to be. There are so many sites and databases that cannot be crawled in advance and stored statically. Answers are not there until questions are asked; information is query driven and often transient. A distributed search architecture will potentially allow people's live queries to travel a short journey in a huge network to chase hidden information out, fresh.

Chapter 2

Literature Review

The problem concerning how information can be quickly found in networked environments has become a critical challenge in Information Retrieval (IR), particularly for IR systems on the Web – a challenge that deserves further investigation from an Information Science perspective. To attack the challenge, nonetheless, will draw on inspirations, proposals, and known principles from multiple disciplines. With the problems of information findability and scalability of findability in mind, this literature review aims to survey the literature in information science (and particularly information retrieval), complex networks, multi-agent systems, and peer-to-peer content distribution and search.

Section 2.1 starts with a brief discussion on the notion of information in this survey (i.e., what is to be found when the survey talks about information findability), reviews the broad research area of *information retrieval* (IR), and discusses some of the basic problems and models. Section 2.2 moves on to information retrieval on the Web and introduces major challenges, solutions, and related areas including *distributed IR*. Further decentralization of distributed IR leads to Section 2.3 on *peer-to-peer* information retrieval, an area where the problem of finding information in networks has a very tangible meaning. Section 2.4 surveys multiple research fronts studying characteristics

and dynamics of *complex networks*, and discusses, in their basic forms, the challenge of findability in *small world*. Finally, Section 4 introduces the notion of *agent* and uses the *multi-agent system* paradigm to revisit the raised IR problems. The literature review concludes with a summary of main points and unanswered questions in Section 2.6.

2.1 Information Retrieval

Information Science is about "gathering, organizing, storing, retrieving, and dissemination of information" (Bates, 1999, p. 1044), which has both science and applied science components. In this survey, framing the problem as finding information in networks requires a clear definition of what information is, or what is to be found. In the literature, however, proposals on defining information abound without broad consensus. Information has been related to uncertainty (Shannon, 1948), form (Young, 1987), structure (Belkin et al., 1982), pattern (Bates, 2006), thing (Buckland, 1991), proposition (Fox, 1983), entropy (Shannon, 1948; Bekenstein, 2003), and even physical phenomena of mass and energy (Bekenstein, 2003). Information is so universal that, as Bates (2006) acknowledged, almost anything can be experienced as information and there is no unambiguous definition we can refer to.

In Saracevic's (1999) terms, there are three senses of information, from the narrow to broader to the broadest sense, used in disciplines such as information science and computer science. The narrow sense is often associated with messages and probabilities ready for being operationalized in algorithms. This particular survey is interested in information that is created, replicated, and transferred in electronic environments, or digital information that is contained in documents. It is in the sense of information associated with digital messages that intelligent information retrieval systems or software agents can be designed, implemented, tested, and used (Saracevic, 1999). Hence, a pragmatic approach, namely the information-as-document approach, is taken to define the scope of discussions in this survey. To be specific, the literature review is interested in the finding of digital information in the form of text documents unless stated otherwise.

Mooers (1951) coined the term *information retrieval* to refer to the investigation of information description and specification for search and techniques for search operations

(see also Saracevic, 1999). As one of the core areas in information science, information retrieval (IR) studies the representation, storage, organization, and access to information items, and is concerned with providing the user with easy access to the information he is interested in (Baeza-Yates and Ribeiro-Neto, 2004). System-centric IR, influenced by computer science, has a focus on studying the effects of system variables (e.g., representation and matching methods) on the retrieval of relevant documents (Saracevic, 1999).

It has long been recognized that system-centric IR and user-centric Information Seeking (IS)[1] are independent research areas (Vakkari, 1999; Ruthven, 2005). While IR research outcomes have become widely adopted well-known due to the development of the World Wide Web and search engines, wider aspects than models and algorithms of IR are resistant to being studied in laboratory settings. Robertson (2008) argued that IR should be heading toward a direction where richer hypotheses – other than the only form of "whether the model makes search more effective" – are tested.

2.1.1 Representation and Matching

The mainstream research in IR falls in the category of partial match, as opposed to exact or boolean match (Belkin and Croft, 1987). A classic IR model is illustrated in Figure 2.1, in which an IR system is to find (partially) matched IR documents given a query (representative of an information need). Researchers have tried to classify IR research by using various facets such as browsing vs. retrieval, formal vs. non-formal methods, and probabilistic vs. algebraic and set theoretic models, etc. (Baeza-Yates and Ribeiro-Neto, 2004; Jarvelin, 2007). Among the subcategories, the formal or classic methods, which include probabilistic models and the vector space

[1]The broader processes of Information Retrieval (IR) and Information Seeking (IS) are largely overlapped (Vakkari, 1999). Here, the concepts of user-centric IR and user-centric IS are exchangeable, as opposed to IR or system-centric IR.

model, have been widely followed and experimented on (Sparck Jones, 1979; Robertson, 1997; Salton et al., 1975).

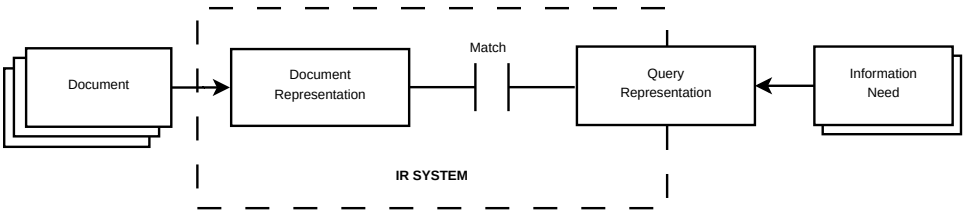


Figure 2.1: Classic Information Retrieval Paradigm, adapted from Bates (1989)

The probabilistic model follows a proposed probability principle in IR (Robertson, 1997), which is to rank documents for the maximal probability of user satisfaction, and use the principle to guide document representation, e.g., term weighting (Sparck Jones, 1979). The probabilistic model has a strong theoretical basis for guiding retrieval toward optimal relevance and has proved practically useful. However, among other disadvantages, early probabilistic models only dealt with binary term weights and assumed the independence of terms. In addition, it is often difficult to obtain and/or to estimate the initial separation of relevant and irrelevant documents.

To overcome limitations of binary representation and make possible accurate partial matching, Salton et al. (1975) proposed the Vector Space Model (VSM) in which queries and documents are represented as n-dimensional vectors using their non-binary term weights (see also Baeza-Yates and Ribeiro-Neto, 2004). In the dimensional space for IR, the direction of a vector is of greater interest than the magnitude. The correlation between a query and a documents is therefore quantified by the cosine of the angle between the two corresponding vectors. VSM succeeded in its simplicity, efficiency, and superior results it yielded with a good variety of collections (Baeza-Yates and Ribeiro-Neto, 2004).

Terms can be used as dimensions and frequencies as dimensional values in VSM. Yet a more widely used method for term weighting is *Term Frequency * Inverse Document*

Frequency (*TF*IDF*), which integrates not only a term's frequency within each document but also its frequency in the entire representative collection (Baeza-Yates and Ribeiro-Neto, 2004). The reason for using the *IDF* component is based on the observation that terms appearing in many documents in a collection are less useful. In the extreme case, useless are stop-words such as "the" and "a" that appear in every English document.

The early tradition of Cranfield[2] has had great influence on how IR research is conducted as an experimental science (Cleverdon, 1991; Saracevic, 1999; Robertson, 2008). The Text REtrieval Conference (TREC), as a platform where IR systems can be more "objectively" compared, continues the system-centric tradition. TREC aims to support IR research by providing the infrastructure necessary for large-scale evaluating of text retrieval methodologies, which includes benchmark collections, pre-defined tasks, common relevance bases, and standardized evaluation procedures and metrics (Voorhees and Harman, 1999).

Of various evaluation metrics used in TREC and IR, *precision* and *recall* are the basic forms. Whereas *precision* measures the fraction of retrieved documents being relevant, *recall* evaluates the fraction of relevant documents being retrieved. IR research has extensively used precision, recall, and their derived measures for system evaluations. For system comparison, techniques such as precision-recall plots, the F measure (or the harmonic mean of precision and recall), the E measure, and ROC are often adopted.

With the inverse relationship of precision and recall (Cleverdon, 1991), research has found recall difficult to scale. Not only is a thorough recall base (e.g., a complete human-judged relevant set) hard to establish when the collection size grows, so

[2]The Cranfield tests refer to a series of early experiments, led by Cyril W. Cleverdon at College of Aeronautics at Cranfield, on retrieval effectiveness (or efficiency then) of index languages/techniques. Prototypical IR experimental setup (e.g., a common query set and relevance judgment) and evaluation metrics such as recall and precision were established and have since been widely used. One important finding from the experiments, surprisingly then, was the superiority of single-term-based index over phrases (Cleverdon, 1991).

is high recall difficult to achieve with large collections. When Blair and Maron (1985) conducted a longitudinal study to evaluate retrieval effectiveness of legal documents, only high precisions and low recalls were achieved, unsatisfactory for lawyers looking for thoroughness. It was perhaps premature for Blair and Maron (1985) to conclude on the inferiority of automatic IR and Salton (1986) later dismissed their conclusion through a systematic comparison.

One approach to improving recall is through identifying similar documents to the relevant retrieved document set. Clustering, through the aggregation of similar patterns, have some potential (Jain et al., 1999; Han et al., 2001). As the Cluster Hypothesis states, relevant documents are more similar to one another than to non-relevant documents (van Rijsbergen and Sparck-Jones, 1973). Hence, relevant documents will cluster near other relevant documents and they tend to appear in the same cluster(s) (Hearst and Pedersen, 1996). Research also discovered that, in various information networks (e.g., WWW), similar nodes (e.g., Web pages) tend to connect to each other and form local communities (Gibson et al., 1998; Kleinberg et al., 1999; Davison, 2000; Menczer et al., 2004). When a relevant document is reached, more can potentially be retrieved.

2.1.2 Relevance

As an IR investigation, this survey is concerned with the retrieval of "relevant" information for the user. *Relevance* is a key notion in IR that drives its objectives, hypotheses, and evaluations, and deserves a good understanding. However, the meaning of *relevance* is usually ambiguous while its sufficiency across domains is questionable. According to Anderson (2006), relevance remains one of the least understood concepts in IR.

Research has studied and debated over the concept of relevance. Although consensus is lacking, researchers do share some common views of relevance as being dynamic and situational, depending on the user's information needs, objectives, and social context (Chatman, 1996; Barry and Schamber, 1998; Chalmers, 1999; Ruthven, 2005; Anderson, 2006; Saracevic, 2007). Ruthven (2005) reasoned that relevance is "subjective, multidimensional, dynamic, and situational" (p. 63). It is not simply "topical" as commonly assumed by system-centric IR research using standardized collections as in TREC tracks, in which relevance was predetermined by other people.

In system-centric IR, the reassessment of relevance and interpretations are rarely scrutinized. Research simplifies the concept and focuses on its "engineerable" component by ignoring its broader context. As Anderson (2006) noted, relevance judgments merely based on topicality do not incorporate multiple factors underlying a user's decision to pursue or use information. Nonetheless, as he pointed out, topical relevance is widely used in IR "because of its operational applicability, observability, and measurability" (Anderson, 2006, p. 8).

It is true that topical relevance is too simplistic and that the static view of information needs is problematic. And it makes sense to incorporate contextual variables in order to approach the real meaning of relevance in situation. Unfortunately, according to Saracevic (1999), "in most human-centered [IR] research, beyond suggestions, concrete design solutions were not delivered" (p. 1057). Research on retrieval algorithms often assumes topicality of relevance to make progress on the system side while leaving user issues for further investigation.

2.1.3 Searching and Browsing

Searching and browsing represent two basic paradigms in information retrieval. While searching requires the user to articulate an information need in query terms understandable by the system, browsing allows for further exploration and discovery of information. The two techniques work differently and often operate separately; sometimes, however, they become more useful when combined.

Bates (1989) argued that the classic IR model, as illustrated in Figure 2.1, offered a rigid, system-oriented, and single-session approach to searching and should take into account other forms of interaction so that users could express their needs directly. An alternative retrieval paradigm, namely, the *berrypicking* search, was proposed to accommodate more dynamic information exploration and collection activities over the course of an evolving search (Bates, 1989). Today's hypertext environments, e.g., the WWW or any network (e.g., wikipedia) connecting documents from one another, can support *berrypicking* searching very well as one can easily "jump" in the wired space during browsing.

Similar to the *berrypicking* approach to browsing and finding information in the evolving dynamics of information needs is the Information Foraging theory in which "information scent" can be followed for seeking, gathering, and using on-line information (Pirolli and Card, 1998). The recognition of various information seeking and retrieval scenarios involving lookup, learning, and investigative tasks have motivated a new research thread in exploratory search (Marchionini, 2006; White et al., 2007b).

As an example for interactive searching and browsing, Scatter/Gather is well known for its effectiveness in situations where it is difficult to precisely specify a query (Cutting et al., 1992; Hearst and Pedersen, 1996). It combines searching and browsing through iterative gathering and re-clustering of user-selected clusters. In each iteration, the system scatters a dataset into a small number of clusters/groups and presents short summaries

of them to the user. The user can select one or more groups for further examination. The selected groups are then gathered together and clustered again using the same clustering algorithm. With each successive iteration the groups become smaller and more focused. Iterations in this method can help users refine their queries and find desired information from a large data collection.

Researchers have studied the utility of Scatter/Gather to browse retrieved documents after query-based searches. It was found that clustering was a useful tool for the user to explore the inherent structure of a document subset when a similarity-based ranking did not work properly (Hearst et al., 1995). Relevant documents tended to appear in the same cluster(s) that could be easily identified by users (Hearst and Pedersen, 1996; Pirolli et al., 1996). It was also shown that Scatter/Gather induced a more coherent view of the text collection than query-based search and supported exploratory learning in the search processes (Pirolli et al., 1996; Ke et al., 2009). Being interactive and flexible, the Scatter/Gather modality has also been applied to browsing large text collections distributed in a hierarchical peer-to-peer network (Fischer and Nurzenski, 2005).

2.1.4 Conclusion

According to Salton (1968), information retrieval (IR) is about the "structure, analysis, organization, storage, searching, and retrieval of information." Over the past decades, however, information retrieval research has been focused on matching and retrieval rather than searching and finding. Morville (2005) defined *findability* as one's ability to navigate a space to get desired information. Whereas *retrieval* and *findability* are highly associated, IR has traditionally assumed that all information (and collections of it) can be navigated to and found. Findability is less an issue given a well-defined scope for retrieval, when information is collected and stored in a known repository (Marchionini,

1995). Rarely is it a question where information collections are or whether relevant information is yet to be located. These questions, however, are critical for searching in a large, heterogeneous space such as the Web, especially the *deep web*, where global information about individual collections does not exist. Solutions are needed for various systems to work together in the absence of a global repository. With this, the survey will now shift to information retrieval on the Web and discuss various challenges, solutions, and problems that remain to be solved.

2.2 Information Retrieval on the Web

With large volumes of information, challenges for information retrieval on the Web also include data (or information) being highly distributed and heterogeneous, sometimes volatile, and of different quality (Bowman et al., 1994; Brown, 2004; Baeza-Yates and Ribeiro-Neto, 2004). All these have important implications on IR operations for information collection (crawling), indexing, matching, and ranking.

2.2.1 Web Information Collection and Indexing

Most Web search engines use crawlers, which can be seen as software agents, to traverse the Web through hyperlinks to gather pages that will later be indexed on main servers. Provided the size of the Web and its continuous growth, multiple crawlers and indexers are usually employed in parallel to do the tasks more efficiently. The coordination of the operations, however, has become a significant challenge. To this end, Bowman et al. (1994) developed an architecture in which *gatherers* and *brokers* focused on individual topics, interacted, and cooperated with one another for data collection, indexing, and query processing.

While a centralized index can hardly scale on the Web, Melnik et al. (2001), for example, presented a distributed full-text indexing architecture that loaded, processed, and flushed data in a pipelined manner. It was shown that the distributed system, with the integration of a distributed relational database for index creation and management, effectively enabled the collection of global statistics such as *IDF* values of terms. In recent years, the demand for large scale data processing has increased dramatically in order to index, summarize, and analyze large volumes of Web pages on large clusters of computers. MapReduce represents one of the parallel computing paradigms for this purpose and has been extensively used by Google (Dean and Ghemawat, 2008).

Various crawler techniques have been developed over the years for collection efficiency and effectiveness , duplicate reduction, focused/topical crawling, and intelligent updates (Cho et al., 1998; Chakrabarti et al., 2002; Menczer et al., 2004; Fetterly et al., 2008). Different strategies were proposed for crawling special web sites such as blogs and forums (Wang et al., 2008). Guidelines were also developed to design better crawler (robot) behavior. However, there is a large portion of the Web, the so-called deep Web, resistant to being crawled easily.

While Gulli and Signorini (2005) estimated that there were more than 11.5 billion indexable Web pages, of which Google was found to index nearly 70% (the largest compared to Ask, Yahoo!, and MSN), the deep (or invisible) Web is said to have more than half million sites and approximately seven petabyte[3] data, 500 times larger than the indexable Web (Mostafa, 2005; He et al., 2007). Pages on the deep Web represent dynamic systems that can only be activated through intelligent interactions, e.g., with the use of proper query terms (Baeza-Yates and Ribeiro-Neto, 2004).

Current solutions primarily rely on available user queries, term predictions, and HTML form parsers to interact with deep Web systems for collecting information from there. Although deep web entrances are easy to reach, they are diverse in topics and structures (He et al., 2007). Only a small percentage is covered by central deep Web directories. To build a centralized system to search on all deep Web sites is doomed to fail because there is no global information about where they are and how they interact. Even if there is such information, implementation of communication channels to all deep Web sites remains practically impossible.

[3] 1 petabyte = 1024 terrabytes = 1024×1024 gigabytes $\approx 10^{15}$ bytes.

2.2.2 Link-based Ranking Functions

Classic IR methods provide the foundation for information retrieval on the Web. Most text-based methods for representation, matching, and ranking can be applied to Web IR (Rasmussen, 2003; Yang, 2005). While searching and browsing are useful paradigms, precision- and recall-based evaluation metrics remain, to some extent, applicable. However, some traditional IR assumptions no longer hold. Ranking Web documents merely based on textual contents does not suffice because web pages created by diverse individuals and organizations, different from a traditional homogeneous environment, are of varied quality levels.

The Web is rich not only in its content but also in its structure (Yang, 2005). Particularly, information is captured not only in texts but also in hyperlinks that collectively construct paths for the user to surf from one page to another. Additional structures such as click-throughs carry implicit clues about what might be relevant to the user's interests. Link-based methods have been widely used by information retrieval systems on the Web.

Techniques for link-based retrieval originated from research in bibliometrics which deals with the application of mathematics and statistical methods to books and other media of written communication (Nicolaisen, 2007). The quantitative methods offered by bibliometrics have been used for literature mining and enabled some degree of objective evaluations of scientific publications, offering answers to questions about major scholars and key areas within a discipline (Newman, 2001a,b).

Link analysis based on citations, authorships, and textual associations provides a promising means to discover relations and meanings embedded in the structures (Nicolaisen, 2007). Despite bias, the use of citation data has proved effective beyond an impact factor in bilbiometrics (Garfield, 1972). Its application in information retrieval has brought new elements to the notion of relevance and produced promising results.

For example, Bernstam et al. (2006) defined importance as an article's influence on the scientific discipline and used citation analysis for biomedical information retrieval. They found that citation-based methods, as compared with content-based methods, were significantly more effective at identifying important articles from Medline.

Besides direct citation counting, other forms of citation analysis involve the methods *bibliographic coupling* (or co-reference) and *co-citation*. While bibliographic coupling examines potentially associated papers that refer to a common literature, co-citation analysis aims to identify important and related papers that have been cited together in a later literature. These techniques have been extended to identify key scholars, groups, and topics in some fields (White and Mccain, 1998; Lin et al., 2003).

In citation analysis, there is no central authority who judges each scholar's merit. Instead, peers review each others' works and cite each other and all this forms the basis for evaluation of scholarly productivity and impact. Authorities might emerge but they come from the democratic process of distributed peer-based evaluations without centralized control.

Similar patterns are exhibited on the World Wide Web where highly distributed collections of information resources are served with no central authorities. Information quality is unevenly maintained provided the heterogeneity. It is challenging to define and measure information quality and relevance merely based on textual contents. Hyperlinks on the Web provide additional clues and are often treated as some indication of a page's popularity and/or importance – similar to the evaluation of citations for scholarly impact. Hence, citation analysis traditionally used in bibliometrics was adopted by IR researchers for ranking web pages.

Although web pages and links are created by individuals independently without global organization or quality control, research has found regularities in the use of text and links. According to Gibson et al. (1998), the Web exhibited a much greater degree

of orderly high-level structure than was commonly assumed. Link analysis confirmed conjectures that similar pages tend to link from one to another and pages about the same topic will be clustered together (Menczer, 2004).

Among link-based retrieval models on the Web, *PageRank* and *HITS* are well known. Page et al. (1998) proposed and implemented *PageRank* to evaluate information items by analyzing collective votes through hyperlinks. Page et al. (1998) reasoned that simple citation counting does not capture varied importance of links and used a propagation mechanism to differentiate them. The process was similar to a random Web surfer clicking through successive links at random, with a damping factor to avoid loops. As experiments showed, *PageRank* converged after 45 iterations on a dataset of more then three hundred million links. It effectively supported the identification of popular information resources on the Web and has enabled Google, one of the most popular search engines today, for ranking searched items[4].

Brin and Page (1998) also presented Google as a distributed architecture for scalable Web crawling, indexing, and query processing, taking into account link-based ranking functions such as PageRank. There has been research on extended versions of *PageRank* in which various damping functions were proposed and effectiveness/efficiency studied (Baeza-Yates et al., 2006; Bar-Yossef and Mashiach, 2008). Nonetheless, in some cases, *PageRank* did not significantly outperform simple citation count (or indegree-based) methods (Baeza-Yates et al., 2006; Najork et al., 2007).

Whereas in PageRank Page et al. (1998) separated popularity ranking from content, the *HITS* (Hyperlink-Induced Topic Search) algorithm addressed the discovery of authoritative information sources relevant to a given broad topic (Kleinberg, 1999). Kleinberg (1999) defined the mutually reinforcing relationship between hubs and authorities, i.e., good *authority* web pages as those being frequently pointed to by good

[4]Detail about Google's current ranking techniques is unknown.

hubs and good *hubs* as those that have significant concentration of links to good *authority* pages on particular search topics. Following the logic, Kleinberg (1999) proposed an iterative algorithm to mutually propagate hub and authority weights. The research proved the convergence of the proposed method and demonstrated the effectiveness of using links for locating high-quality or authoritative information on the Web. A recent study comparing various ranking methods found that effectiveness of link-based methods such as *PageRank* and *HITS* depended on search query specificity and, in agreement with Kleinberg (1999), they performed better for general topics and worse for specific queries compared to content-based *BM25*F[5] (Najork et al., 2007).

For similar page searching, Dean and Henzinger (1999) proposed and implemented two co-citation-based algorithms for evaluation of page similarity and used them to identify related pages on the Web given a known one. Without any actual content or usage data involved, the algorithms produced promising results and outperformed a state-of-the-art content-based method. Link-based methods are useful not only for retrieval ranking but also for better web page crawling (Menczer, 2005; Guan et al., 2008). Besides the use of hyperlinks, anchor texts on the links were found to be useful to improve retrieval effectiveness. For web site entry search, Craswell et al. (2001) conducted multiple experiments to show that a ranking method based on anchor text was twice as effective as another based on document content. Menczer (2005) suggested content- or link-based methods be integrated to better approximate relevance in the user's information context.

Another type of analysis involves usage data. For example, Craswell and Szummer (2007) applied a *Markov* random walk model to a click log for image ranking and retrieval. They proposed a query formulation model in which the user repeatedly follows

[5]BM25, or Okapi BM25, was a ranking function developed by Robertson and Spark-Jones and implemented in the Okapi information retrieval system at the City University of London. BM25F takes into accout not only term frequencies but also document structure and anchor text.

a process of query-document and document-query transitions to find desired information. Results showed a "backward" random walk algorithm opposite to this process, with high self-transition probability, produced high quality document rankings for queries. Research also extended the PageRank method to leverage user click-through data. The *BrowseRank* algorithm relied on a user browsing graph instead of a link graph for inferring page importance and was shown in experiments to outperform *PageRank* (Liu et al., 2008).

Arguably, analysis of actual information usage such as clickthrough data provide clues for better relevance-based ranking. It is true that clickthroughs have been popularly used as implicit relevance; however, its reliability as relevance assessments should be further examined. Joachims et al. (2005) analyzed in depth user clickthrough data on the Web and showed that clicking decisions were biased by the searchers' trust in the retrieval function and should not be treated as consistent relevance assessments. For instance, when a hyperlink is listed first in the search results, its probability of being chosen increases regardless of its relevance. It is therefore premature to simply assume that clicking on a listed item indicates relevance.

2.2.3 Collaborative Filtering and Social Search

The Web is additionally rich in its users and interactions between users and information items. While many retrieval systems are replacing relevance with authority or popularity on the "free" space of the Web, most of the tools thus built do not support the diversity of voice/opinions. In light of preferential attachment and power-law distribution of connectivity, only a very small number of people and sites catch most of the attention while many are simply isolated and ignored (Morville, 2005). This calls for recognition of the diversity of information sources and interests in system design in order to better serve individual needs.

Automatic recommendation for personalization is widely needed and many systems take advantage of collective opinions embedded in links between users and items such as ratings and clickthroughs for *collaborative filtering*. Under the name of social information filtering, Ringo was one early example of collaborative filtering systems, in which personalized recommendations for music albums and artists were made based on "word-of-mouth" and similarities of people's tastes (Shardanand and Maes, 1995). Presenting the Tapestry project for email filtering, Goldberg et al. (1992, p. 291) coined the phrase "collaborative filtering," which, according to Schafer et al. (2007), is the process of filtering or evaluating items through the opinions of other people. Collaborative Filtering (CF) is to take advantage of behaviors of people who share similar patterns for recommendations. The basic idea is that if one has a lot in common with another, they are likely to share common interests in additional items as well. It demonstrates the usefulness of collective intelligence for personalization.

Schafer et al. (2007) pointed out that pure content-based techniques are rarely capable of properly matching users with items they like because of keyword ambiguity (e.g., for synonyms) and the lack of "formal" content. There are also cases where the users feel either reluctant or difficult to articulate their information needs. Under these circumstances, automatic CF can be used to leverage existing assessments/judgement – sometimes implicit – to predict an unknown correlation between a user and an item. The need for filtering non-text documents, such as videos, further motivated research on collaborative filtering (Konstan, 2004). Content-based filtering and CF are complementary to each other and often used together.

The basic task of CF is, based on a matrix or a network of users and items connected by existing rating values, to predict the missing values. Various models such as nearest-neighbor-based and probabilistic methods have been developed. Most research uses accuracy-based measures such as mean average error (MAE) for system evaluation.

However, several other measures such as coverage, novelty, and user satisfaction have shown to be useful and need further exploration (Herlocker et al., 2004; Schafer et al., 2007).

The effectiveness of collaborative filtering is domain dependent. Specifically, the technique is very sensitive to patterns of a user-item matrix, or the availability of ratings, often sparse. Typically, there are a relatively small number of ratings provided large populations of users and items. The situation is even worse when dealing with new users – it is hard to overcome *cold start* when users' interests are barely known. In the literature, several solutions have been proposed to alleviate this problem. One example is to enrich the user-item matrix by propagating rating signals among the nodes of users and items (Huang et al., 2004). Improvement, however, remains limited. Schafer et al. (2007) recognized the challenge of making meaningful recommendations with scant ratings and suggested that incentives be designed to encourage user participation.

Challenges also involve rating bias. Different users rate items differently – some users tend to give higher ratings than others do. Normalizations of Pearson correlation against average values, for instance, can potentially reduce the bias (Herlocker et al., 1999). In addition, while many items are rated differently by different users, some are commonly favored (e.g., for a popular movie). Ratings of highly popular items tell very little about the users' interests, and if not handled properly, contribute more noise than information. Jin et al. (2004) proposed an improved Pearson coefficient that learned to reevaluate item ratings from training data and computed user-user associations based on weighted values.

Another type of bias, caused by people who rate inconsistently to mislead/cheat the system, is more dangerous. O'Donovan and Smyth (2005) argued that while trust is an important issue in CF, it has not been emphasized by similarity-based research. The

study used prediction correctness to evaluate trustworthiness of neighbors (or producers) and incorporated the trust factor to re-weight recommendations made by neighbors. It was demonstrated that the proposed method improved system performance (a maximum 22% error reduction). It is useful for the detection of malicious users who have provided misleading recommendations inconsistent to predictable patterns. However, it has been shown that users may adjust to match recommenders' bias, making it more challenging to probe rating consistency and trustworthiness for the detection of malicious users (Schafer et al., 2007).

The efficiency of CF largely depends on the user and item population sizes. Although various techniques such as subsampling, clustering, and dimensionality reduction have been developed to tackle the problem, reducing algorithmic complexity remains a great challenge. Many of today's CF applications have to deal with a huge number of rating records. For instance, Netflix has billions of user ratings on films (Netflix, 2006). A data collection of this scale offers opportunities for CF technologies to explore the rich information space for making more accurate predictions. Yet the challenge of efficiency and scalability remains for future research.

One potential direction is the use of distributed architectures for collaborative filtering. While many current CF systems are centralized, using distributed nodes to share the computational burden and collaborate in CF operations makes intuitive sense. Wang et al. (2005, 2006) presented a distributed collaborative filtering system that self-organized and operated in a peer-to-peer network for file sharing and recommendation. Similarly, Kim et al. (2006) employed distributed agents to cooperate in collaborative filtering to address the problem of efficiency and scalability while showing effective performance comparable to centralized methods.

The framework of Collaborative Filtering, or the idea of leveraging collective intelligence, has wide applications in search and retrieval on the Web. By analyzing shared

queries and commonly revisited Web destinations, a system can borrow collective opinions from others to assist individuals in Web search. Smyth et al. (2004), for example, observed that there was a gap between the query-space and the document-space on the Web and presented evidence that similar queries tended to recur in Web searches. They argued that searchers look for similar results when using similar queries and this *query repetition* and *selection regularity* could be used to facilitate searching in specialized communities. A collaborative search architecture called *I-SPY* was developed and evaluated. The basic idea was to build query-page relevance matrices based on search histories and relevance judgements done by a community of searchers, which were later used to quickly identify pages related to the exact or similar queries and to rerank search results. In a similar spirit, White et al. (2007a) presented a new Web search interface that identified frequently visited Web sites, or authoritative destinations, and used this information to boost searches. The user study showed that providing popular destinations made searches more effective and efficient, especially for exploratory tasks.

2.2.4 Distributed Information Retrieval

Classic IR research takes the view of *information centralization* (i.e., a single repository of documents) and focuses on matching and ranking of relevant documents given information needs expressed in queries (Baeza-Yates and Ribeiro-Neto, 2004). On the Web, however, document collections are widely distributed among systems and sites. And often, due to various reasons such as copyright, a centralized information repository is hardly realistic (Callan, 2000; Bhavnani, 2005).

In response to the challenges for information retrieval on the Web, researchers discussed the potential of exploiting a distributed system of computers to spread the work of collecting, organizing, and searching all documents (Brown, 2004). Distributed IR research investigates approaches to attacking this problem and has become a fast-growing

research topic over the last decade. Recent distributed IR research has focused on intra-system retrieval fusion/federation, cross-system communication, and distributed information storage and retrieval algorithms (Callan et al., 2003).

A classic distributed (meta, federated, multi-database) IR system is illustrated in Figure 2.2, in which the existence of multiple text databases is modeled explicitly (Callan, 2000; Meng et al., 2002). Basic retrieval operations include database content (and characteristics) discovery (Si and Callan, 2003), database selection (French et al., 1998, 1999; Shokouhi and Zobel, 2007), and result fusion (Aslam and Montague, 2001; Baumgarten, 2000; Manmatha et al., 2001; Si and Callan, 2005; Hawking and Thomas, 2005; Lillis et al., 2006).

The first layer of challenges involves knowing what each database is about. In a controlled environment (e.g., within one organization), the policy of publishing resource descriptions can be enforced for databases to cooperate. In an uncooperative environment, however, this information is not always known. Query-based sampling is widely used to learn about hidden database contents through querying (Thomas and Hawking, 2007; Shokouhi and Zobel, 2007). The technique has also been used for collection size estimation (Liu et al., 2001; Shokouhi et al., 2006). Some researchers have studied strategies for updating collection information as they evolved over time (Shokouhi et al., 2007). Others focused on the estimation of database quality and its impact on database selection and result fusion (Zhu and Gauch, 2000; Caverlee et al., 2006).

Researchers have proposed many query-based database selection techniques, among which the inference-network-based *CORI* (collection retrieval inference network) algorithm and the *GlOSS* (glossary of servers server) model based on database *goodness* were extensively studied (Gravano et al., 1994; Callan et al., 1995; French et al., 1999). Callan et al. (1995) proposed and evaluated the *CORI* net algorithm for collection ranking, collection selection, and result merging in distributed retrieval environments. Using

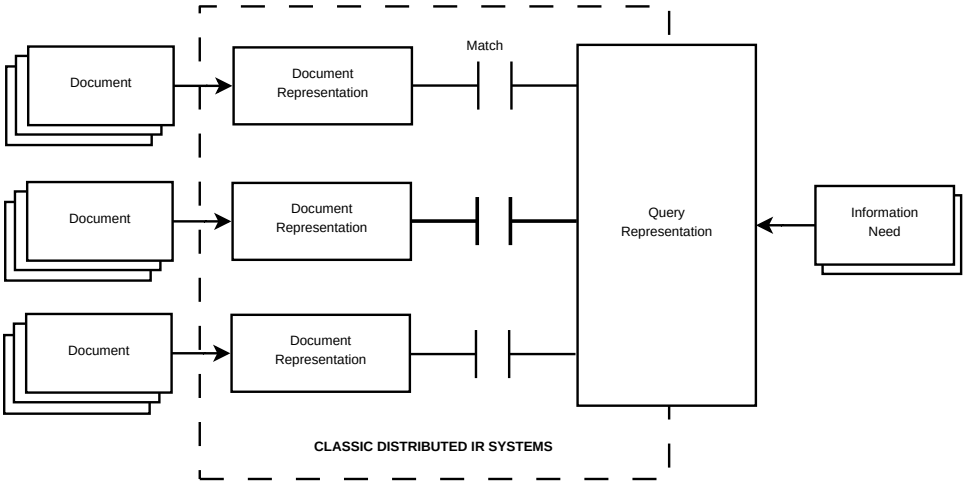


Figure 2.2: Classic Distributed Information Retrieval Paradigm

only collection wide information such as document frequency (*df*) and inverse collection frequency (*icf*) values, the *CORI* method was efficient in terms of communication and storage use on the central server. It was realized that term weights were not comparable across databases among which frequency distributions varied and normalizations were needed. Further on the result merging stage, Callan et al. (1995) proposed the use of weighted scores based on individual documents scores and collection ranking information, which provided an efficient alternative to computationally expensive methods based on term weight normalization. Results showed both efficiency and effectiveness of the proposed algorithm. Further optimizations reduced communication costs through focused collection selection and control on the number of documents to be returned from each collection.

With an aim for efficient text resource discovery in heterogeneous environments (e.g., the Web), Gravano et al. (1994, 1999) developed the *GlOSS* (or Glossary of Servers Server) model that evaluated a database's usefulness or *goodness* for a query. The *goodness* measure used information about the number of documents in a database similar to the query and their similarity scores. Similar to *CORI*, *vGlOSS* (*v* for vector space) was designed to scale using only collection-wide information such as *DF* and a

sum of weights for each term. Various estimators for the ideal rank of databases were proposed and studied. Gravano et al. (1999) also discussed a decentralized version of *GlOSS*, or *hGlOSS*, based on a hierarchical structure in which higher-level *GlOSS* servers summarized underlying distributed databases.

Research found that the proposed methods for database selection, including *CORI* and *GlOSS* had significant room for improvement when only a small number of databases were to be queried (Powell et al., 2000; Powell and French, 2003). Experiments conducted by French et al. (1999) showed that the *CORI* algorithm, as compared to *GlOSS*, produced more accurate predictions and required relatively fewer databases (of totally up to a thousand databases) to be searched. Given a recall level, search effort with the algorithms scaled linearly with the number of available databases, which is hardly scalable for searching a bigger portion of the Web where databases in the range of hundreds of thousands are served.

Powell et al. (2000) compared retrieval performances among three scenarios, namely, a) the centralized scenario where all documents are located in a single database; b) the distributed *CWI* where a testbed was divided into multiple databases and collection-wide information[6] such as global *idf* values was maintained; and c) distributed *LI* where only local (database-wise) information is known. Surprisingly, results supported that a distributed system with a good database selection function can achieve better retrieval effectiveness than a centralized database. Increasing the number of databases being selected improved effectiveness. Nonetheless, with a small number of databases selected, good performance was still maintained. Powell et al. (2000) further discovered that collection-wide information (*CWI*, across all databases) was not necessarily useful. Local (database-level) information sufficed for superior performance.

[6]Collection-wide information (*CWI*) referred to information across all distributed databases divided from a entire testbed collection.

Among many challenges, current distributed IR research encounters difficulties in being selective and accurate on database selection while achieving high precision. For scalability, it is desirable that a very small percentage of all databases are searched (Callan, 2000). This, however, often results in relevant databases being missed (Powell et al., 2000; Powell and French, 2003).

Despite the name *distributed* information retrieval, most classic distributed or federated IR systems work in a centralized manner – there is one meta search server that accepts all queries, distributes them to selected databases, and merges returned results. Many distributed IR systems having been investigated only dealt with dozens of databases (Shokouhi and Zobel, 2007); in rare cases, they reached the scale of thousands to test effectiveness, efficiency, and scalability (Callan, 2000; Thomas and Hawking, 2007). However, the real world scalability of implementation is yet to be considered. Given the heterogeneity of the Web, different communication protocols abound and it requires tremendous effort to implement communication channels to hundreds, if not thousands, of databases.

Classic distributed or federated IR models build on assumptions that do not always hold in the context of the Web. First, a meta search engine has to know databases relevant to a user's query. If a database is unknown or not known yet, obviously, information, even when relevant, will not be retrieved from there. Second, the user is supposed to know the meta search engine and will come to it to conduct searching. In reality, people are involved in various types of information seeking tasks and, when dealing with a particular topic, do not necessarily know where to find it. There could be meta search engines and meta meta search engines and so on to integrate all Web databases for the user to always have a short list of known portals to visit. However, it is not clear whether such a structured modeling will scale. Neither is there evidence that organizations and individuals on the Web will be motivated to organize in this

way.

2.2.5 Conclusion

Information retrieval research has responded to challenges the Web poses given its large size, heterogeneity, and dynamics. Various techniques have been developed to collect and index large volumes of Web pages more effectively and efficiently. Link-based ranking methods address the issue of no central quality control and the need to establish alternative metrics such as popularity, authority, or importance. Interconnections among information items and users, either explicit or implicit, tend to pull related ones together and form semantic clusters; they have been utilized by IR systems to make better recommendations. The use of collective intelligence, as demonstrated by link-based ranking functions and collaborative filtering, displays one aspect of many potentials a networked society has.

Of several known challenges, the problem of the *deep Web* remains barely solved. Distributed information retrieval has shown some potential of bringing different parts together from the hidden space. However, its reliance on centralization of a metasearch server will always suffer from critical problems of scalability, single point failure, and fault tolerance. Further decentralization of meta search models will involve issues beyond the main focus of federated IR research. With this, peer-to-peer information retrieval should be discussed next.

2.3 Peer-to-Peer Search and Retrieval

Classic distributed IR research continues the centralization tradition of IR, in which a single meta search system bears all burden for user interaction, database selection, and result fusion. Such systems are difficult to scale and vulnerable to system failures. Croft (2003) discussed information retrieval in the bigger context of computer science research and pointed out a promising area for distributed, heterogeneous information system research that required contributions from peer-to-peer architectures and retrieval models. Talking about challenges in meta-search and distributed IR, Allan et al. (2003, p. 43) shared this vision:

> A future wherein ubiquitous mobile wireless devices exist, capable of forming ad hoc peer-to-peer networks and submitting and fielding requests for information, gives rise to a new host of challenges and potential rewards.

2.3.1 Peer-to-Peer Systems

Recent years have seen growing popularity of peer-to-peer (P2P) networks for large scale information sharing and retrieval. However, research lacks agreement on what peer-to-peer means. Definitions of peer-to-peer computing either too narrowly refer to purely distributed peers of equivalent functionality or too broadly include servers with centralized operations. Recognizing two common characteristics of peer-to-peer from an "external" perspective, Androutsellis-Theotokis and Spinellis (2004, p. 337) offered the following definition.

> Peer-to-peer systems are *distributed* systems consisting of *interconnected* nodes able to *self-organize* into network topologies with the purpose of *sharing* resources such as content, CPU cycles, storage and bandwidth, capable of adapting to failures and accommodating *transient* populations of nodes

> while maintaining acceptable connectivity and performance, *without* requiring the intermediation or support of a *global centralized* server or authority.

Whereas Grid computing focuses on coordination of persistent and homogeneous computing nodes for high performance, peer-to-peer systems deal with instability, transient populations, and self-adaptation (Androutsellis-Theotokis and Spinellis, 2004) – sometimes, though, the boundary is blurred and a peer-to-peer architecture can be used for grid computing (Batko et al., 2006a,b; Luu et al., 2006; Skobeltsyn et al., 2007). The P2P paradigm holds such promises as scalability, failure resilience, and autonomy of nodes, and has attracted researchers from databases, distributed systems, networking, and information retrieval (Nottelmann et al., 2006). P2P has a wide range of applications such as for communication and collaboration, distributed computation, distributed database systems, and content distribution and retrieval (Androutsellis-Theotokis and Spinelli 2004; Lua et al., 2005). Research also studied distributed collaborative filtering in P2P environments (Wang et al., 2005, 2006; Kim et al., 2006).

According to Androutsellis-Theotokis and Spinellis (2004), peer-to-peer technologies have been used for file exchange (e.g., Napster and Gnutella) and content publishing and storage systems (e.g., Scan and Freenet). There exist various infrastructures for routing and location, based on anonymity and/or reputation management. It is often perceived that peer-to-peer networks are *purely decentralized* without central coordination. Nonetheless, there exist *partially centralized* architectures in which supernodes assume important roles in sub-communities and *hybrid* architectures which include a central server. There were attempts to classify peer-to-peer information retrieval research into a three-dimensional scheme, which includes the application scenario (e.g., enterprise search), the task (e.g., recall, precision, or efficiency-oriented), and the techniques employed (e.g., retrieval, clustering, filtering) (Nottelmann et al., 2006).

Zeinalipour-Yazti et al. (2004) discussed the significance of efficiently finding information in peer-to-peer networks and compared various methods for centralized search, distributed IR, distributed file (object identifier) search, and peer-to-peer IR. While centralized approaches are fast, thorough, and arguably not scalable, distributed IR works in a separated manner but usually includes a central meta server and assumes a global view of individual systems in an always-on environment.

While distributed IR research has made advances in enabling searches across hundreds of repositories and focused on a mediator-based architecture that scales in such environments, peer-to-peer IR has additional challenges. A P2P network usually has a much larger number of participants (often tens of thousands, if not millions) who dynamically join and leave the network, and only offer idle computing resources for sharing and searching (Tsoumakos, 2003; Zeinalipour-Yazti et al., 2004). Usually there is no global information about available collections; seldom is there centralized control or a central server for mediating (Lua et al., 2005). Whereas peer-to-peer file (object identifier) search requires low dimensionality that can be indexed through distributed hashing, peer-to-peer IR involves the complexity of relevance or similarity that challenges the applicability of existing unique-identifier-based routing techniques.

2.3.2 Peer-to-Peer File Search

Some peer-to-peer systems impose no rules for the distribution of files and contents. They are *unstructured* in the sense that the placement of content is not associated with the overlay topology (Lua et al., 2005). Napster (hybrid) and Gnutella (purely decentralized) are unstructured networks. In *decentralized unstructured* networks, locating a file is not straightforward and flooding, though computationally expensive, was among the initial techniques used for searching (Adamic et al., 2001; Androutsellis-Theotokis and Spinellis, 2004). Structured networks, on the other hand, maintain rules on how files should be

placed in terms of the topology. Chord (Stoica et al., 2001) and CAN (Content Addressable Network) (Ratnasamy et al., 2001) use *Distributed Hashing Tables* (*DHTs*) like methods for file distribution, indexing, and location.

Adamic et al. (2001) acknowledged the ad hoc nature of networks such as Gnutella where file locations were unknown before search and examined the question about how files could be found in different network topologies, namely, a power-law graph and a Poisson degree distribution network[7]. Various strategies were proposed and applied to the network search problem. Mathematical analyses and simulations showed that the search strategy of following high-degree nodes worked better than a random walk method in power-law graphs. Search time (in terms of # hops or path length) and coverage (e.g., half graph cover time) were used to evaluate the results, showing costs of the algorithms scaled sublinearly with network size. Adamic et al. (2001) also demonstrated the utility of related search strategies in the Gnutella network, which was found to be a power-law graph.

Similarly, Amoretti et al. (2006) studied the characterization of peer-to-peer network growth and introduced a new routing method called *HALO*, which followed highest degree neighbors and used a *distributed hashing* function for corrections. The work focused on indexing and searching in unstructured P2P networks with super-nodes. Simulations showed *HALO* achieved good performance on scale-free networks in terms of query efficiency.

Lv et al. (2002a,b) argued that flooding-based methods for peer-to-peer search are hard to scale and structured P2P system design, even with better efficiency, is not resilient in the face of a transient population of participants. They proposed the use of random walk search in the presence of heterogeneity of a network (e.g., seen from a

[7]Various classes of graphs, including random, small world, and power-law networks, will be discussed in depth in Section 2.4.

power-law degree distribution perspective such as that of Gnutella) to optimize load balance of a decentralized unstructured network. Various document replication strategies and network topologies were studied (Lv et al., 2002a). Simulation results showed the proposed algorithm reduced network traffic by two orders of magnitude as compared to Gnutella flooding and achieved the same level of efficiency for resolving queries. Random networks yielded best performances in the experiments (Lv et al., 2002a). Interestingly, it was demonstrated that *heterogeneity* is not only a challenge but also a feature that can be taken advantage of for efficient searches in unstructured peer-to-peer networks.

Tsoumakos (2003) reviewed several different peer-to-peer search algorithms in the categories of blind search and informed search methods. The authors conducted experimental simulations on six algorithms from both categories, namely, a) in the blind search category: (1) a modified Breadth First Search (*BFS*) that used "small" floods to optimize the original Gnutella flooding, (2) a random walk that reduced message production to $k \times TTL$[8] in the worst case, and (3) a *GUESS* algorithm that relied on ultra-peers as proxies to communities of leaf-nodes, and b) in the informed search category: (4) an intelligent *BFS* that stored recent answered queries and ranked neighbors in each node and chose most productive neighbors given a recent similar query, (5) a modified Adaptive Probabilistic Search (*s-APS*) method that kept track of neighbors performances on requested objects, and (6) a Distributed Resource Location Protocol (*DRLP*) algorithm that stored the found objects in all nodes on the search path and reused the information for direct access when hit.

Experiments examined that algorithms' effectiveness (success rate), bandwidth consumption (message production), and their responses to topological changes (removal and/or relocation of peers) and object popularity. Results showed that modified- and

[8]TTL, or time to live, denotes the number of hops a message is allowed to travel in a P2P network.

intelligent-*BFS* flooding methods achieved very high success rates and were hardly affected by either topological change or object popularity (Tsoumakos, 2003). However, both profited effectiveness at the cost of huge bandwidth consumption – two orders of magnitude more than the other four. *GUESS* and random walk were not designed to learn from topology nor previous results and achieved low success rates with the least amount of messages.

Tsoumakos (2003) also found that informed search methods such as *DRLP* and *s-APS* exhibited high accuracy at a low cost of bandwidth consumption in static environments. However, they were largely affected by dynamics of the environment. With *DRLP*, the frequency of flooding for reinitiating searches became critical. On the one hand, stored addresses became outdated over time due to network dynamics and needed regular updates. On the other, reinitiation of searches, similar to modified-*BFS* flooding, was costly and required many subsequent successful requests to amortize the initial cost. Interestingly, the *DRLP* was affected more by object relocation than by node departures and, surprisingly in contrast to other algorithms, achieved increased accuracy on less popular objects due to the low frequency of object relocation.

In unstructured P2P networks, flooding-based algorithms exhibited high performance at the huge cost of bandwidth consumption. Modified versions of flooding can produce fewer messages but often fail to perform well. Additionally, these techniques do not adapt to dynamics of the environment. Informed search methods, in general, are more efficient but incur large overheads for initiation and updates of indices, which can be amortized if a significant number of consequent searches will take advantage of them. Although these search mechanisms are not as efficient as algorithms such as CAN and Chord in structured environments, unstructured P2P systems are widely adopted due to the uncontrolled manner and resilience to dynamic, transient populations (Lua et al., 2005).

2.3.3 Peer-to-Peer Information Retrieval

One important objective of network search optimization is overall system utility, i.e., to find targets as quickly as possible without burdening many peers. Flooding like methods often reach a good coverage of a network and are very expensive. Every gain in coverage means costs – even if the algorithms do not have to visit a peer to cover it, looking through a large distributed index of neighbors' files requires significant computational effort. Beyond file name lookup, distributed information retrieval through flooding techniques is arguably impractical (Lv et al., 2002b; Cooper and Garcia-Molina, 2005).

As the peer-to-peer paradigm becomes better recognized for IR research, there have been ongoing discussions on the applicability of existing P2P search models for IR, the efficiency and scalability challenges, and the effectiveness of traditional IR models in such environments (Zarko and Silvestri, 2007). Some researchers reasoned that an IR search query is more complex than key-based file search and exact lookup techniques such as Distributed Hash Tables (DHTs) have limited utilities for peer-to-peer IR (Bawa et al., 2003; Lu and Callan, 2006). Others, nonetheless, applied DHTs to structured P2P environments for distributed retrieval and focused on building an indexing structure over peers for popular queries (Luu et al., 2006; Skobeltsyn et al., 2007). Bender et al. (2005) relied on a Chord-style dynamic DHT in the MINERVA architecture for distributed indexing and studied precise overlap-aware collection selection in structured peer-to-peer environments.

Similar in spirit to DHTs is the duplication of neighbors' indices or the so-called look-ahead strategy for indexing files from neighbors within some defined distance (Adamic et al., 2001; Amoretti et al., 2006). Kurumida et al. (2006), for example, used combined strategies of random-walk, look-ahead, and restrictive back-walk for searching in random, small world (WS model), and scale-free networks. Although the methods produced promising results, their utility very much depends on the assumption that

peers have capacities to index document in the neighborhood. These strategies are feasible for exact file name searches on keys (names) and values (locations).

For information retrieval based on a large feature space, which often requires frequent updates in a dynamic environment, it is challenging for distributed hashing to work in a traffic and space efficient manner. For such a distributed index to be manageable, the ALVIS architecture, for example, employed various strategies to choose highly discriminative keys and to truncate less popular key-document postings (Luu et al., 2006; Skobeltsyn et al., 2007).

Whereas P2P IR research was primarily concentrated on searching in distributed environments, some have studied information browsing in structured peer-to-peer networks. Fischer and Nurzenski (2005), for example, applied the *Scatter/Gather* modality for content browsing in a hierarchical P2P system called *Pepper*, in which *leaves* (or ordinary peers) maintained local collections while *hubs*, as intermediaries, organized the network in a three-tier hierarchy. The system took advantage of precomputed cluster structures in peers for global *Scatter/Gather* browsing and used various strategies to minimize traffic for communicating cluster selection and document data. Experimental simulations showed that the P2P system offered efficient clustering for *Scatter/Gather* browsing of a distributed collection. Surprisingly, for finding desired peers through Scatter/Gather, connecting similar peers to the same hub did not show advantage over a randomly connected network.

Zeinalipour-Yazti et al. (2004) reviewed various techniques used for information retrieval in peer-to-peer environments, which included flooding techniques and intelligent search mechanisms (*ISM*), and conducted simulated experiments on a network of 104 peers, each containing a subset of the Reuters-21578 document collection, to test information retrieval effectiveness (recall) and efficiency (# messages used). The following four techniques that only required local knowledge for IR search were studied, namely,

1) a breadth-first search *BFS* or flooding method, used as the baseline given its extreme cost and thoroughness; 2) an *RBFS*, which was to improve the efficiency of *BFS* by estimating the probability of a query reaching some large network segments; 3) a $>$*RES*, which forwarded a query to a subset of peers based on aggregated statistics of previous performance; and 4) an *ISM*, which maintained a profile mechanism, explored and learned about neighbors' topicality, and forwarded queries to peers who were predicted to have more relevant documents.

Results showed that *RBFS*, *ISM*, and $>$*RES* used significantly few messages for peer-to-peer retrieval than flooding. *ISM* found the largest number of relevant document (best recall). $>$*RES* and *ISM* started with low recall but caught up after peers learned about their neighbors. Zeinalipour-Yazti et al. (2004) indicated that *ISM* worked well on networks where peers had specialized knowledge and where strong degrees of query locality presented. The authors discussed existing challenges for efficient information retrieval in peer-to-peer networks and the use of semantic segmentation to facilitate search.

Unstructured overlay systems work in a nondeterministic manner and have received increased popularity for being fault tolerant and adaptive to transient populations (Lua et al., 2005). In recent years, semantic overlay networks (*SONs*) have been widely used for P2P IR, in which peers containing similar information formed semantic groups for efficient searches (Crespo and Garcia-Molina, 2005; Tang et al., 2003; Raftopoulou and Petrakis, 2008). Some research followed a very structured style for distributed indexing and network topology construction (Tang et al., 2004). Some central control or flooding mechanism was needed for maintaining overlay hierarchies (Crespo and Garcia-Molina, 2005). Others applied the semantic overlay technique to purely decentralized unstructured P2P systems through self-organization and local reconstruction (Doulkeridis et al., 2008).

Research has studied hybrid peer-to-peer architectures with loosely structured overlay networks, in which regional directory services and rules for content placement were used to facilitate search (Bawa et al., 2003; Lu and Callan, 2003, 2004; Hawking and Thomas, 2005; Lu and Callan, 2006). Freenet, loosely structured, uses a similarity-based approach for location estimation. In Freenet, it was shown that the enforcement of clustering in the key space significantly improved retrieval performance (Lua et al., 2005).

Drawing on inspirations from social network theory and existing IR techniques, Bawa et al. (2003) presented *SETS*, a distributed architecture for peer-to-peer retrieval, which partitioned sites into topical segments and took advantage of long-distance (weak) and short-distance (strong, local) links for efficient lookup of relevant information. Following the cluster hypothesis that closely related documents tend to be relevant to the same requests (van Rijsbergen and Sparck-Jones, 1973; Rijsbergen, 1979), the study relied on the topic segmentation and focused on *recall* of relevant documents through local propagation. As the authors acknowledged, the importance of *recall* is domain dependent (e.g., critical for legal or patent retrieval) and subject to peer constraints.

Experimental results showed that a cosine-similarity-based query-driven routing strategy substantially outperformed a random approach and was within a small margin to the optimal (best possible) in terms of efficiency (overall processing cost or the number of peers/sites involved) and effectiveness (recall) (Bawa et al., 2003). Scalability of the architecture was demonstrated on a CiteSeer collection of about eighty thousand sites, with which the average latency of finding relevant information (or the number of sites involved to find the first relevant document) was eight. This is not a surprise given that there were many relevant documents.

Lu and Callan (2003) compared various combinations of algorithms for resource selection and document retrieval in a hybrid hierarchical peer-to-peer networks (of $2,500$

peers/collections from TREC WT10g) and found that content-based selection and text retrieval algorithms were substantially more accurate and efficient than name-based and flooding methods for IR purposes. However, it was acknowledged that the communication costs for updating resource content description required further investigation and might complicate scalability in environments where bandwidth is an issue (Lu and Callan, 2003). Lu and Callan (2007) further experimented on a larger testbed of $25,000$ collections from .GOV2 and demonstrated the effectiveness of hierarchical overlay networks for search. In these studies, relevant documents were loosely defined based on top-ranked items from a centralized system. Given a moderate size of relevance base, recall was one of the major evaluation metrics. Lu and Callan (2006) also studied user modeling for personalization and transient information needs in this environment.

Doulkeridis et al. (2008) developed an iterative method that employed zone initiators (randomly selected) to create initial groups of peers (zones), perform hierarchical clustering on information collections within each zone, and work with other initiators to form higher hierarchical levels. The final result of the process was a semantic tree structure spanning the entire network, which enabled efficient location of relevant collections through super-peers without global control.

While certain network structures might be desirable for efficient query routing, one would argue that the expected structures can hardly be supervised. In the *SETS* architecture, for example, the assumptions about distributed collections of information and limited local intelligence were strictly followed. Nonetheless, some global information about peers and topic segmentation was maintained by a central administrative site to guide new participants and to propagate information about updated segments (Bawa et al., 2003). Although the centralization itself might become a scalability issue, the potential overload was alleviated through the use of a *leases* strategy in which a

peer/site contacted the central server only when its lease expired.

With network topology and placement of content tightly controlled, structured peer-to-peer networks have the advantage of search efficiency. However, they are not widely used for peer-to-peer IR systems and their ability to handle unreliable peers was not sufficiently tested (Lua et al., 2005). Although *supernodes* or central servers in a hybrid or partially decentralized peer-to-peer system can potentially make searches efficient (e.g., in KaZaA), they have to coordinate a significant amount of communication traffics and may eventually become overloaded if not designed properly.

According to Cooper and Garcia-Molina (2005), super-node networks were shown to be fault susceptible, with a failure (or attacks on the supernodes) potentially leading to a large disconnected community and a significant decrease in coverage (see also Albert and Barabási, 2002). A self-organized (ad hoc) network distributes load more evenly and is less vulnerable to single point failures. In a purely decentralized network, individual systems or peers give priority to and exercise their self-interest, with autonomy to connect to others. Distributed system design usually has to take the network structure of a connected community as it is and develop better mechanisms to take advantage of it for efficient search. Given such constraints, it is more "naturalistic" to study peer-to-peer search in networks self-organized by peers with local visions.

It is worth noting that in many purely decentralized *unstructured* networks where there is no global rule for file placement, there is a tendency for similar peers to connect to each other. Hence, similar contents are likely to appear in self-formed clusters, potentially enabling efficient searches (Adamic et al., 2001; Kleinberg et al., 1999; Albert and Barabási, 2002). Research has found that semantic locality can be reinforced and communities formed through peer interactions (Akavipat et al., 2006).

In this direction, Cooper and Garcia-Molina (2005) investigated a self-organized network for efficient search and load reduction, and focused on how peers self-tuned,

with two operations *connect* and *break*, to make the network even more efficient. Whereas *connect* enabled peers to search and link to one another, *break* allowed them to remove a link that caused too much trouble. With all local/individual decisions on how one peer connected to another for searching and indexing, the network thus formed was shown to be even more efficient (while reducing peer load significantly) than those with supernode topologies. This demonstrated the potential of peers' self-adaptation (self-organization through *connect* and self-tuning through *break*) for global optimization for search.

In this work, *connect* was designed as a random process for efficiency and the system later relied on *break* to reconfigure or fine tune the network. A further version of *connect*, namely, *propertied connect* was also developed to avoid redundant links such as *one-index-cycle* and *search-fork* for potentially better network efficiency. In terms of an efficiency measure based on *messages per covered node* (*MCN*), arguably not an ideal evaluation metric, the self-organization largely outperformed super-node networks with more central control and scaled very well to a thousand node level (with almost constant *MCN*). Performances in terms of search latency showed a mixed story and a conclusion hard to reach. Overall, Cooper and Garcia-Molina (2005) focused on improving peer-to-peer search in the spirit of intelligent flooding, where coverage was favored for findability. The work did not study a large number of searches concurrently traveling in the network and scalability of search algorithms in such a realistic environment. These questions were left for further research on related methods.

Cooper and Garcia-Molina (2005) observed that breadth first search (flooding) is more responsive given that searches are conducted in parallel. However, if peers are burdened by many concurrent queries, the entire network will be slowed down as well. Some researchers reasoned that flooding is not desirable as it costs too much network

traffic and greedy routing (a depth first, random walk style method) scales well because it uses a single query instance for network traversal (Lv et al., 2002a). Li et al. (2007), in favor of restrictive flooding, recognized that greedy routing is likely less responsive from a single query perspective but is potentially superior for overall system utility. Effort is needed for further investigation of individual responsiveness vs. overall scalability for information retrieval in peer-to-peer networks.

2.3.4 Conclusion

To facilitate searching, many peer-to-peer IR systems used hierarchical structures with central/regional servers as fast channels that connected various remote parts (e.g., Lu and Callan, 2003, 2007; Fischer and Nurzenski, 2005; Zhang and Lesser, 2005). Semantic overlay networks (*SONs*) were widely adopted as well to support topical segmentation for efficient search operations (e.g., Crespo and Garcia-Molina, 2005; Doulkeridis et al., 2008). Such architectural designs did lead to improved findability of information items. However, the central servers or supernodes in these networks are often an issue of scalability and fault tolerance – they could become overloaded and make the entire system vulnerable to attacks.

Additionally, an artificial structure such as a hierarchy is not commonly seen in self-organized networks. Some would argue that such a structure cannot be imposed in many situations given individual objectives for participating in a peer-to-peer environment. As we will see in the Section 2.4, many real networks, very different from hierarchical structures, manifested small world, scale free (or broad scale), and highly clustering properties that make efficient searching promising (Albert and Barabási, 2002; Kleinberg, 2006a). These network structures, produced under individual peer capacities and constraints, have revealed to us how peers can collectively scale given how much they individually can afford to do. So far, the literature review has come close

to a point where an unstructured, bottom up (decentralized) approach without global control seems favourable (Lua et al., 2005).

Existing peer-to-peer IR research has produced promising results. Particularly, systems such as semantic overlay networks were able to find topically associated segments quickly and retrieve a significant number of relevant documents. Queries used in these studies were often broad and the emphasis was usually on *recall*. Even when the network was large, related segments were not extremely difficult to reach given a large relevance base (see also Figure 2.9 in Section 2.6 and Table C.1 in Appendix for detailed data.). Findability has yet to be tested on large networks when very personalized or specific items are to be found – or when people only want to receive a few highly relevant items because more is painful (Mooers, 1951). How to find an information needle from the haystack remains an issue of scalability.

2.4 Complex Networks and Findability

Previous sections discussed several challenges faced by information retrieval in general and IR on the Web in particular. With regard to the problems of large, distributed, heterogeneous, and dynamically changing information collections on the Web, the focus has been shifted from distributed information retrieval with some degree of centralization to recent development in peer-to-peer search and retrieval. Some studies have shown promising results for findability of information items in distributed networked environments (e.g. Bawa et al., 2003; Zhang et al., 2004; Lu, 2007; Doulkeridis et al., 2008). Yet the scalability of findability in huge networks remains an open question. More has to be known about common mechanisms in such networks, allowing for better understanding of the problems at a proper abstraction level and generalization to broader contexts. Research on complex networks has studied related problems in their basic forms and demonstrated useful results.

2.4.1 The Small World Phenomenon

The common experiences of meeting a random person who shares a mutual friend inspired studies on the small world phenomenon. In 1960s, Milgram (1967) asked the question about how many intermediate links were needed for any two people in the world to be connected. Research by Itheilde Sola Pool at MIT and Manfred Kochen at IBM studied the problem in mathematical terms and found a $50 - 50$ chance that any two persons in the U.S. (of 200 million then) could be linked up with two intermediate acquaintances given each person knew 500 random others. Apparently, the method based on the assumption of randomness did not take into account the complexity of social structures, in which a society tends to be fragmented into social classes and cliques.

Milgram (1967) studied the small world problem through a direct experimental approach, in which *random* people were chosen to start forwarding mail folders through friends and relatives to targeted persons (one in each experiment set). Among the successful chains (e.g., 44 packets out of 160 in the Nebraska study reached the target), the number of intermediate links ranged from *two* to *ten*, with the median at $five$ and projected average length roughly six. As Kleinberg (2000b, 2006b) noted, Milgram's research established not only the abundance of short chains connecting pairs of people in a large social network but also people's collective ability, without global information, to find the short chains[9].

Milgram (1967) found valuable patterns from the experiments. Interestingly, with regard to the geographic movement of mail folders being forwarded, "there was a progressive closing in on the target area as each new person is added to the chain" (Milgram, 1967, p. 66). Results also indicated that participants were three times as likely to forward a mail to a same sex person as to someone of the opposite sex.

Similar results were found when Dodds et al. (2003) conducted an experimental study that involved more than sixty thousand email users to forward messages to one of the eighteen targets in thirteen countries. The study found a typical pair-wise chain length between five and seven, and people often used very simple rules to nominate their subsequent recipients, e.g., based on geographical proximity and occupational similarity. Surprisingly, highly connected "hubs," or people with many social connections, were rarely useful in successful chains, which primarily relied on friendships formed through work or school affiliations and took advantage of weak ties to bridge "distant" parts of

[9]On the one hand, in Milgram's experiments, chain lengths observed might be longer than shortest paths that existed – people made good choices but not necessarily best choices to follow the shortest paths in the experiments. On the other, the high drop-out rates in the studies (e.g., 126 of 160 in the Nebraska study) not only added uncertainty about the observed chain lengths but also raised doubts about people's collective ability to find short cuts if they do exist. A recent analysis conducted by Goel et al. (2009) projected that search distances in previous small world experiments were much longer topological distances.

the network (Granovetter, 1973, 1983; Dodds et al., 2003). It was shown that small changes in chain lengths and participation rates can change the rate of reached targets dramatically. Hence, individual incentives, besides network structure, are crucial for enabling a searchable social network.

Treating the Web as a graph whose vertices are documents and whose edges are directed hyperlinks, Albert et al. (1999) estimated that there was a nineteen-degree separation of all documents on the Web. However, to find a relevant document, the authors argued, is not as easy as the small number 19 looks – not only the desired document is nineteen clicks away but so are all documents on the Web. In Broder et al.'s (2000) macroscopic view of the Web, while the majority of web pages could reach one another along directed links, a significant portion formed single direction paths to others but could not be reached the other way. Albert et al. (1999) observed that efficient traversal of such a network for finding desired information requires an agent be sufficiently intelligent to interpret links and follow relevant paths. Kleinberg (2000b, 2006b), on the other hand, concluded that certain network structural characteristics have to be met in order for efficient navigation to be possible.

2.4.2 Complex Networks: Classes, Dynamics, and Characteristics

Albert and Barabási (2002) conducted a comprehensive review of research on complex networks and focused on topological statistics. While many real networks were traditionally treated as random graphs, recent studies showed that most of them departed far from the random model first proposed and studied by Erdös and Rényi (1959). In order to compare and evaluate various real networks and models, Albert and Barabási (2002) proposed the use of quantities measuring the property of small world (average path length), clustering (clustering coefficient), and degree distribution.

In a network, the distance or path length between two nodes is the number of edges along the shortest path connecting them (Albert and Barabási, 2002). Average path length is the average of all pair-wise distances or path lengths in the network whereas diameter refers to the longest pairwise distance. Clustering coefficient measures a network's tendency to clustering and is defined as the average ratio of a node's neighbors being connected as well, or in terms of Newman et al. (2002), the ratio of connected triples being triangles (*fraction of transitive triples*).

Erdös and Rényi (1959) used probabilistic methods to study problems in random graphs and offered some basic understandings of networks. Let N be the total number of nodes and p the probability of every pair being connected. It was shown that the critical probability at which almost every graph contains a subgraph with k nodes and l edges is: $p_c(N) = cN^{-k/(k-1)}$. Different subgraphs (e.g., trees and cycles of different orders) appear at different critical probability levels. For most values of p (not too small), random graphs tend to have similarly small diameters, i.e., the maximal distance between any pair of its nodes. In random networks, the clustering coefficient always follows $C_{rand} = p$, given that the probability of two neighbors being connected is equal with the probability of any randomly selected nodes being connected. This is usually much smaller than small world networks of the same size and an equal number of edges, in which nodes tend to form local communities and are therefore highly clustered.

Most real networks, including the Internet, WWW, and scientific collaboration networks, were found to display small world properties (Albert et al., 1999; Amaral et al., 2000; Newman et al., 2001; Albert and Barabási, 2002). These networks have a relatively short path between any two nodes, similar to random graphs in which the typical distance between two nodes scales as a logarithmic function of the size. However, a real network usually has a much larger clustering coefficient than a random network of equal numbers of nodes and edges.

Watts and Strogatz (1998) proposed a small world network model, namely, the Watts-Strogatz (WS) model, to accommodate networks that lay in between an ordered finite dimensional lattice and a random graph. The model starts with an ordered ring lattice with N nodes, each of which connects to K nearest neighbors ($K/2$ on each side), and then randomly rewires each edges with probability p. For $p = 0$ the original network is unchanged whereas for $p = 1$ it becomes a random network. The model was based on the observation that people have many local connections (e.g., with family members, friends, and colleagues who often know each other) and some long-range contacts, or weak ties, that bridge subcommunities (Granovetter, 1973). In response to the problem of potential isolated clusters, Newman and Watts (1999a,b) also developed a variant of the WS model, in which edges were added to randomly chosen pairs without any existing edges being removed.

Interestingly, the coexistence of small average path length l and large clustering coefficient C were found in the WS models, in agreement with characteristics of many real networks – widely known as small world networks. The average path length l scales linearly with the network size for small p and logarithmically for large p. The large clustering coefficients, in social networks, are a result of *strong ties* within *local* groups. Weak ties, as Granovetter (1973, 1983) suggested, bring various subgroups of the network together and prevent the system from being fragmented and incoherent, leading to a more connected world with shorter paths.

Many real networks also follow a *power-law* degree distribution[10], largely deviating from a Poisson distribution exhibited in random networks. In a power-law network, the distribution of connectivity decays with a power law function linear on log-log coordinates. Intuitively speaking, in a power-law network, as exemplified in Figure 2.3, while many nodes are highly connected (rich), the majority of nodes have a very small

[10]Degree distributions of some networks follow a power-law with an exponential or Gaussian tail.

number of connections (very poor). Figure 2.3 shows a power-law indegree distribution of 200 million Web pages (with 1.4 billion links), in which a very small number of pages received more than $10,000$ incoming links (bottom right) and the majority were rarely pointed to by others (top left, more a hundred million pages with indegree ≤ 10) (Donato et al., 2007).

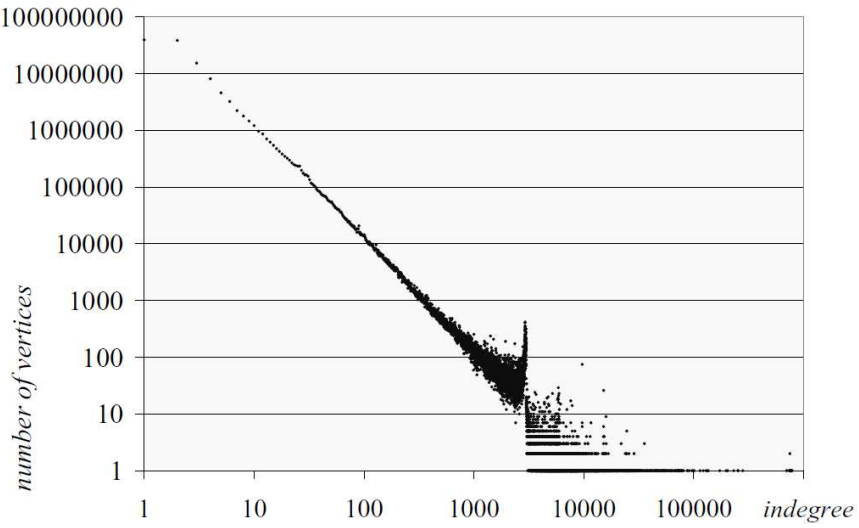


Figure 2.3: Power-law Indegree Distribution of the Web, on log-log coordinates, adapted from Donato et al. (2007). The X axis denotes *indegree*, or the number of incoming links a web page has received. The Y axis represents $frequency$, or the number of web pages that received a number of incoming links as indicated on X. Note that power-law has a linear display on log-log coordinates.

To accommodate real networks with power-law degree distributions, research proposed generalized random graph models by introducing degree distributions to guide the connections. However, these models did not project other quantities consistent to real networks. For example, real networks tend to have a larger average path length than that of random graphs with power-law degree distribution. They usually have much larger clustering coefficients, a property independent of network size, due to strong connections *within groups*.

Barabási and Albert (1999) proposed the *scale free* (SF) model to simulate the dynamics of power-law network growth based on the observation of preferential attachment, i.e., the probability of two nodes being connected has dependence on the nodes' current connectivity (or degrees). Barabási and Albert (1999) reasoned that growth (adding news nodes to a network) and preferential attachment (a degree-dependent probability for adding connections) are simultaneously needed to capture the degree distribution as a result of a dynamic process. With new nodes joining the network, they tend to attach to existing nodes that are already highly connected – the rich get richer. Results of network growth and preferential attachment in the *scale free* model were found to be consistent with observed power-law network properties.

While various preferential attachment formulations have been studied, researchers also relied on other mechanisms than explicit preferential attachment and offered different perspectives on network dynamics. Inspired by the observation that many hyperlinks were "imported" from one site to another on the Web, Kleinberg et al. (1999) proposed the use of a copying mechanism to explain the power-law distribution of the Web. The model, without explicitly including preferential attachment, does have a degree-dependent component of the probability for connectivity. Such models, similar in the spirit to preferential attachment, plausibly explain the dynamic process of indegree growth on the Web. The out-degree distribution of the Web and its dynamics, however, remain barely understood in research and have yet to be modeled and scrutinized (Donato et al., 2007).

Many complex networks exhibit a high degree of robustness. Because of redundant wiring of network structure, local failures rarely lead to global reduction of network capacity. At the topological level, simulation experiments and analytical results showed that scale free networks are more robust against random local failures than random networks do (Albert and Barabási, 2002). However, they are more vulnerable to attacks

targeted on highly connected nodes. This has important implications on partially centralized peer-to-peer networks, in which reliability of super-nodes is crucial to overall system performance (Lua et al., 2005).

Some researchers explained the common presence of scale-free and high clustering characteristics in many real networks as a consequence of hierarchical organization (Ravasz and Barabasi, 2003). That is, individuals form small groups and organize hierarchically in increasingly larger groups, resulting in a scaling function in which clustering of a node is inversely proportional to its number of links. It was shown that several networks such as the World Wide Web followed the scaling function, consistent to the hierarchical interpretation (Ravasz and Barabasi, 2003). This view, together with the clustering effect, is very useful for search in small world networks (Kleinberg, 1999; Watts et al., 2002). Hierarchical segmentation or semantic overlay, as discussed in Section 2.3, has been widely used in peer-to-peer systems for efficient search (e.g., Lu and Callan, 2003, 2007; Doulkeridis et al., 2008).

Amaral et al. (2000) presented evidence in small world networks that, besides *scale-free* networks characterized by a power-law connectivity distribution (Barabási and Albert, 1999), several known real networks displayed *broad-scale* or *single-scale* characteristics. Particularly, some networks such as an actor-actor collaboration graph are categorized as *broad-scale* or truncated scale-free because they follow a distribution of a power-law region with a sharp cutoff. Others such as power-grid and airport connectivity distributions follow a fast decaying tail of, e.g., exponential or Gaussian, and are called *single-scale* networks (Amaral et al., 2000).

The original scale-free model, relying on network growth and preferential attachment, properly captures power-law degree distributions but fails to explain the nature of broad-scale and single-scale networks (Barabási and Albert, 1999). Amaral et al. (2000) reasoned that two classes of factors or constraints potentially limit the networks

from a constant preferential attachment of new links and hinder the formation of scale-free networks. The effect *aging of the vertices* refers to the potential of a vertex or node becoming inactive and rejecting new links, e.g., when an actor stops acting. The other effect, namely, the cost of adding links to vertices or the *limited capacity of a vertex*, denotes physical limits of nodes. For example, an airport can only serve a limited number of landings/departures per hour and do not have the capacity to be a hub for all airlines. Extensions of the Scale-Free model (Barabási and Albert, 1999) using the two effects produced connectivity distributions with broad-scale or single-scale characteristics (Amaral et al., 2000). With moderate constraints of aging or limited capacity, distributions display a power-law decay followed by a cutoff. Strong constraints, however, lead to no visible power-law region.

2.4.3 Search/Navigation in Networks

Research not only showed the prevalence of the *small world phenomenon* but also demonstrated that nodes, with very local intelligence or limited information, are able to collectively construct short paths to globally identifiable targets in large networks (Milgram, 1967; Kleinberg, 2000b; Dodds et al., 2003; Goel et al., 2009). Previous works have studied dynamics of networks and the potential application of the *small-world phenomenon* in searching for information in networks.

Kleinberg (2000b, 2006a) reflected on why people in Milgram's early small world experiments were able to follow short paths to expected targets and proposed that there be "gradient" of some sort, or some particular network properties, to orient searches and guide them toward destinations. There are, as Kleinberg realized, certain "global reference frames" in which the network is embedded and by which the targets are defined and searches guided. Kleinberg (2000b, 2006a) studied the small world phenomenon from a mathematics perspective and conducted algorithmic investigations of finding

short paths using local information. It was shown that finding short chains in some types of networks is more promising than in others.

Starting from a two dimensional lattice, as shown in Figure 2.4 (a), the study built a model in which nodes are rich in short distance connections and poor in long distance links, with the probability of connecting to a long-distance node Pr proportional to $r^{-\alpha}$, where r is the distance between the pair being considered. Results, as shown in Figure 2.4 (c), indicated that only when $\alpha = 2$ delivery time τ (or the number of nodes involved for each search) is bounded by a function proportional to $(\log N)^2$ on a $2D$ lattice. When α is larger (rare long-distance connections and more homogeneous neighborhood) or smaller (many remote connections and more heterogeneous/diverse neighborhood), an asymptotically much larger delivery time is required regardless of the algorithm used. This finding is generalizable to d-dimensional lattices, where for any value $d \geq 1$, efficient navigability can be achieved with a critical value $\alpha = d$ (Kleinberg, 2000b,a, 2006a).

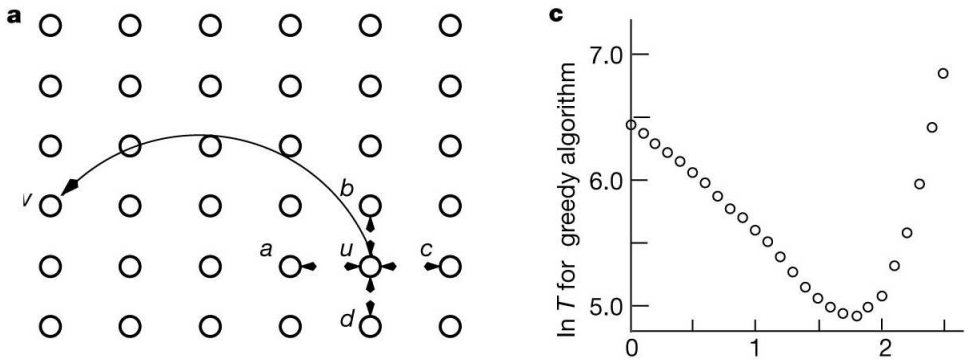


Figure 2.4: Findability in 2D Network Lattice Model, from Kleinberg (2000b,a, 2006a), derived from an $n \times n$ lattice. **A**, each node, u, has a short-range connection to its nearest neighbours (a, b, c and d) and a long-range connection to a randomly chosen node, where node v is selected with probability proportional to $r^{-\alpha}$, where r is the lattice ('Manhattan') distance between u and v, and $\alpha \geq 0$ is a fixed clustering exponent. More generally, for $p, q \geq 1$, each node u has a short-range connection to all nodes within p lattice steps, and q long-range connections generated independently from a distribution with clustering exponent α. **C**, Simulation of the greedy algorithm on a $20,000 \times 20,000$ toroidal lattice, with random long-range connections as in a. Each data point is the average of $1,000$ runs.

When long-distance connections are selected at random (i.e., given a uniform distribution over distance at $\alpha = 0$), individuals are disoriented and unable to find short chains when they indeed exist. Strong clustering (i.e., given a large α), on the other hand, increases the separation of all nodes in the network without sufficient weak ties for searches to "jump" (Kleinberg, 2000b; Singh et al., 2001). The critical value of α, in the tradeoff between strong (local) ties and weak (remote) ties, offers some fundamental clues for individuals to find short paths with local information.

While *2D* or geographical models were broadly adopted for studying the network search problem, hierarchical network organization offers an alternative view. The hierarchical view, as discussed earlier, has been used to effectively explain scale-free and strong clustering properties in real networks (Ravasz and Barabasi, 2003).

Watts et al. (2002) reasoned that our social space could be broken down into multiple hierarchical dimensions, in which individuals formed groups and groups of groups in more than one ways. Following this observation of social partitioning, Watts et al. (2002) developed a social network model of H independent hierarchical dimensions, which was iteratively partitioned with a branching ratio b into l levels and individual groups (tree leaves) of size g. While lowest common ancestor height in a hierarchy was used to measure pairwise distance x[11], the probability of two nodes connecting each other followed the function: $p(x) = c \exp(-\alpha x)$. Figure 2.5 shows an example of the model representation, in which $b = 2$, $l = 4$, and $g = 6$.

Considering the probability of a node terminating a search $p = 0.25$ and the chance of any search chain eventually reaching the target at probability $q = 0.05$ (i.e., 5% completed searches), a maximum search chain length $\langle L \rangle \leq 10.4$ was required – the longer the chain, the more likely it would be terminated by someone. Provided all these conditions, Watts et al. (2002) ran simulations on various population sizes, from

[11]The measured distance of two nodes was the minimum value of all dimensional distances.

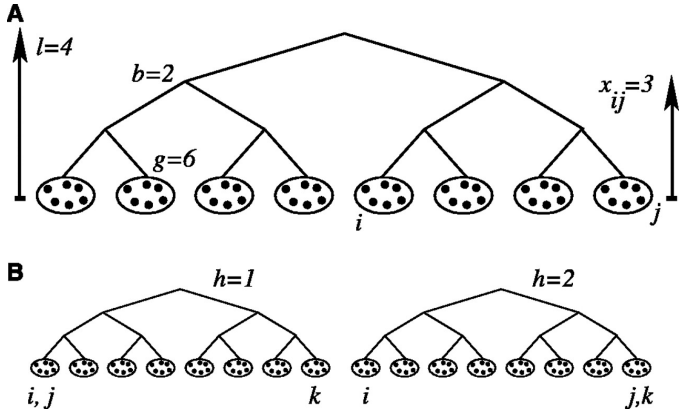


Figure 2.5: H Hierarchical Dimension Model, adapted from Watts et al. (2002). **(A)** Individuals (dots) belong to groups (ellipses) that in turn belong to groups of groups, and so on, giving rise to a hierarchical categorization scheme. In this example, groups are composed of $g = 6$ individuals and the hierarchy has $l = 4$ levels with a branching ratio of $b = 2$. Individuals in the same group are considered to be a distance $x = 1$ apart, and the maximum separation of two individuals is $x = l$. The individuals i and j belong to a category two levels above that of their respective groups, and the distance between them is $x_{ij} = 3$. Individuals each have z friends in the model and are more likely to be connected with each other the closer their groups are. **(B)** The complete model has many hierarchies indexed by $h = 1...H$, and the combined social distance y_{ij} between nodes i and j is taken to be the minimum ultrametric distance over all hierarchies $y_{ij} = min_h x_{ij}^h$. The simple example shown here for $H = 2$ demonstrates that social distance can violate the triangle inequality: $y_{ij} = 1$ because i and j belong to the same group under the first hierarchy and similarly $y_{jk} = 1$ but i and k remain distant in both hierarchies, giving $y_{ik} = 4 > y_{ij} + y_{jk} = 2$.

a hundred thousand to two hundred million nodes, to discover searchable zones in terms of α (the homophily or clustering exponent) and H (the number of hierarchical dimensions).

Results in Figure 2.6 showed that most searchable networks were with parameters $\alpha > 0$ (i.e., when nodes associated preferentially with similar/like others) and $H > 1$ (i.e., using more than one dimensions in searches). Interestingly, over the largest searchable range of α, best performance was achieved with $H = 2$ or $H = 3$. That is, individuals were able to find an efficient path to a target by using two to three dimensions when forwarding a message, consistent to existing small world experiments (Milgram, 1967; Dodds et al., 2003). Increase of H reduced the number of connections on each dimension and weakened the correlation of network ties, leading to increased randomness of the network and inefficient searching.

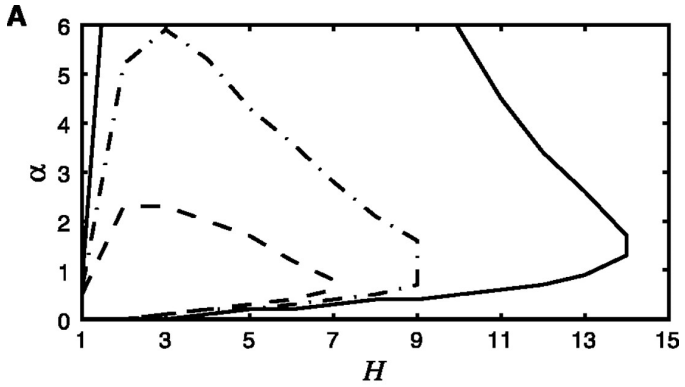


Figure 2.6: Findability in H Hierchical Dimensions, adapted from Watts et al. (2002). **(A)** Regions in $H - \alpha$ space where searchable networks exist for varying numbers of individual nodes N (probability of message failure $p = 0.25$, branching ratio $b = 2$, group size $g = 100$, average degree $z = g - 1 = 99$, 10^5 chains sampled per network). The searchability criterion is that the probability of message completion q must be at least $r = 0.05$. The lines correspond to boundaries of the searchable network region for $N = 102,400$ (solid), $N = 204,800$ (dot-dash), and $N = 409,600$ (dash). The region of searchable networks shrinks with N, vanishing at a finite value of N that depends on the model parameters. Note that $z < g$ is required to explore $H - \alpha$ space because for $H = 1$ and α sufficiently large, an individual's neighbors must all be contained within their sole local group.

Watts et al.'s (2002) model is potentially applicable to information retrieval in distributed networked environments (e.g., for P2P IR) and the reported simulation results will provide guidance on how efficient, scalable searches are possible through hierarchical clustering. Research on semantic overlay networks for P2P systems shares a similar hierarchical clustering view on search efficiency (Crespo and Garcia-Molina, 2005). Yet it is unclear how such multiple hierarchical dimensions can be collectively constructed and maintained by participating individuals who autonomously strive, with local intelligence, to meet their own objectives. Its broader applicability remains a question.

Liben-Nowell et al. (2005) argued that Kleinberg's model was too simplified to capture behavior in real-world social networks and proposed a new model that incorporated a correlation between geography and friendship (social connection), together with population density. Using about 1.3 million blogger sites from the LiveJournal online community, in which inverse relationship with some variance between geographical distance and probability of friendship was observed, experiments showed some degree of

findability of target cities within short paths, particularly when the connection probability function $f(\delta) = 1/\delta^{-1}$, where δ is pairwise geographical distance. Observing the insufficiency of a purely distance-based function, the study then adopted a rank-based friendship function, in which the probability of connecting (or befriending) a person was inversely proportional to the number of closer candidates. Taking into account the variable of population density, Liben-Nowell et al. (2005) demonstrated that the rank-based relationship was exhibited in the LiveJournal data and that short paths are discoverable in such networks[12].

Hu and Di (2008) acknowledged the importance of navigability in networks but observed discrepancies among existing research, particularly, where findings disagreed on what network structures enable optimal search (Kleinberg, 2000b; Lambiotte et al., 2008; Liben-Nowell et al., 2005). Whereas Kleinberg (2000b) and Lambiotte et al. (2008) showed that navigation in small worlds is optimal given a clustering/homophily exponent of 2.0 (in a $2D$ lattice space), Liben-Nowell et al. (2005) found that the optimal parameter should be 1.0. Hu and Di (2008) tried to reconcile the models and reasoned, alongside with Liben-Nowell et al. (2005), that the previous results were actually consistent – the problem was caused by the difference of population density (uniform vs. nonuniform). In addition, as Liben-Nowell et al. (2005) acknowledged, the effective dimensionality of the network also matters – it was estimated that the fractional dimension of the LiveJournal network was 0.8, which can be represented by a single-dimension space that requires optimal clustering exponent $\alpha \approx 1$, consistent with Kleinberg's model.

Simsek and Jensen (2008) identified two features of many networks that are critical for efficient navigation, namely, 1) homophily, which depicts the tendency of connected

[12]Intuitively, Liben-Nowell's (2005) model can be seen as Kleinberg's 2D lattice distorted by a population density distribution, in which connections between very close nodes remain rich and remote links sparse.

nodes/peers being correlated (in terms of the search space), and 2) out-degree that denotes the number of connections a node has. It was 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. The authors hence proposed a measure based the product of a degree term (k_s) and a homophily term (q_{st}) to approximate the expected distance. A method called *EVN* was designed to forward a message/query to the neighbor that minimized the distance expectation by maximizing $k_s \cdot q_{st}$. The experiments found that the simple combined measure (*EVN*) was very effective, especially in power-law (degree distribution) and medium homophily networks where both factors could guide the navigation. One additional advantage of the *EVN* is that it is only sensitive to the ratio of values between two neighbors, not the actual values that might not be accurately measured.

Recognizing the small world properties in a wide range of real networks and their abilities of efficient information routing/signalling without global intelligence, Boguñá et al. (2009) described a general mechanism to explain the connection between a network structure and the function for navigation. They suggested a hidden metric space behind the observable network topology. Experimental simulations revealed that certain characteristics of the correlation between the two spaces – similar to the clustering exponent α in Kleinberg (2000b) and the concept of homophily in Simsek and Jensen (2008) – enable efficient search or navigation through the visible networked space. The authors discussed the implications in Internet routing scalability, efficient searching for individuals or contents on the Web, and studies of signal flows in biological networks.

Boguñá et al. (2009) interestingly introduced *hidden space* for discussions on efficient navigation in complex networks. The concept, however, was not novel in the literature. Kleinberg (2000b, 2006a) used a *clustering exponent* α to correlate the two spaces whereas Watts et al. (2002) and Simsek and Jensen (2008) adopted the term *homophily*

in reference to the correlation. The hidden space, interestingly, is not always as hidden as the phrase may indicate and is often quite visible. In the air travel example used by Boguñá et al. (2009), the hidden space actually referred to the geographical space in which destinations were defined. Apparently, the hidden space should be defined by the search or navigation function so that we can take advantage of it to optimize search. For example, to find relevant information in a large peer-to-peer network, we need to define what relevance is and operationalize it by projecting peers in the information space thus defined. Potentially, how peers connect to each other will have dependence on how close they are in the information space, which, in turn, will guide the finding of relevant information through the visible connection structure.

Although research has widely used the geographic space as a basis for modeling network routing, its applicability in organizational settings is questionable. Adamic and Adar (2005) explored various search strategies based on connectivity, physical proximity, and closeness in an organizational hierarchy for finding short paths in social networks. Simulations on email communication data of 430 individuals within one single organization showed that the strategies using a contact's position in the physical or hierarchical space resulted in effective search results. A similar level of effectiveness was not achieved on an online frienship network of 2000 students, in which a formal hierarchical structure could hardly be constructed.

In experimental simulations on synthetic networks, Boguñá et al. (2009) further manipulated two common properties that appeared in many real small-world networks, namely, scale-free degree distribution and local clustering. Whereas the scale-free distribution was controlled by a power-law exponent γ, the following mechanism parameterized the correlation of the network space and hidden space and indirectly controlled various levels of clustering. That is, the pair-wise connection probability $r(d; k, k')$ of two nodes depends on the distance d of the two nodes (in the hidden space related to

search) and their degrees k and k': $r(d; k, k') = r(d/d_c) = (1 + d/d_c)^{-\alpha}$, where $\alpha > 1$ and $d_c \approx kk'$. With a larger α, remote connections become rare and nodes more locally connected, leading to stronger clustering[13].

Simulation results using greedy routing showed that for smaller degree exponents γ and stronger clustering exponent α, searches traveled shorter paths τ. When clustering was above some threshold, some critical value of γ (≈ 2.6) maximized the success ratio p_s. Based on the results, examples of real networks were plotted on an identified navigable region of clustering and degree exponents.

Investigation of air travel through connected airport illustrated how *greedy routing* can take advantage of the geographical hidden space to follow zoom-out (coarse-grained long-distance search) and zoom-in (find-grained local search) mechanisms to quickly get to destinations (Boguñá et al., 2009). It was realized that the most navigable topologies were enabled by small exponents of power-law distribution (i.e., large number of hubs) and strong clustering (i.e., strong coupling between the hidden geometry and the observed topology). Boguñá et al. (2009) further illustrated that, with this configuration, the routing process quickly found a way to high-degree hubs, moved further from there, and settled toward a low-degree destination.

Some conflicts in research findings appeared. Boguñá et al. (2009) observed that search paths were shorter for smaller power-law degree exponents γ (e.g., 2.0) and stronger clustering (larger α values, 4.5). However, Simsek and Jensen (2008) showed different best results for power-law networks at $\gamma \approx 1.0$ and $\alpha \approx 1.5$. These differences were probably caused by a variety of factors such as network model, average degree, and algorithms employed.

[13]The α parameters, although appeared in different names in Kleinberg (2000b); Simsek and Jensen (2008); Boguñá et al. (2009), had very similar (not identical) functions. They all influenced the formation of local clusters and how likely nodes from different parts of the network connected to each other.

The usefulness of high-degree nodes (hubs) shown in some research (e.g., Boguñá et al., 2009) is at odds with other studies in which hubs were found rarely useful, if not harmful, for small-world searches (Dodds et al., 2003). Different degree distributions might explain the discrepancy. For example, Adamic et al. (2001) found the effectiveness of a degree-based search function worked well in a power-law network but poorly in a Poisson network (see also Adamic and Adar, 2005). If high-degree nodes are indeed useful to redirect long-distance traffics, cautions should be taken at the application level for load balance – super-nodes should have sufficient capacities to handle the traffics (Adamic et al., 2001; Zhang and Lesser, 2006). As was demonstrated by Amaral et al. (2000), structural characteristics of a network manifest individual capacities and constraints in the network. Decentralized systems should be designed in such a way that peers have connectivity in accord with their capacities.

2.4.4 Conclusion

Whereas small worlds have small diameters, collectively constructing short paths to desired targets without global information is not an easy task (Albert et al., 1999; Kleinberg, 2006a). It is fair to say that small worlds do not automatically resolve findability (Morville, 2005). Additional topological characteristics, such as some correlation between the network space and the search (hidden) space, are needed to support efficient network navigation (Kleinberg, 2000b; Watts et al., 2002; Liben-Nowell et al., 2005; Simsek and Jensen, 2008; Boguñá et al., 2009). Fortunately, these characteristics or properties, as suggested by the literature, are not uncommon in real networks.

Information retrieval in a purely distributed networked environment has additional layers of complexity to the problem of finding targets in a dimensional space. So far peer-to-peer and multi-agent IR research has produced promising results. But they do not appear to be as excitingly scalable as findings in complex network research

on abstract models. It remains highly challenging to traverse roughly one hundred peers to find a unique information item in a four hundred million population (i.e., $20,000 \times 20,000$) as was the case in, among others, Kleinberg's (2000b) simulations.

In distributed networked information retrieval, there is no globally unambiguous way to define peers' topical identities, their relevance to queries, and where targets are, as was so in abstract models in complex network research (Kleinberg, 2000b; Liben-Nowell et al., 2005; Simsek and Jensen, 2008; Boguñá et al., 2009). Moreover, relevance also depends on the peers who measure it and is never universally precise. Even when one node or peer is connected to a relevant neighbor (if the relevance can somehow be judged), the node holding the query might not make the right decision to choose the target.

2.5 Agents for Information Retrieval

Complex network research has focused on dynamics of various classes of networks collectively formed by peers with individual objectives, capacities, and constraints, and demonstrated great potentials for efficient traversal in such environments. Discussions on Web information retrieval and peer-to-peer systems show a picture of heterogeneous information collections dynamically changing in a networks of nodes which actively interconnect and interact with one another. Seen from this perspective, components in the traditional view of an information retrieval system, as well as in a distributed IR system, are pushed further apart from one another. In a dynamically evolving environment such as the web, it can no longer be assumed that various parties – people, information, and technologies – will automatically know where to find each other and interact.

2.5.1 A New Paradigm

Baeza-Yates et al. (2007) reasoned that a centralized search engine will become inefficient in the face of Web growth and change, and argued for fully distributed search engines. As illustrated in Figure 2.7, information needs arise from every location in the cloud (a networked space) where information collections "hide,"appear, and evolve. No central system can potentially have full knowledge about where all information collections are and will be. Neither can one predict where particular information needs might arise. One who has an information need does not necessarily know where to search.

Huhns (1998) argued that today's large, open, heterogeneous environments call for cooperative information systems, as being studied in multi-agent systems, that can span enterprise boundaries and make intelligent local decisions without global control in a scalable and cost-effective manner. A schematic view of cooperative information systems is shown in Figure 2.8, in which *agents* interact with one another to provide

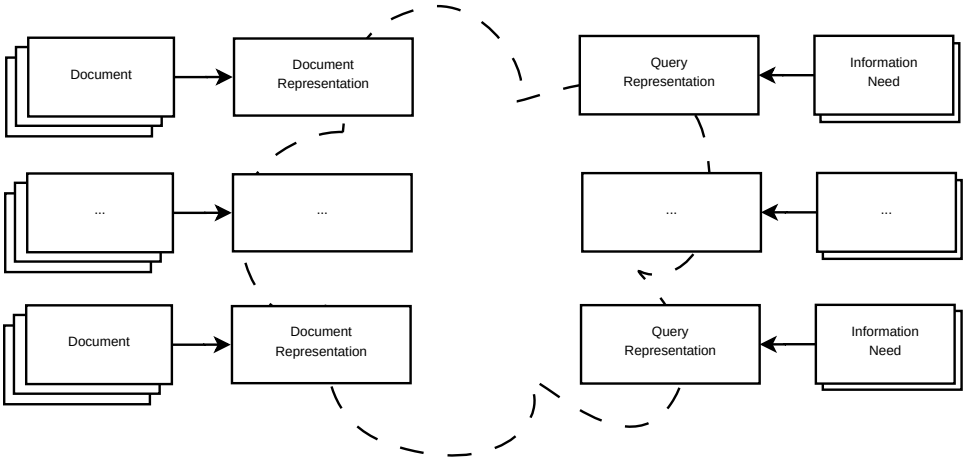


Figure 2.7: Fully Distributed Information Retrieval Paradigm

a human user with a natural means for finding, accessing, and interacting with information over uncontrollable environments. The concept of *mediator*, which enables mapping of resources and applications for others, is key in this environment. Classic distributed information retrieval systems (or meta search engines), as discussed in Section 2.2, can be integrated as mediators in this view.

With respect to interaction with computer systems for information access, some researchers argued for direct manipulation that affords the user control and predictability, others believe in some form of delegation, namely software agent, to reduce the user's work and information overload (Maes, 1994; Shneiderman and Maes, 1997). From a human-information interaction perspective, Marchionini (2008) reflected on the dynamic interactions of information, people, and technologies and proposed a shift from an information-centric view to an interaction-centric view of information, where people and active information interact in a technological substrate and all evolves over time. Information objects may have varied forms and meanings depending on spatial, temporal conditions, and the ones who interact with them. People are no longer passive information consumers but actively participate in the creation, revision, and extension of it. Marchionini (2008) suggested there be an ecological approach to supporting

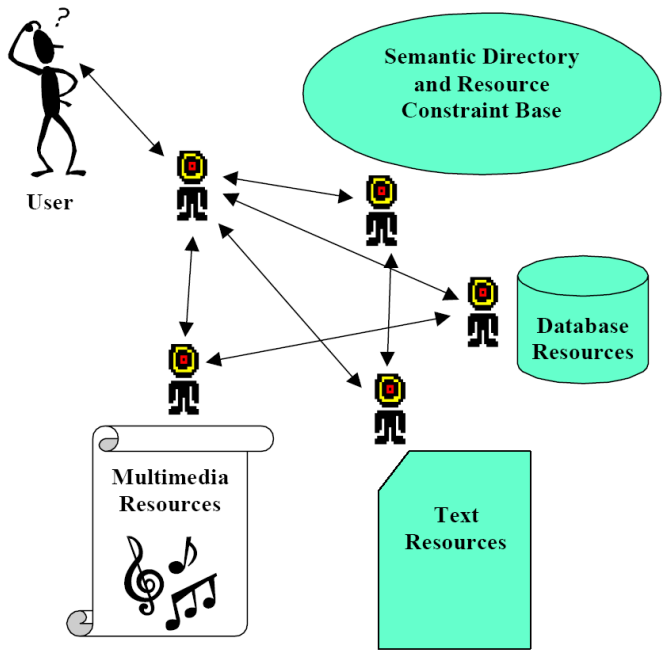


Figure 2.8: Multi-Agent Cooperative Information System, adapted from Huhns (1998).

mutual interactions of all active elements in such an environment. Seen in this view, mechanisms such as cooperative agents are needed to bring related *live* parties together in the dynamic environment.

Agents can not only interact with the user and application but also work with other agents to better assist their human principals. Finding information in networked environments, especially a dynamic one, is not straightforward and can be overwhelming. If one considers retrieval coverage extends to proprietary sites and content in the deep web, individual users will be able to maximize their potential of retrieving relevant information through delegations. One way to achieve this is to allow so-called *agents* to take partial control and play active roles for searching, learning, collaboration, and adaptation in the networked environment. This view of information retrieval systems, as pictured in Figures 2.7 and 2.8, is congruent with a fully distributed information retrieval paradigm.

2.5.2 Agent

Jennings and Wooldridge (1998a, p. 4) defined an *agent* as "a computer system situated in some environment, and that is capable of *autonomous action* in this environment in order to meet its design objectives." In Huhns's (1998) terms, an agent is an active, persistent computational entity that can perceive, communicate with others, reason about, and act in its environment. Agents are not invoked or controlled by others – neither humans nor other agents – but may respond to requests from them. In addition, an *intelligent agent* is capable of *flexible* autonomous action, in the sense of being *responsive* to changes in the environment, *proactive* to it, and *social* in it (Jennings and Wooldridge, 1998b). Subject to local perspectives and no global control, agent-based techniques offer great potential for *reactive* systems too open (dynamically changing), complex, and ubiquitous to be correctly designed and implemented.

Agent techniques have been used in a wide range of areas such as workflow control, information retrieval and management, network management, digital libraries, and entertainment (Jennings and Wooldridge, 1998b; Jennings, 2001; Huhns et al., 2005). The agent paradigm was extensively used in IR research for modeling peer-to-peer search and retrieval (Yu and Singh, 2003; Zhang et al., 2004; Zhang and Lesser, 2007), distributed intelligent crawling (Davison, 2000; Menczer et al., 2004), expert finding (Zhang and Ackerman, 2005; Ke et al., 2007), and information filtering (Mostafa et al., 2003; Mukhopadhyay et al., 2005), among others. Agents are key elements in the Semantic Web of "actionable information" – they can provide, connect with, and process semantic content and services in the flexible environment (Berners-Lee et al., 2001; Shadbolt et al., 2006).

Classification of current agent technologies involves multiple facets. Nwana and Ndumu (1998) categorized software agents in terms of characteristics such as mobility (static vs. mobile agents), internal models for the external environment (deliberative vs. reactive),

and learning and cooperation. Some agents are called information or Internet agents because of their role of gathering information from the network. Others, with mixed functionality embedded in a single agent, are referred to as hybrid. On the WWW, for example, intelligent topical crawlers were widely used as information agents that traversed hyperlinks to collect topical relevant web pages or followed references in information repositories to answer user questions (Menczer and Belew, 1998; Davison, 2000; Pereira and Costa, 2002; Menczer et al., 2004; Guan et al., 2008).

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. The decision about whether to adopt a single-agent or multi-agent approach, Jennings and Wooldridge (1998a) reasoned, depends on the domain of application and can be seen in the light of whether monolithic, centralized solutions or distributed, decentralized solutions are appropriate. IR research has used the single-agent paradigm to model personalization and how changes of personal information needs can be quickly detected and served (Mostafa et al., 1997, 2003). With multi-agent systems, researchers investigated the design of distributed systems for information retrieval and filtering operations (Mukhopadhyay et al., 2005; Ke et al., 2007). Research also compared single-agent and multi-agent systems for information retrieval purposes and argued that multi-agent systems have such advantages as fault tolerance, adaptability, and flexibility (Peng et al., 2001; Zhang and Lesser, 2007).

2.5.3 Multi-Agent Systems for Information Retrieval

For the design of complex software systems, Jennings (2001) argued for an agent-oriented approach in which a collection of interacting, autonomous agents can offer designers and engineers significant advantages over existing methods. A multi-agent paradigm enables the decomposition of a complex system into multiple, autonomous

components that can act and interact with flexibility to collectively achieve their set objectives. While complex systems are decomposable in such a way that it is natural to design agents working with each other from bottom up for overall system functionality, complexity often goes beyond what can be accurately foreseen in advance. Agent interaction and autonomy enable independent decision making at runtime and collective intelligence through cooperation, negotiation, and compromises.

Narrowly speaking, multi-agent systems are useful for modeling decentralized information retrieval, service location, and expert finding[14] in various information networks. Particularly, referral systems for expert finding have attracted increasing research attention. Kautz et al. (1997b) observed that much valuable information was not kept on-line for issues such as privacy and yet this information might be provided when the right people were asked (Kautz et al., 1997a; Yu and Singh, 2003). The fact that people shared information about experts through word-of-mouth motivated researchers to study automated information filtering and expert location based on referral chains (Shardanand and Maes, 1995; Kautz et al., 1997a; Foner, 1997).

Kautz et al. (1997a) developed the *ReferralWeb* system for automatically finding experts through social networks. With a vision of multi-agent systems, the authors used the co-occurrence of name in close proximity from Web sources to reconstruct social networks and focused on utilizing collective intelligence of a networked community, similar in spirit to collaborative filtering. *ReferralWeb* prototypes demonstrated the potential of expert finding through referral chains and provided useful results on referral accuracy and responsiveness (Kautz et al., 1997a).

Searching social networks for experts has attracted increased research attention and agent techniques have been extensively used for this purpose. Foner (1997) recognized

[14]Expert finding, in the context of this survey, is essentially a task of searching for relevant information collections representative of individual agents' (and their human principles') expertise.

the challenge of finding experts because many are not known to the public and developed the *Yenta* multi-agent system for matchmaking, which, through referrals, identified people with similar interests and introduced one to another. While functioning in a decentralized fashion, agents grouped themselves into clusters of related topics, which, in turn, facilitated agent communications for common interests. Provided the local constraint of an agent knowing a limited number of neighbors, Yenta-Lite demonstrated computational efficiency (in terms of network traffics) for referral-based matchmaking.

Research on multi-agent systems has supported development in service-oriented computing, making possible aggregation of dynamic information and services across enterprises and on the Web. According to Georgakopoulos and Papazoglou (2009), service-oriented computing represents a world of loosely coupled cooperating services in which systems can autonomously and dynamically adapt to changes. Seeing multi-agent systems and service-oriented computing as deeply coupled, Huhns et al. (2005) envisioned pervasive service environments in which such applications as heterogeneous information management and mobile computing are supported, and computational service mechanisms that enable dynamic interactions with active services.

Singh et al. (2001) contrasted the ideas of intelligent networks and "stupid networks," and observed a trend toward more distributed information sources and services in communication networks. The authors focused on the automatic location of good, trustworthy services in an open environment of autonomous, heterogeneous, and dynamic components – a stupid network without central control. A referral approach to service location was proposed and studied. With the help of software agents, human principals of the networked community were able to assist each other for locating quality services. While agents explored the environment through interactions, they learned about each other through evaluations of expertise (the ability to provide good service) and sociability (the ability to provide good referrals).

Simulations on $20-60$ agents showed that agent interactions and learning improved the quality of the network for service location over time, which stabilized at an improved quality level, while new peers joining the network drifted toward neighbors who helped (Singh et al., 2001). The existence of pivot agents, or higher out-degree agents with potential weak ties connecting subcommunities, significantly improved the network quality for service location. Clustering also had an impact on the location of services – results showed that network quality decreased with increased clustering. Singh et al. (2001) reasoned that clustering tended to increase the distance to useful experts because more links are used up within a small community. According to Kleinberg (2000b, 2006a), a balance should be maintained in order for searches to efficiently traverse a network. In highly clustered networks, long-distance connections are rare for searches to jump. On the other hand, too many remote connections will disorient a search from gradually moving toward the target, especially when it comes near.

Yu and Singh (2003) developed *MARS*, a multi-agent referral system prototype, and conducted experimental simulations on a co-authorship network of about $5,000$ scholars in the area of artificial intelligence (AI) with a task to find expert scholars on given topics. The effects of branching factor (F, width of search) and referral depth were studied under settings of learning and no learning. Results showed that learning improved expert finding effectiveness (the number of experts found) and efficiency (the number of referrals per expert) in dynamic environments. While both the branching factor and referral depth had a positive impact on the findability of experts, the effect of the branching factors converged at $F = 4$. The focus of the study was on intelligent referral flooding to reach a good number of experts. Yu and Singh (2003) also experimented on minimizing the referral graph by selectively sending a query to the best candidate and so forth.

Zhang and Ackerman (2005) studied strategies for expert finding in social networks

and considered three categories of characteristics, namely, social connections (e.g., the number of neighbors/friends), strength of association, and relevance to desired expertise (e.g., individual expertise and sociability, see also Yu and Singh, 2003). The study identified eight strategies based on these characteristics and compared them through agent simulations on the Enron email dataset containing 147 accounts. Results showed that while most strategies worked effectively, out-degree based strategies outperformed the others due to the existence of well connected nodes. Particularly, the Hamming Distance Search (HDS), which picked the neighbor with the most uncommon social connections from the current agent and favored neighbors with high out-degrees, produced superior results in terms of success rate (effectiveness) and the number of agents involved in searches (efficiency).

The works above demonstrated the usefulness of multi-agent simulation for distributed expert finding and/or service location. Multi-agent systems can also be naturally applied to the study of peer-to-peer systems, in which peers, seen as agents, have individual objectives and assume some degree of independence and autonomy (Androutsellis-Theotokis and Spinellis, 2004; Lua et al., 2005). Research on peer-to-peer information retrieval was often conducted using a multi-agent framework (e.g., Zhang et al., 2004; Kim et al., 2006; Zhang and Lesser, 2006, 2007).

Some researchers used multi-agent systems to model distributed information retrieval in semantic overlay peer-to-peer networks and focused on federated IR operations such as resource representation, database selection, and result fusion in P2P environments (Zhang et al., 2004; Fischer and Nurzenski, 2005; Bender et al., 2005; Vouros, 2008). Some studied agent learning and adaptation for efficient retrieval in dynamic environments, and emphasized the overall system utility and throughput (Zhang and Lesser, 2006, 2007). Others employed multi-agent techniques to build recommender systems based on agent-user and agent-agent interactions (Birukov et al.,

2005). In addition, complex network modeling often relied on agent simulations under the assumptions of local intelligence without global control (Albert and Barabási, 2002; Adamic and Adar, 2005; Kleinberg, 2006a; Simsek and Jensen, 2008). Studies about peer-to-peer information retrieval in Section 2.3 and complex network simulations in Section 2.4 are within the scope of this section and compatible with discussions here.

2.5.4 Incentives and Mechanisms

As noted in previous research on complex networks, many social networks are theoretically searchable but success depends heavily on individual incentives (Milgram, 1967; Watts et al., 2002; Dodds et al., 2003). Provided the autonomous nature of agents and different objectives of participants in information sharing networks, there is no guarantee that each search query will reach the target even when it is algorithmically reachable. Proper incentive mechanisms are needed to ensure good behaviors of individual agents and a network's overall utility (Yu et al., 2003; Kleinberg and Raghavan, 2005; Kleinberg, 2006b).

Yu et al. (2003) observed problems of network congestion and performance degradation caused by uncontrolled free riding in P2P networks and reasoned that agent-based system design should take into account rationality of individuals. Yu et al. (2003) focused on mechanism design for incentives in referral systems. Two micropayment protocols, namely, the fixed pricing and dynamic pricing mechanisms, were introduced to charge agents for queries they posted and reward them for referrals or answers they gave. Experiments showed that free riders, without any contribution and therefore reward, could not survive either payment protocol. Agents had to help others in order to get helped in the long term. Further experiments also demonstrated the potential of such mechanisms to guide price adjustment for high-quality services.

Seeing networks as market places and information as goods, Kleinberg and Raghavan (2005) formulated a model for query incentive networks, in which information seekers posted queries with incentives for answers that were propagated along referral paths. As each node expected to take some portion of the reward by passing the query on to the "right" answer (i.e., the one that was eventually chosen), the incentives shrank in the branching propagation tree until it reached an answer or a balance of zero. Provided answer rarity n and network structure for propagation, Kleinberg and Raghavan (2005) examined how much initial incentive was needed and showed that initial utility of $O(\log n)$ sufficed for a large branching parameter $b > 2$ to cheaply find answers. For $b < 2$, much greater investment was needed from the node originated a query.

Apparently, for larger branching factors, there is a larger cost associated with the number of agents involved in the searching process and therefore greater communication traffics. Kleinberg and Raghavan's (2005) model was query-centric and did not consider the overall network throughput as one objective in the incentive design. It is very likely the result will be different if this is taken into account. With a similar model, Li et al. (2007) compared query efficiency of the incentive mechanism with existing methods but left out the depth-first search (DFS) or greedy routing approach for, arguably, its long response time. Experiments showed superior system utility based on the incentive design. Nonetheless, Li et al. (2007, p. 275) did acknowledge that DFS or greedy routing "undoubtedly outperforms the others in terms of system utility." The argument about greedy routing's inferior responsiveness because of its sequential nature remains arguable in open environments, as was briefly discussed in Section 2.3.3 (see also Lv et al., 2002a,b; Cooper and Garcia-Molina, 2005).

While incentive design often involves payment in the sense of reward, some P2P applications have included paying mechanisms for legal requirements (e.g., for users to pay for music downloads). For example, Yang and Garcia-Molina (2003) developed *PPay*,

an alternative to centralized broker based micro-payment systems, with an aim to distribute transaction load among peers while maintaining sufficient security. Mechanisms were designed to prevent frauds and to punish cheaters.

Besides incentives, security, trust, and privacy are especially important for systems without centralized authority and control. Singh et al. (2001) discussed the impact of community-based service location in the perspective of trust management, in which security techniques do not guarantee the accountability of peers even when they are authenticated. While it is too challenging for a centralized system to manage all trust related aspects, the problem "must be handled from the edges of the network where different parties can build their reputations for trustworthiness in an application-specific or community-specific manner" (Singh et al., 2001, p. 54). Research has studied related issues through social network analysis and decentralized reputation management (e.g., Sabater and Sierra, 2002), distributed policy specification and management (e.g., Udupi and Singh, 2007), and self-organization and referral exchanges (e.g., Yolum and Singh, 2005), etc. Although not the focus of this survey, these issues have impacts on whether agents will behave as expected and how the entire system can perform in a manner within set objectives to support findability.

2.5.5 Conclusion

Dynamics and heterogeneity of large networked environments require information systems span organizational boundaries and work with one another in the absence of global control. Multi-agent systems provide a new paradigm in which a complex system – an information retrieval system in particular – can be naturally decomposed into autonomous, heterogeneous, and cooperative components to cope with the complexity and unpredictability (Jennings, 2001). A monolithic, centralized model is not capable

of managing the complexity of today's distributed, dynamic, and heterogeneous information space. Baeza-Yates et al. (2007) argued for fully distributed search engines for high quality answers, fast response time, high query throughput, and scalability. A multi-agent system approach to information retrieval in such environments is indeed needed. Multi-agent systems offer an integral view in which research on Web IR, distributed and peer-to-peer retrieval, and complex networks can all be discussed. It additionally brings perspectives on designing mechanisms for incentives, trust, privacy, and security in open environments.

2.6 Summary

This literature review discussed the general problem of finding information in distributed networked environments and surveyed research areas in information retrieval (IR), Web and distributed IR, peer-to-peer content sharing and search, complex networks and their navigability, and the multi-agent paradigm for IR. While traditional IR and distributed IR research provides classic tools for attacking the problem, the evolving dynamics and heterogeneity of today's networked environments have challenged the sufficiency of classic methods and call for new innovations. Whereas peer-to-peer offers a new type of architecture for application-level questions and techniques to be tested (Croft, 2003), research on complex network studies related questions in their basic forms (Albert and Barabási, 2002). Table B.1 in Appendix B summarizes in a matrix major research problems and example frameworks in the areas being surveyed.

Seen from the agent perspective of cooperative information systems (Huhns, 1998), the actionable information view of the Semantic Web (Berners-Lee et al., 2001), or the interaction-centric perspective of human information interaction (Marchionini, 2008), the problem of information findability and its scalability becomes crucial. In an open, dynamic information space such as the Web, people, information, and technologies are all mobile and changing entities. The classic view of "knowing" where information is and indexing "known" collections of information for later retrieval is hardly valid in such environments. Finding where relevant repositories are for the *live* retrieval of information is critically needed. Without global information, new methods have to rely on local intelligence of distributed peers and/or their delegates to collectively find a way to desired information. Multi-agent systems provide an important paradigm and tools to attack the problem.

Scalability of findability is about the cost of traversing a network to reach desired information. Unstructured or semi-structured peer-to-peer networks, being widely

studied, represent a connected space self-organized by individuals with local objectives and constraints, exhibiting a topological underpinning on which all can collectively scale (Amaral et al., 2000; Lua et al., 2005). While the small world phenomenon displays a connected world in which every one (and every piece of information) is within reach, research suggested there are certain structural characteristics to guide searches (Kleinberg, 2000b; Watts et al., 2002; Liben-Nowell et al., 2005; Simsek and Jensen, 2008; Boguñá et al., 2009).

Research based on abstract network models has produced exciting results for both findability and scalability – it has been demonstrated that short paths to desired targets can be found even in networks of a billion nodes. Information retrieval in networked environments, nonetheless, has been more complex than that. Not only is the dimensionality of a "hidden" (search) space difficult to define (Kleinberg, 2000b; Yu and Singh, 2003; Boguñá et al., 2009), the ambiguity of relevance further complicates the problem. Due to local constraints, relevance has to be seen from individual perspectives and a global measure of it cannot be enforced to guide searches. IR research in distributed networked environments, with tools from peer-to-peer and multi-agent research, has produced promising results on finding (or *recalling*) relevant information (Bawa et al., 2003; Crespo and Garcia-Molina, 2005; Zhang and Lesser, 2006; Lu and Callan, 2007). The scalability of findability, however, requires further scrutiny.

To further illustrate the point, Figure 2.9 samples findability and scalability results from previous research on complex networks, peer-to-peer, and multi-agent IR. Search experiments based on abstract models and synthetic networks have shown useful results on very large scales. Kleinberg (2000b), for example, conducted simulations on four hundred million nodes in which unique targets were found in roughly one hundred steps (note the top right data point on Figure 2.9). Experiments on real IR data (e.g., TREC collections) were typically concentrated on recall and less about findability of

very specific items. In other words, even when dealing with large networks[15], queries used in the experiments were often so broad that they had a large relevance base. In general, relevant documents appeared in many segments and finding one of them was not a huge challenge (see solid points of small rarity N_R values on the bottom left of Figure 2.9).

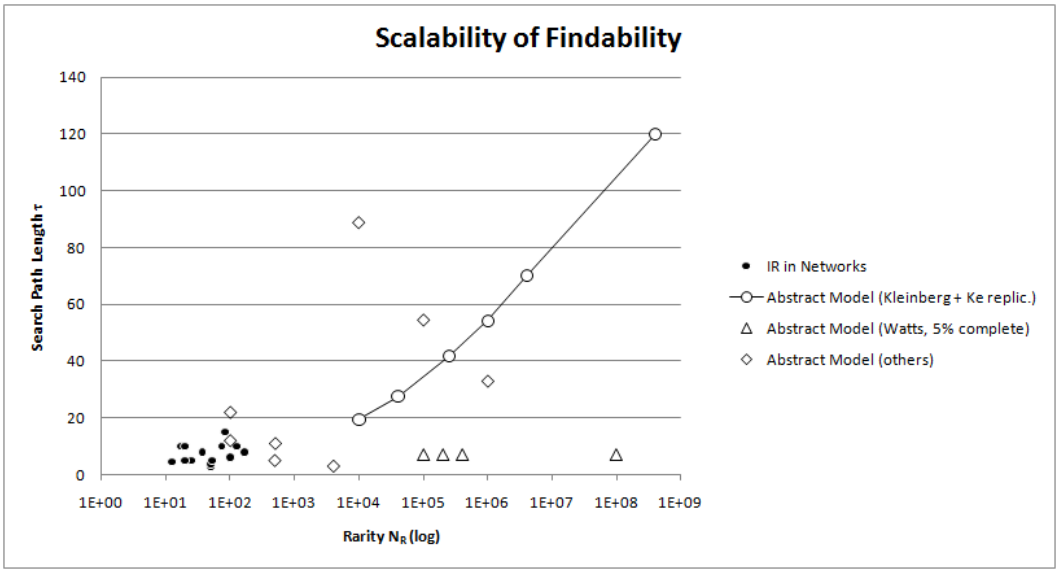


Figure 2.9: Summary of Existing Findability/Scalability Results. The X axis denotes log-transformed rarity: $N_R = N/N_{rel}$, where N is the total number of peers and N_{rel} the number of all relevant or target peers. This represents the average size of a peer population for ONE relevant/target peer to appear. The larger the rarity N_R, the more difficult it is to find one target. The Y axis denotes the path length, or number of peers involved, for finding ONE target (first if there are more than one). Data can be found in Table C.1 of Appendix C.

Serving diverse users in an open, dynamic environment implies that some queries are likely to be narrowly defined. Calvin Mooers' (1951) statement about information being painful was a realization that humans have limited ability to process voluminous information and often tend to avoid it. It has long been observed that people rarely demand high *recall* – a couple of highly relevant items often suffice even when many more are presented (Cleverdon, 1991; Zobel et al., 2009). Finding highly relevant information

[15]The CiteSeer dataset used in Bawa et al. (2003) had more than eighty thousand sites or collections. Lu and Callan (2007) had twenty five thousand sub-collections from .GOV2.

in large distributed environments poses great challenges and offers potential rewards.

Chapter 3

Research Angle and Hypotheses

> Although many fads have come and gone in complexity, one thing is increasingly clear: Interconnectivity is so fundamental to the behavior of complex systems that networks are here to stay. – Barabási 2009

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 broadly studied.

3.1 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 hidden web where many large databases reside, and networks formed by information agents. Close examination of these networks reveals some common characteristics illustrated in Figure 3.1.

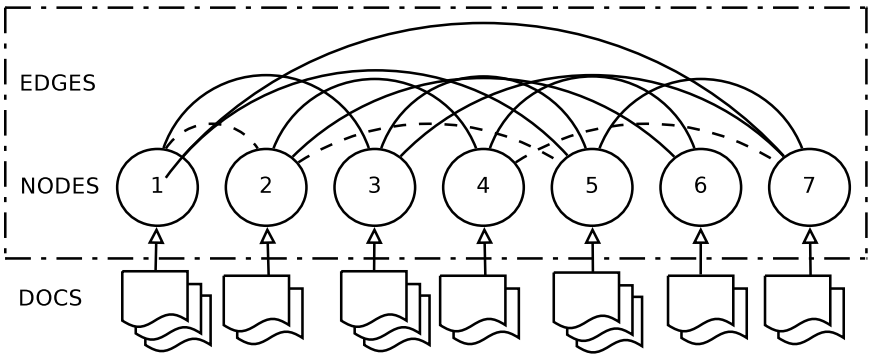
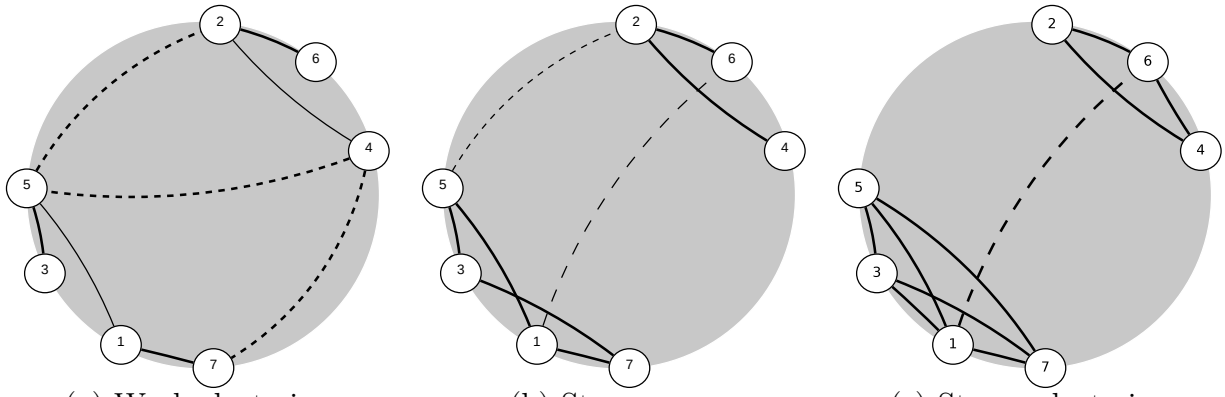


Figure 3.1: Information Network

As shown in Figure 3.1, an information network is formed by nodes (e.g., peers, web sites, or agents) through edges, e.g., through network communication/interaction/links. A node has a set of information items or documents, which in turn can be used to define its topicality or expertise. If we can somehow discover the content of each node and layout the nodes in terms of their topicality, then the information network in Figure 3.1 can be visualized in the form of Figure 3.2 (a).

Figure 3.2 (a) shows a circle representation of the topical (semantic) space, in which



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

Figure 3.2: Evolving Semantic Overlay

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.

Long-distance connections, shown as dashed lines in Figure 3.2, bring randomness to the network. When there are many long-range connections, the topological (network) space tells little about the topical 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 3.2 (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 3.2 (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 connectivity – even if two nodes are physically connected, semantic overlay may maintain a probability function that unlikely allows them to contact each other for search.

3.2 Clustering Paradox

Semantic overlay discussed above is essentially a type of *clustering*, which is the process of bringing similar items together. Research has found *clustering* on 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 rarely been scrutinized.

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, 2000b; 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. With very strong clustering, however, a network tends to be fragmented into local communities with abundant *strong ties* but few *weak ties* to bridge remote parts (Granovetter, 1973; Singh et al., 2001). Although searches might be able to move gradually toward targets, necessary "hops" become unavailable.

We refer to this phenomenon as the *Clustering Paradox*, in which neither strong clustering nor weak clustering is desirable. 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.

3.2.1 Function of Clustering Exponent α

One key parameter/variable in complex network research for decentralized search is the clustering exponent α. Kleinberg (2000), who pioneered this line of research, 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 (Kleinberg, 2000b). 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*[1]. It was shown that only when *clustering exponent* $\alpha = 2$, search time (i.e., search path

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

length) was optimal and bounded by $c(\log N)^2$, where N was the network size and c was some constant (Kleinberg, 2000b).

The *clustering exponent* α, as shown in Figure 3.3, describes a correlation between the network (topological) space and the search (topical) space (Kleinberg, 2000b; Boguñá et al., 2009). When α is small, connectivity has little dependence on topical closeness – local segments become less visible as the network is built on increased randomness. As shown in Figure 3.4 (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 3.4 (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 3.4 (b), the network becomes a narrowly defined *small world*, in which both local and remote connections present.

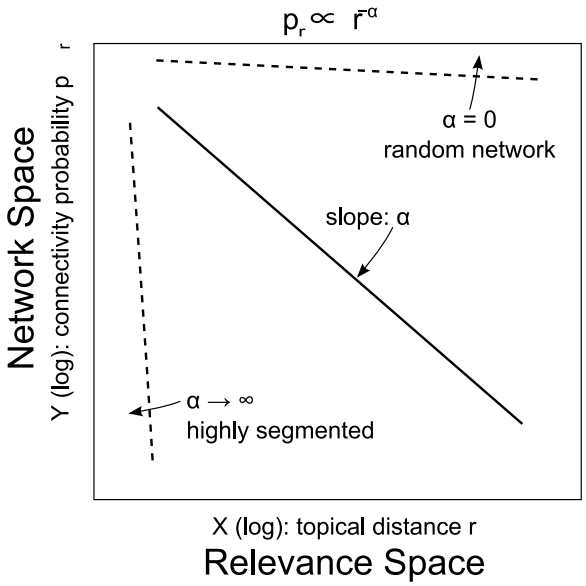
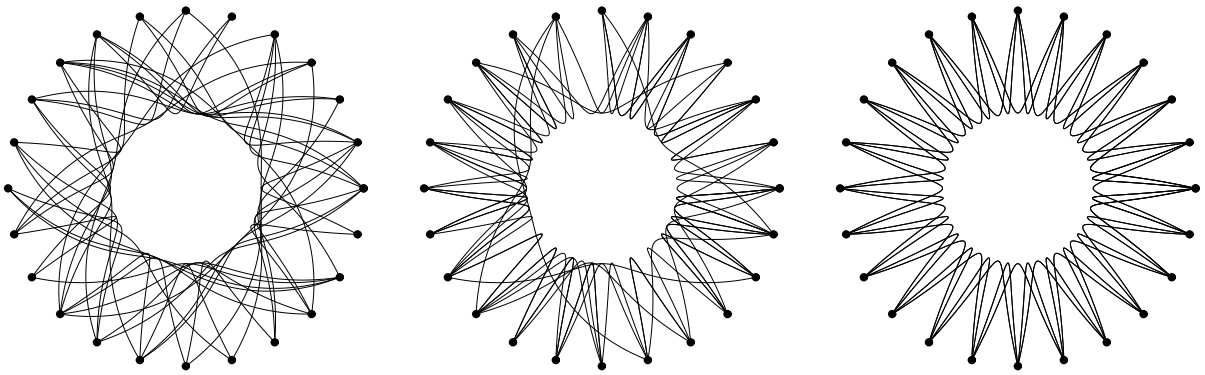


Figure 3.3: Network Clustering: Function of Clustering Exponent α

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



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

Figure 3.4: Network Clustering: Impact of Clustering Exponent α. Compare to Watts and Strogatz (1998). (a) a random network, provided no association between connectivity 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.

similar to that of a rewiring probability $p \in [1, 0]$ in Watts and Strogatz (1998). However, α additionally defines the association of connectivity 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). However, the results were primarily produced by research on low dimensional synthetic spaces using highly abstract models.

In a decentralized expert finding context, we observed some patterns of the *Clustering Paradox*, in which either strong clustering or weak clustering led to degraded search performance (Ke and Mostafa, 2009). More critically, the Clustering Paradox appeared to have a scaling effect. Although overclustering only moderately degraded search performance on small networks, it seemed to cause dramatic performance loss for

large networks. In other words, little performance disadvantage in small networks might become too big to ignore in large-scale systems. All this requires further scrutiny in experiments on benchmark IR data collections. In addition, how the clustering paradox interplays with other variables such as degree distribution remains to be studied.

3.3 Search Space vs. Network Space

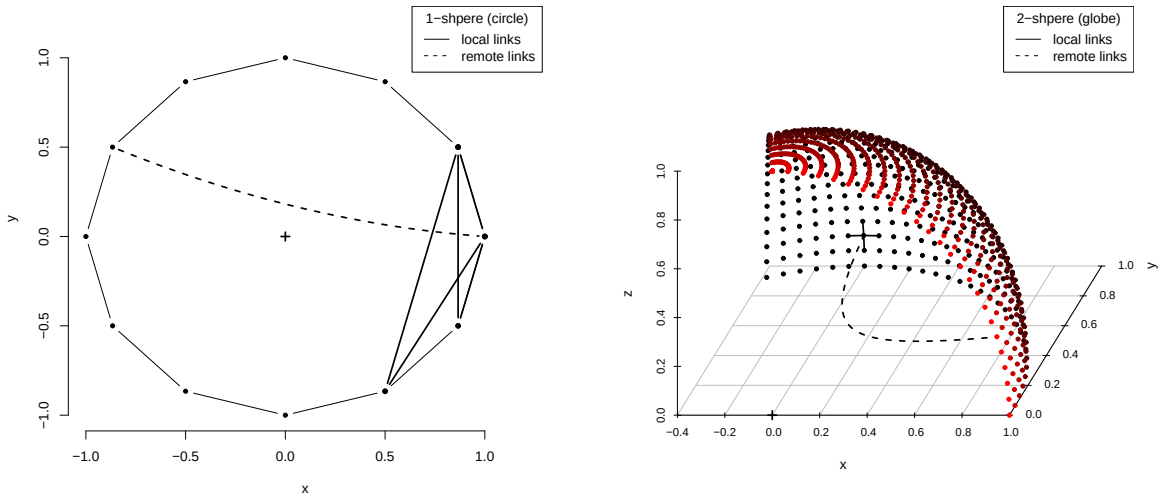
As discussed earlier, if queries are to traverse the topological (network) space to find topical relevance (in the search space), some association between the two spaces is required to guide searches. The clustering paradox, if applicable in the IR context, indicates that some balance of network clustering supports best mapping of the topological space to the topical space, potentially enabling optimal retrieval performance. It is therefore important to examine the two spaces to figure out what additional variables should be considered.

3.3.1 Topical (Search) Space: Vector Representation

The topical (search) space is about how nodes can be represented in terms of information they possess and how relevant they are to each query. Salton et al. (1975) proposed the Vector Space Model (VSM) in which queries and documents are represented as n-dimensional vectors using their non-binary term weights (see also Baeza-Yates and Ribeiro-Neto, 2004). This dimensional view potentially enables us to build a connection between the IR challenge in this research and general results from previous studies on complex networks (e.g., Kleinberg, 2000b; Watts et al., 2002).

In the dimensional space for IR, the direction of a vector is of greater interest than its magnitude. The correlation between two information items (e.g., a query and a document) is therefore quantified by the cosine of the angle between two corresponding

vectors. Provided the irrelevance of vector length, all vectors can be normalized to a common distance from the origin, resulting in a hypersphere representation of documents and nodes. Figure 3.5 (a) and (b) illustrate a 1-sphere (2D circle) and a 2-sphere (3D globe), given all vector lengths normalized to 1.



(a) 1-Sphere (Circle) (b) 2-Sphere (Globe)

Figure 3.5: Hypersphere Representation of Search Space. Each node is topically represented by a vector from the origin (a solid point in the figures). Vector lengths are normalized to 1 because only vector direction matters. Both figures illustrate *local connections* with close or topically similar neighbors and *remote connections* with topically distant nodes.

Terms can be used as dimensions and frequencies as dimensional values in VSM. Yet a more widely used method for term weighting is *Term Frequency * Inverse Document Frequency* (*TF*IDF*), which integrates not only a term's frequency within each document but also its frequency in the entire representative collection (Baeza-Yates and Ribeiro-Neto, 2004). The reason for using the *IDF* component is based on the observation that terms appearing in many documents in a collection are less useful. In the extreme case, useless are stop-words such as "the" and "a" that appear in every English document.

Among other limitations, VSM usually uses single terms without examining proximity and co-occurrence patterns for their semantic meanings. While existing models

often assume term independence, generalized VSM and latent semantic indexing (LSI) techniques acknowledge the non-orthogonality of natural language terms and project the observed term space to a smaller dimensional space to improve retrieval effectiveness (Wong et al., 1987; Landauer et al., 1988; Deerwester et al., 1990). VSM succeeded in its simplicity, efficiency, and superior results it yielded with a wide range of collections (Baeza-Yates and Ribeiro-Neto, 2004).

In the proposed research, we plan to use the Vector Space Model for document and query representation. Given that a node is more than one single document but rather a collection of documents, strategies are needed for aggregation of individual representations. A widely used strategy in distributed information retrieval is document frequency based collection representation (Callan et al., 1995; Phan et al., 2000). A node can be seen as a metadocument represented by terms using their document frequencies, i.e., in how many documents each term appears.

3.3.2 Topological (Network) Space: Scale-Free Networks

To facilitate searching, many peer-to-peer IR systems used hierarchical structures with central/regional servers as fast channels that connected various remote parts (e.g., Bawa et al., 2003; Fischer and Nurzenski, 2005; Lu and Callan, 2007; Doulkeridis et al., 2008). Nonetheless, most real world networks, very different from hierarchical structures, manifest small world, scale free (or broad scale), and highly clustering properties for potential efficient searching (Albert and Barabási, 2002; Kleinberg, 2006a). These network structures, produced under individual peer capacities and constraints, have revealed to us how peers can collectively scale given how much they individually can afford to do.

Many small world networks follow a *power-law* degree distribution, deviating from a

Poisson distribution exhibited in random networks. In a power-law network, the distribution of connectivity decays with a power law function linear on log-log coordinates. Intuitively speaking, in a power-law network, while some nodes are highly connected (rich), the majority of nodes have a small number of connections (poor). So far, power-law networks have been well explained by the *Scale Free*[2] model, in which network growth and preferential attachment are both essential (Barabási and Albert, 1999).

Many complex networks exhibit a high degree of robustness. Because of redundant wiring of network structure, local failures rarely lead to global reduction of network capacity. At the topological level, simulation experiments and analytical results showed that scale-free networks are more robust against random local failures than random networks do (Albert and Barabási, 2002). However, they are more vulnerable to attacks targeted on highly connected nodes.

The common presence of scale-free networks and their mathematical simplicity allow researchers to study complex problems in a very systematic way. Given a constant average degree and range, the power-law exponent γ (i.e., the slope value on a log-log distribution plot) is the only variable needed to control the distribution. We will investigate the impact of degree distribution on search performance. We will also propose degree-based search methods and study their effectiveness and efficiency under various experimental settings.

3.4 Strong Ties vs. Weak Ties

In the *Clustering Paradox*, *strong ties* and *weak ties* play important roles. According to Granovetter (1973), *strong ties* were widely studied in network models for small, well-defined groups in which individuals have strong neighborhood overlap and are similar

[2]The scale free model is by far the most effective approach to explaining the emergence of power-law networks. In this research, we use the terms *power-law network* and *scale-free network* exchangeable.

to one other. Emphasis on *weak ties*, however, shifts the discussion to relations *between* groups and to analysis of "segments of social structure not easily defined in terms of primary groups" (Granovetter, 1973, p. 1360). *Weak ties* often serve as bridges of groups, removal of which will lead to fragmented larger structures.

For clarification and operationalization purposes, in this research, the *strength* of a tie – the meanings of strong vs. weak ties – will be defined on three levels, namely, 1) the dyadic meaning in terms of the relationship of interaction between two nodes, 2) the topological meaning in terms of a tie's macro-level impact on the network structure, and 3) the topical definition based on pairwise similarity/relevance in the IR context. These three levels will enable us to scrutinize network clustering from multiple perspectives.

3.4.1 Dyadic Meaning of Tie Strength

Granovetter (1973, p. 1361) loosely defined the *strength* of an interpersonal tie as "a combination of the amount of time, the emotional intensity, the intimacy (mutual confiding), and the reciprocal services which characterize the tie." While implications of tie strength are beyond the dyadic characteristics of an interpersonal relationship, it is still useful to define it on a similar level in the decentralized IR context, in which interactions and trust among distributed nodes (agents) are important aspects. The strength of a tie, on the dyadic level of this research, is thus defined as a combination of time, mutual trust of two nodes (agents) and the value of help they have offered each other. It can be operationalized as the number of times they interact with each other and rewards exchanged in interactions.

3.4.2 Topological Meaning of Tie Strength

Whereas strong ties are unlikely to be bridges, all bridges are weak ties. Following the "bridge" notion of tie strength, the *weakness* of a tie was referred to as the number of

broken paths or changes in average path length due to its removal (Granovetter, 1973). More precisely, it can be defined as a bridge of *degree* n_d, where n_d is the shortest path between its two points if the tie is removed. Besides this, the *betweenness centrality* measure, developed by Anthonisse (1971); Freeman (1977), can also be used to evaluate node or tie centrality/*weakness*:

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{3.1}$$

where $\sigma_{st} = \sigma_{ts}$ is the number of shortest paths from s to t and $\sigma_{st}(v)$ the number of shortest paths from s to t that pass through v (either a tie or a node) in graph V (see also Brandes, 2001; Girvan and Newman, 2002).

3.4.3 Topical Meaning of Tie Strength

In the IR context, closeness or remoteness of two nodes depends on their topical relevance or similarity. Provided the vector representation, distance can be measured by the angle of two vectors and similarity measured as cosine of the angle (Baeza-Yates and Ribeiro-Neto, 2004). On this level, therefore, the strength of a tie is defined as the pairwise relevance and operationalized as cosine similarity. Given two nodes represented by vectors $u = [u_1, .., u_t]^T$ and $v = [v_1, .., v_t]^T$, if they form a tie/link, the strength can be calculated using cosine coefficient defined in Section 4.2.1. Thereby, tie *weakness* can be equated with pairwise topical distance or angle value: $\angle_{uv} = \arccos(c_{uv})$, where c_{uv} is the cosine coefficient of vectors u and v.

$$c_{uv} = \cos(u, v) = \frac{\sum_{i=1}^{t} x_i \cdot y_i}{\sqrt{(\sum_{i=1}^{t} x_i^2) \cdot (\sum_{i=1}^{t} y_i^2)}} \tag{3.2}$$

Here we present three levels of tie strength, namely, the dyadic, topological, and

topical meanings of strong vs. weak ties. They are operationlizable metrics, in addition to the clustering exponent α, that can be used to scrutinize network clustering. Potentially, these angles will help us analyze experimental results and understand what is going on in a network community and why searches do or do not perform well.

3.5 Hypotheses

Earlier discussions provide evidence for potential hypotheses. In sections 3.2 and 3.2.1, 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.

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 distribution has an impact on network optimization for search – that is, different distributions may require different network clustering*

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

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

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 (e.g., Adamic et al., 2001; Boguñá et al., 2009). We therefore conjecture that:

Hypothesis 4 *In large scale 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.*

Chapter 4

Simulation System and Algorithms

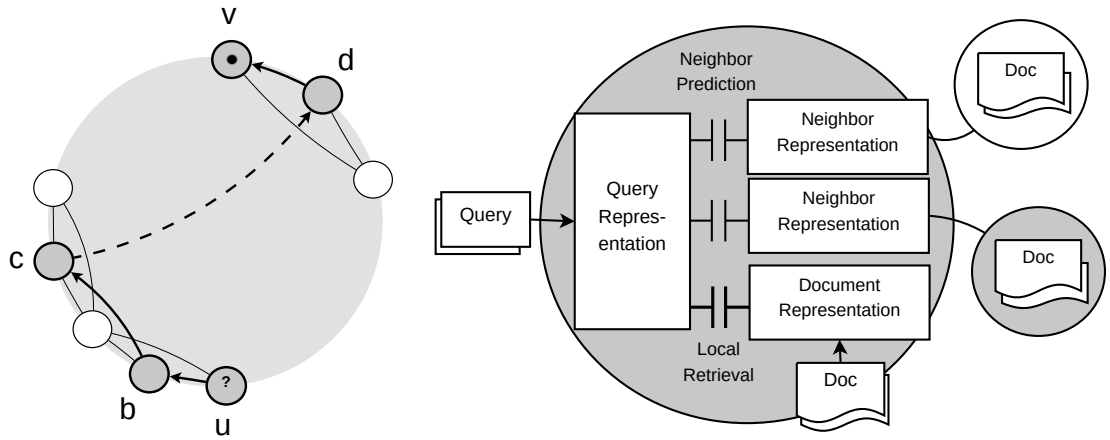
The problem of decentralized search in networks is too complex to be studied in a top-down manner. In this research, we propose to use multi-agent systems for a bottom-up investigation. Jennings and Wooldridge (1998a, p. 4) defined an *agent* as "a computer system situated in some environment, and that is capable of *autonomous action* in this environment in order to meet its design objectives." In Huhns's (1998) terms, an agent is an active, persistent computational entity that can perceive, communicate with others, reason about, and act in its environment.

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 network-based information retrieval system in particular – can be naturally decomposed into autonomous, heterogeneous, and cooperative components to cope with the complexity and unpredictability (Jennings, 2001). A monolithic, centralized model is not capable of managing the complexity of today's distributed, dynamic, and heterogeneous information space. Baeza-Yates et al. (2007) argued for fully distributed search engines for high quality answers, fast response time, high query throughput, and scalability. A multi-agent system approach to information retrieval in such environments

is indeed needed.

4.1 Simulation Framework Overview

Based on multi-agent systems, we have developed a decentralized search architecture named *TranSeen* for finding relevant information distributed in networked environments. We emphasize the societal view of agents who have local intelligence and can collaborate with one another to perform global search tasks. Similar agent-based approaches have been adopted by various research groups to study efficient information retrieval, resource discovery, service location, and expert finding in decentralized environments (Singh et al., 2001; Yu and Singh, 2003; Zhang and Ackerman, 2005; Zhang and Lesser, 2007). One common goal was to efficiently route a query to a relevant agent or peer. We illustrate the conceptual model in Figure 4.1 (a) and elaborate on major components shown in Figure 4.1 (b).



(a) Global View (b) Agent Internal View

Figure 4.1: Conceptual Framework. (a) Global View of agents work together to route a query in the network space. (b) Agent Internal View of how components function within an agent.

Assume that agents, representatives of information seekers, providers (sources), and mediators, reside in an n dimensional space. An agent's location in the space represents its information topicality. Therefore, finding relevant sources for an information need

is to route the query to agents in the *relevant* topical space. To simplify the discussion, assume all agents can be characterized using a two-dimensional space. Figure 4.1 (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. In Figure 4.1 (a), the query traverses a search path $A_u \rightarrow A_b \rightarrow A_c \rightarrow A_d \rightarrow A_v$ to reach the target. While agents A_b and A_d help move the query toward the target gradually (through strong ties), agent A_c has a remote connection (weak tie) for the query to "jump."

Neighbor Selection for Query Forwarding

For decentralized search, direction matters. Pointing to the right direction to the relevant topical space means agents have some ability for query analysis and determine which neighbor(s) to be contacted given a query representation. 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 contact his neighbors for help. Therefore, there should be a mechanism of match query representation with potential *good* neighbors. By *good neighbor*, we mean an agent on a short path to the targeted information space – either the neighbor is likely to have relevant information to answer the query directly or in a neighborhood closer to relevant targets[1]. Agents explore their neighborhoods through interactions and develop some knowledge about who serves and/or connects to what information. The agent environment is assumed to be cooperative – that is, agents are willing to share information about their topicality and connectivity.

[1]See also Singh et al. (2001) and Yu and Singh (2003) for related concepts *expertise* and *sociability*.

Network Clustering for Global Search Guidance

Network topology plays an important role in decentralized search. As discussed earlier, topical segmentation based techniques such as semantic overlay networks (SONs) have been widely used for efficient peer-to-peer information retrieval (Doulkeridis et al., 2008). Through self-organization, similar peers form topical partitions, which provide some association between the topological (network) space and the topical space to guide searches. The clustering paradox, if applicable in the IR context, implies that such an association, in the form of *clustering exponent* α, is critical for efficient navigation in networks (Kleinberg, 2000b; Liben-Nowell et al., 2005; Boguñá et al., 2009; Ke and Mostafa, 2009). The *TranSeen* framework has a mechanism for self-organized rewiring and network clustering, which influences the balance of *strong ties* vs. *weak ties* for efficient routing, as discussed in depth in Section 3.2.1 and illustrated in Figure 3.3. Section 4.2.3 has the algorithmic detail about network clustering.

4.2 Algorithms

In the previous section, we described the *TranSeen* multi-agent framework for decentralized search. Figure 4.1 (b) illustrates how various components work together within each agent. The TranSeen system is being 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 will elaborate on specific algorithms implemented in the TranSeen framework and used in the research. Section 4.2.1 (A) presents the *TF*IDF* weighting

scheme for information representation (to represent documents and queries) while section 4.2.1 (B) discusses a similar method we refer to as *DF*INF* for neighbor (agent) representation. Section 4.2.1 (C) discusses the cosine coefficient for measuring the similarity of two information items. Section 4.2.2 describes five search (neighbor selection) algorithms based on neighbor relevance (similarity) and/or connectivity. Section 4.2.3 elaborates on the function for agent rewiring (clustering) based on *clustering exponent* α and *degree exponent* γ.

4.2.1 Basic Functions

(A) 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 thesaurus is not assumed. Instead, each agent has to process 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 (e.g., a document) is then converted to a numerical vector of terms where term t is computed by:

$$W(t) = tf(t) \cdot log(\frac{N}{df(t)}) \tag{4.1}$$

where $tf(t)$ is the frequency of 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 term t of the term space. We refer to $log(\frac{N}{df(t)})$ as IDF. IDF values were computed within the information space of an agent given no global information. This is to follow the assumption that global information is not available to individuals and it is impossible to aggregate all

documents in the network to get global DF values.

(B) DF*INF Agent Representation

For neighbor (agent) representation, we will 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 metadocument of terms. Distributed IR research has shown DF information useful for collection selection (Callan et al., 1995; Powell and French, 2003). Treating each metadocument as a normal document, it becomes straightforward to calculate *neighbor frequency* (NF) values of terms, i.e., the number of metadocuments (neighbors) that contains a particular term. A metadocument (neighbor) is then represented as a vector where term t is computed by:

$$W'(t) = df'(t) \cdot log(\frac{N'}{nf'(t)}) \tag{4.2}$$

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

(C) 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 :

$$\sum_{t \in q} tf(t) \cdot idf^2(t) \cdot coord(q,d) \cdot queryNorm(q) \tag{4.3}$$

where $tf(t)$ is term frequency of term t in document d, idf(t) the inverse document frequency of t, $coord(q, d)$ a coordination factor based on the number of terms shared by q and d, and $queryNorm(q)$ a normalization value for query q given the sum of squared weights of query terms. The function is a variation of the well-known cosine similarity measure. Additional details can be found in Hatcher et al. (2010); Baeza-Yates and Ribeiro-Neto (2004). 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.

(D) Retrieval Federation/Fusion Method

In some of the search tasks we plan to investigate (e.g., Relevance Search and Authority Search tasks described in Section 5.3), 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 ideas from classic federation models such as DORI and GlOSS (Gravano et al., 1994; Callan et al., 1995; French et al., 1999), we plan to use the following method in our experiments.

First, when a search is done (i.e., a query finishes traversing a network for relevant documents), the method will select top n_s (e.g., 5) systems whose metadocuments are most relevant/similar to the query (based on the DF*INF and similarity scoring functions described above). Each of the selected systems will be 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 metadocument similarity score S_m, the document's similarity score is then normalized to:

$$S'_d = S_d \cdot S_m \tag{4.4}$$

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, e.g., 10) 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 5.5.

4.2.2 Neighbor Selection Strategies (Search Algorithms)

The similarity scoring function above produces 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. These strategies will be tested and compared in experiments.

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.

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 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 findability 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 4.2), to generate the cosine similarity vector S.

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).

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.

4.2.3 System Connectivity 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 main experiments on the ClueWeb09B collection (details in Section 5.1), we collect information about each web site/system's incoming hyperlinks and normalize the in-degrees as their d_u values. We will 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 will be computed by:

$$d_u = d_{min} + \frac{(d_{max} - d_{min}) \cdot (d'_u - d'_{min})}{d'_{max} - d'_{min}} \tag{4.5}$$

where d'_{max} is the maximum degree value in the hyperlink indegree 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 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 current agent (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 (overlay):

$$p_{uv} \propto r_{uv}^{-\alpha} \tag{4.6}$$

where α is the *clustering exponent* and r_{uv} the pairwise topical (search) distance. The finalized neighborhood size will become 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 3.4, large α values lead to highly clustered networks while small values produce random networks with many topically remote connections or weak ties.

Chapter 5

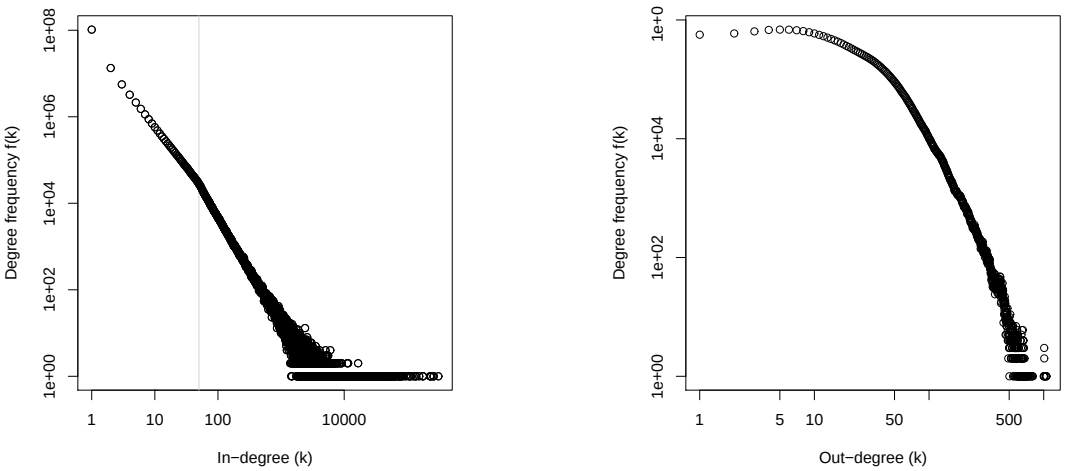
Experimental Design

5.1 Data Collection

We plan to use 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 (25 TB uncompressed) and 8 billion outlinks (71 GB uncompressed) crawled during January - February 2009. The Category B is a smaller subset containing the first crawl of 50 million English pages (1 TB uncompressed) from 3 million sites with 454 million outlinks (3 GB uncompressed). The ClueWeb09 dataset, though new in its first year, has been adopted by several TREC tracks including Web track and Million Query track. 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. Anchor text, however, is not provided as part of the link graph. 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 5.1 (a) in-degree frequency distribution and (b) out-degree distribution (on log/log coordinates). The in-degree distribution has two linear parts on the log/log coordinates, with a cutoff at $k \approx 50$.



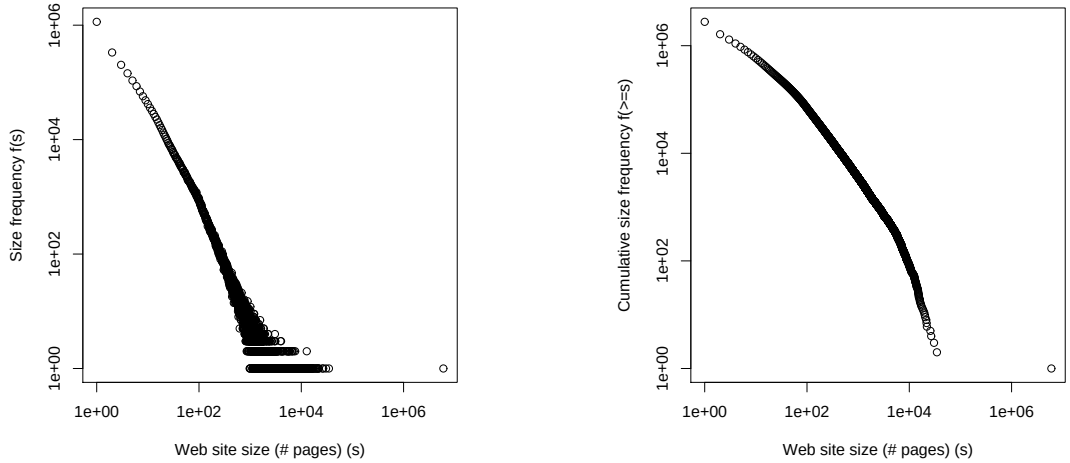
(a) In-degree distribution (b) Out-degree distribution

Figure 5.1: ClueWeb09 Category B Web Graph: Degree Distribution

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 5.2 (a). Figure 5.2 (b) shows the cumulative distribution, in which the Y dimension denotes frequency of web sites with a size $\geq s$ represented on X.

Figure 5.3 (a) 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. Figure 5.3 (b) shows the cumulative form, in which the Y dimension denotes the frequency of page size $\geq l$ represented on X.

We also analyzed the distribution of web pages across major top level domains such as .com and .edu. Figure 5.4 shows major top level domains with the largest numbers



(a) Site size (# pages) distribution (b) Cumulative size distribution

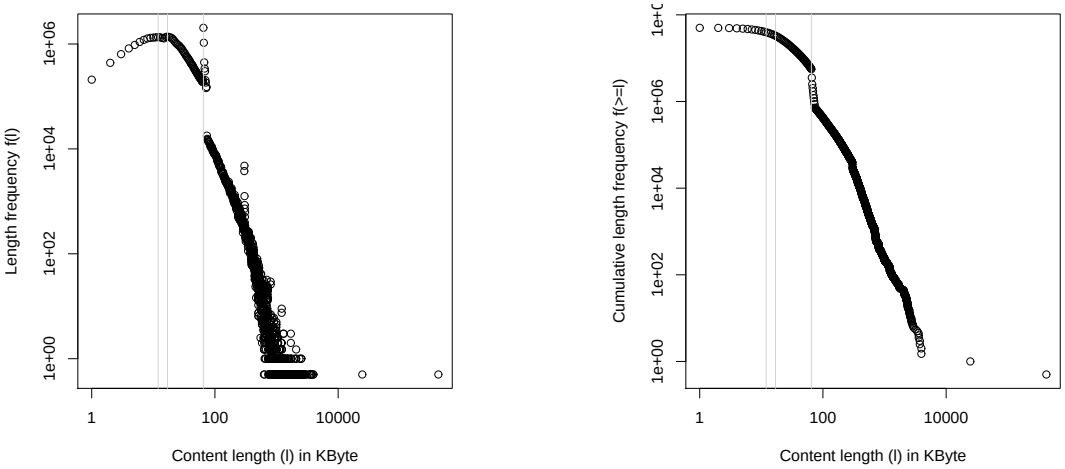
Figure 5.2: ClueWeb09 Category B Data: # pages per site distribution

of web pages. Note that Y is log-transformed.

Another dataset from TREC, namely Genomics track 2004 benchmark collection, is being considered in this research for additional experiments. The data collection is a ten-year subset of Medline from 1994 to 2003, with roughly $4,591,008$ citations containing titles, abstracts, authors, etc. (Hersh et al., 2004). The number of articles in each year is shown in Figure 5.5 (a). There are $808,771$ unique scholars and $17,443,160$ author-article pairs. On average, each scholar (co-)authored five to six articles while each article has roughly three to four authors. Figure 5.5 (b) shows the frequency distribution of scholarly productivity (or the number of articles each scholar published) in the TREC Genomics collection. Probably due to name ambiguity, there are several authors who published more than one thousand papers (bottom-right of Figure 5.5 (b)).

5.2 Network Model

Based on the TREC data collections, two types of networks can be constructed, namely, document networks and agent (system) networks. The primary focus of the proposed study is on decentralized search in networks where information is hosted by distributed



(a) Page size (length) distribution (b) Cumulative size distribution

Figure 5.3: ClueWeb09 Category B Data: Page length distribution

systems/agents. Hence, experiments will conducted on the Agent Network (AN) model described here. We provide information about additional models that can be used in future studies in Appendix E.

In the Agent Network (AN) model, each agent represents an IR system serving a collection of multiple 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 web data such as the ClueWeb09 collection, we can simply treat a web site as an agent and use hyperlinks between sites to construct the initial network. For a bibliographical dataset such as the TREC Genomics 2004, we can treat a scholar/author as an agent hosting articles they have published and use collaboration data (e.g., co-authorship) to establish the initial network topology. Network clustering will then be performed using the method described in Section 4.2.3.

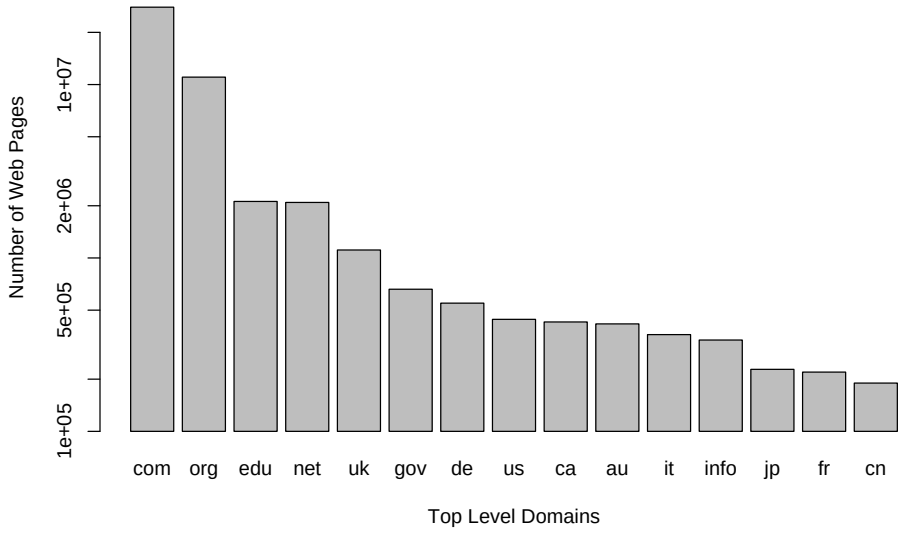


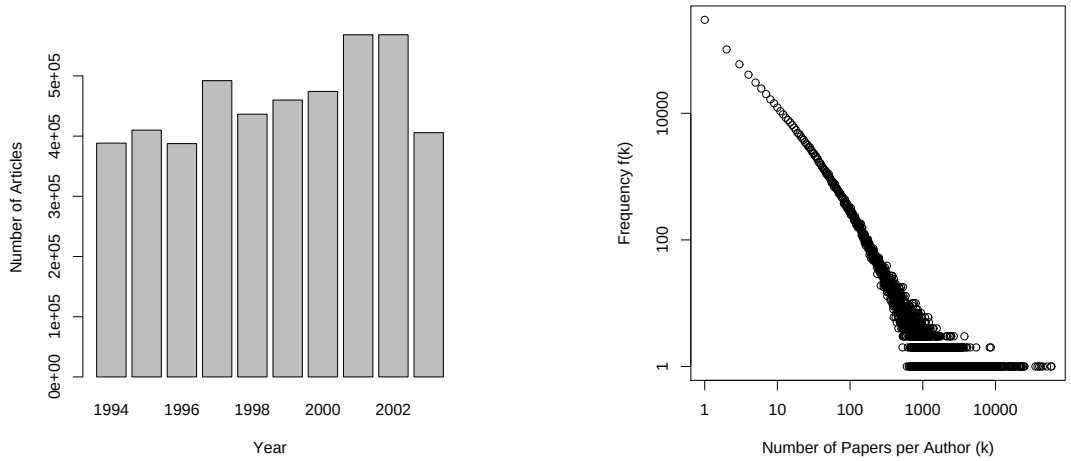
Figure 5.4: ClueWeb09 Category B Data: # web pages per top domain

5.3 Task Levels

Given the large size of TREC data collections to be used, it is nearly impossible to manually judge the relevance of every document and establish a complete relevance base. Hence, we will rely on existing evidence in data to do automatic relevance judgment. We plan to 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. We elaborate on the three levels below.

5.3.1 Task Level 1: Threshold-based Relevance Search

The first level involves finding documents with relevant information. Relevant docuemnts 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



(a) Yearly distribution of # articles (b) # article per author distribution

Figure 5.5: TREC Genomics 2004 Data Distributions

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.

5.3.2 Task Level 2: Co-citation-based Authority Search

The second task level involves finding agents that are best "regarded" as relevant to the 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. Agents who host one or more of such documents are therefore considered relevant to the query. 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 (e.g., Dean and Henzinger, 1999). This task level, relying on co-citation patterns as relevance/authority judgment, is potentially more objective but

challenging than the first level. It can also be seen as popular[1] item search because a web document receives many in-links (and co-citations) only when it has achieved some popularity level.

5.3.3 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. However, in a sense, this is a finding-needle-in-haystack task. 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.

5.4 Additional Independent Variables

5.4.1 Degree Distribution: d_{min} and d_{max}

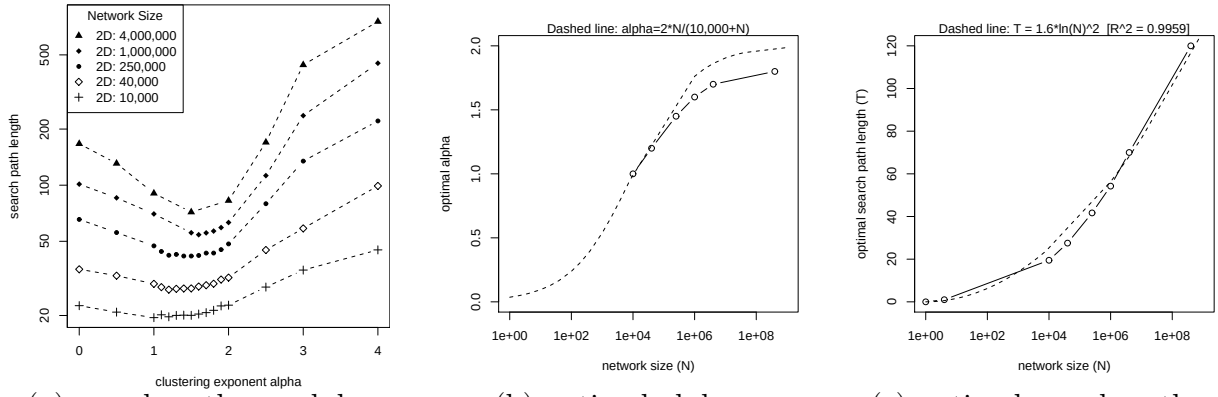
We will use the degree (in-degree) distribution of the ClueWeb09B hyperlink graph and normalize the distribution to fall in a range $[d_{min}, d_{max}]$. With different d_{min} and d_{max} values, the degree distribution will continue to follow a pattern similar to Figure 5.1 but is with a different degree distribution exponent γ because the slope on log-log changes. We plan to use three degree ranges, namely, $[30, 30]$, $[30, 60]$, and $[30, 120]$ to examine

[1] *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.

the impact of degree distribution on decentralized searches. With the range $[30, 30]$, all agents/systems share one common degree, i.e., 30.

5.4.2 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 4.2.3 for details). We will study the impact of $\alpha \in [0, \infty)$ on search performance. As shown in Figures 3.3 and 3.4, when $\alpha = 0$, the network becomes a random network as connectivity 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.



(a) search path vs. alpha (b) optimal alpha (c) optimal search path

Figure 5.6: Results on Search Path Length τ vs. Clustering Exponent α, based on experimental replications of Kleinberg (2000b).

To establish a reasonable range of α for experimentation in the proposed study, we have replicated experimental simulations of Kleinberg (2000b) on various network size scales. As shown in Figures 5.6 (a) and (b), optimal α is smaller than dimensionality of the network model (e.g., < 2 for a 2D space) and potentially converges to the dimensionality when network size becomes extremely large. Further experiments in a distributed IR environment (in a network of thousands of agents) indicated that search

is potentially optimal in the range of $\alpha \in [3, .., 4]$. Hence, we plan to use a slightly wider range $\alpha \in [0, 1, 2, 3, 4, 5]$ plus a very large value (1000) to simulate ∞.

Clustering exponent α offers one simple parameter to control network clustering. This allows simplicity in the analysis of clustering impact on search performance. Nonetheless, in order to understand network dynamics that support searches, we may also conduct analysis on tie strength, i.e., strong ties vs. weak ties, to provide potentially more intuitive insight. Discussions on measuring tie strength on multiple levels can be found in Section 3.4.

5.4.3 Maximum Search Path Length L_{max}

Provided the importance of 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 is allowed to traverse. 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. In our replicated experiments on abstract models, as shown in Figures 5.6 (a) and (c), optimal search path length τ roughly follows $\tau = 1.6 \cdot log_{10}^2(N)$, where N is network size. Treating this as one unit τ_{unit}, we will run experiments on a useful range of $L_{max} \in [\tau_{unit}, 2 \cdot \tau_{unit}, 4 \cdot \tau_{unit}, 8 \cdot \tau_{unit}, 16 \cdot \tau_{unit}]$ in terms of the experimented network size.

5.5 Evaluation: Dependent Variables

IR research in distributed networked environments, with tools from peer-to-peer and multi-agent research, has produced promising results on finding relevant information (Bawa et al., 2003; Crespo and Garcia-Molina, 2005; Zhang and Lesser, 2006; Lu and Callan, 2007). These experiments, however, were typically concentrated on recall. Even when

dealing with large networks, queries used in the experiments were often very broad to have a large relevance base.

Serving diverse users in an open, dynamic environment, implies that some queries are likely to be narrowly defined. The proposed study will focus 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.

5.5.1 Effectiveness: Traditional IR Metrics

We plan to 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:

$$F_1 = \frac{2 \cdot P \cdot R}{P + R} \tag{5.1}$$

In addition, Jarvelin and Kekalainen (2002) proposed several cumulative gain based metrics for IR evaluation. Specifically, given a rank list of retrieval results, the discounted cumulative gain at a rank position p is defined as:

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2 i} \tag{5.2}$$

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:

$$nDCG_p = \frac{DCG_p}{iDCG_p} \tag{5.3}$$

We will primarily use precision, recall, and F_1 for evaluating results from *exact match* searches. For each query, recall is 1 when an exact match is found; recall is 0 if otherwise. Normalized discounted cumulative gain at position 10 ($nDCG_{10}$) will be used in *relevance search* and *authority search* experiments, where a federated rank list of documents gets retrieved for each query.

5.5.2 Effectiveness: Completion Rate

In some search tasks, the goal is not to retrieve relevant documents, but to find relevant peers/systems. We refer to this type of task as *expert finding* or *relevant peer search*, which will be conducted on the TREC Genomics 2004 collection. A search is considered successful when at least one relevant peer is found. *Completion rate* R_c can then by computed by:

$$R_c = \frac{N_{success}}{N_{queries}} \tag{5.4}$$

where $N_{queries}$ is the total number of queries or searches having been conducted and $N_{success}$ the number of successful searches given parametrized limits.

5.5.3 Efficiency

For efficiency, the maximum search path length L_{max} (or the max number of hops allowed) will be controlled in each experiment while the actual search path length will be recorded. The average search length of all tasks can therefore be calculated to

measure efficiency:

$$\bar{L} = \frac{\sum_{i=1}^{N_q} L_i}{N_q} \tag{5.5}$$

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.

Like precision vs. recall, there is tradeoff between effectiveness and efficiency. By definition, precision is 1 when no document is retrieved; recall is 1 when all documents are retrieved. Evaluation is useful only when both metrics are considered. The same applies to effectiveness and efficiency. In the proposed study, our goal is to achieve both high effectiveness and high overall network utility. As discussed in Section 1, methods such as flooding are not desirable even when achieving 100% completion rate because they involve a large number of agents for each search. Effectiveness vs. efficiency plots will be used for comparison.

5.6 Scalability Analysis

One important objective of this research is to learn 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.

Complex network research has found a logarithmic function between search efficiency and network size – that is, under optimal settings, decentralized search time τ is

bounded by $c(\log N)^2$, where N is the network size. Hypothesis 2 states that a similar poly-logarithmic function is possible for IR in networked environments.

In the decentralized IR context, one additional factor in the scalability analysis is *relevance rarity* N_R, defined as:

$$N_R = N/N_{rel} \tag{5.6}$$

where N is the total number of agents (network size) and N_{rel} the number of all agents hosting relevant information to a query. This represents the average size of an agent population for one relevant agent to appear. The larger the rarity N_R, the rarer relevant agents are – so it becomes more challenging to find them. Scalability analysis will also be conducted on search effectiveness/efficiency vs. relevance rarity to identify their functional relationships in optimal searches. In exact match tasks, relevance rarity N_R is identical to network size N given that there is *only one* document relevant to each query ($N_{rel} = 1$).

5.7 Parameter Settings

Table 5.1 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.

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)

Table 5.1: 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 $d \in [30, 60]$.

5.8 Simulation Procedures

Experiments will be conducted on a Linux cluster of 10 PC nodes, each has Dual Intel Xeon e5405 (2.0 Ghz) Quad Core Processors (8 processors), 8 GB fully buffered system memory, and a Fedora 7 installation. The nodes are connected internally through a dedicated 1Gb network switch. The agents (distributed IR systems) will be equally distributed among the 80 processors, each of which loads an agent container in Java, reserves 1GB memory, and communicates to each other. The Java Runtime Environment version for this study is 1.6.0_07. Simulation runs will be mostly automated. We provide the pseudo code on how experiments will be conducted in Algorithm 1.

Algorithm 1 Simulation Experiments

```
1: for each Network Size N ∈ [10^2, 10^3, 10^4, 10^5] do
2:    for each Task Level ∈ [Relevant, Authority, ExactMatch] do
3:       for each Clustering Exponent α ∈ [0, 1, 2, 3, 4, 5] do
4:          rewire the network using α
5:          for each Search Method ∈ [SIM, SimDeg, DEG, RW] do
6:             for each Query do
7:                repeat
8:                   forward a query from one agent/system to another
9:                until relevant found OR search path length L ≥ L_max
10:               if relevant found, i.e., similarity scores surpass a threshold then
11:                  if task is Relevant Search OR Authority Search then
12:                     query additional neighbors for more relevant documents
13:                  else if task is Exact Match Search then
14:                     retrieve the most similar/relevant document
15:                  end if
16:                  send the results back to the first agent/system
17:                  merge and rank all retrieved documents
18:               else
19:                  send message back about failure
20:               end if
21:            end for
22:            measure search effectiveness: precision, recall, F_1, nDCG_10, and/or R_c
23:            measure search efficiency: search path length L and search time τ
24:         end for
25:      end for
26:   end for
27: end for
```

Chapter 6

Experimental Results

In the presentation of experimental results, we focus on *rare known-item (exact match) searches* on the ClueWeb09B collection described in Section 6.1. We report on detailed results in Section 6.2 and analyze the *clustering paradox* in Section 6.3. We evaluate scalability of searches in Section 6.4 and scalability of network clustering in Section 6.5. Section 6.6 presents results on search performances when degree distribution varies. Section 6.7 discusses additional results from *relevance search* (Section 6.7.1), *authority search* (Section 6.7.2), and experiments on the TREC Genomics 2004 collection (Section 6.7.3). We summarize evidence for answers to main hypotheses in Section 6.8.

6.1 Main Experiments on ClueWeb09B

For experiments on the ClueWeb09B collection, 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. These 85 web documents were used as queries in most of our decentralized search experiments[1]. Main experiments were

[1]Only 38 queries were used for the task level of authority searches because the others did not have sufficient incoming hyperlinks for authority evaluation.

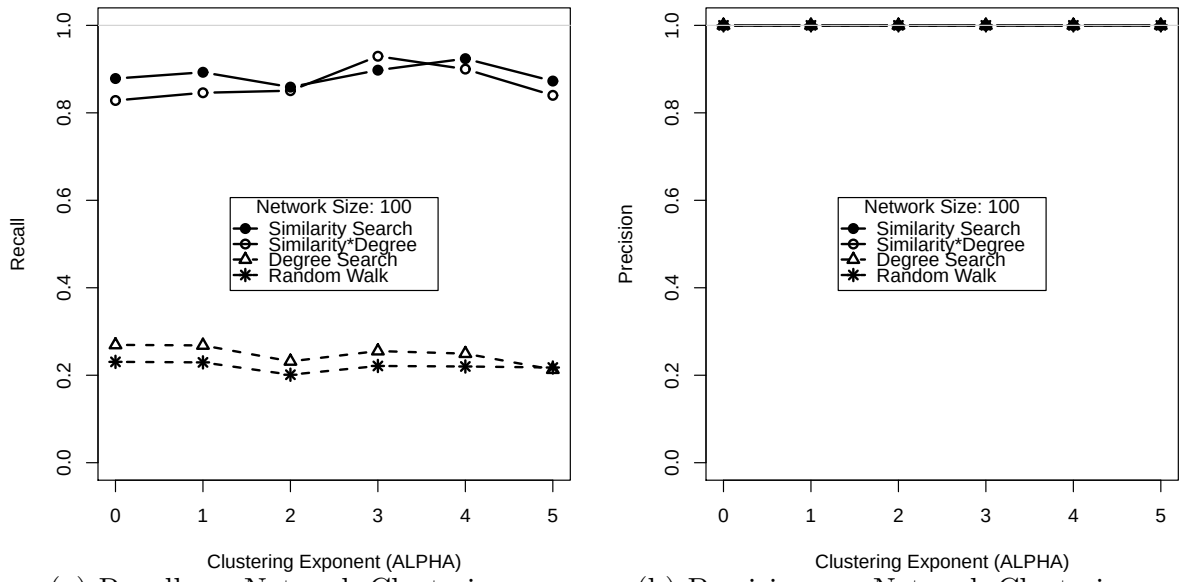
focused on finding exact match documents (rare known items) because this task level, challenging in a distributed environment, can be objectively evaluated.

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 length length L_{max} to $[20, 100, 500, 2500]$ for the network sizes respectively. Table 6.1 shows the number of web documents in each network thus constructed.

Network Size N	100	$1,000$	$10,000$	$100,000$
Number of Documents N_D	0.5 million	1.7 million	4.4 million	10.5 million

Table 6.1: Network Sizes and Total Numbers of Docs

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 $[30, 60]^2$. In all experiments, no document identification information was used for indexing or searching. Sections 6.2.1, 6.2.2, 6.2.3, and 6.2.4 present main experimental results (both effectiveness and efficiency) on the different network size scales.



(a) Recall vs. Network Clustering α (b) Precision vs. Network Clustering α

Figure 6.1: Effectiveness on 100-System Network

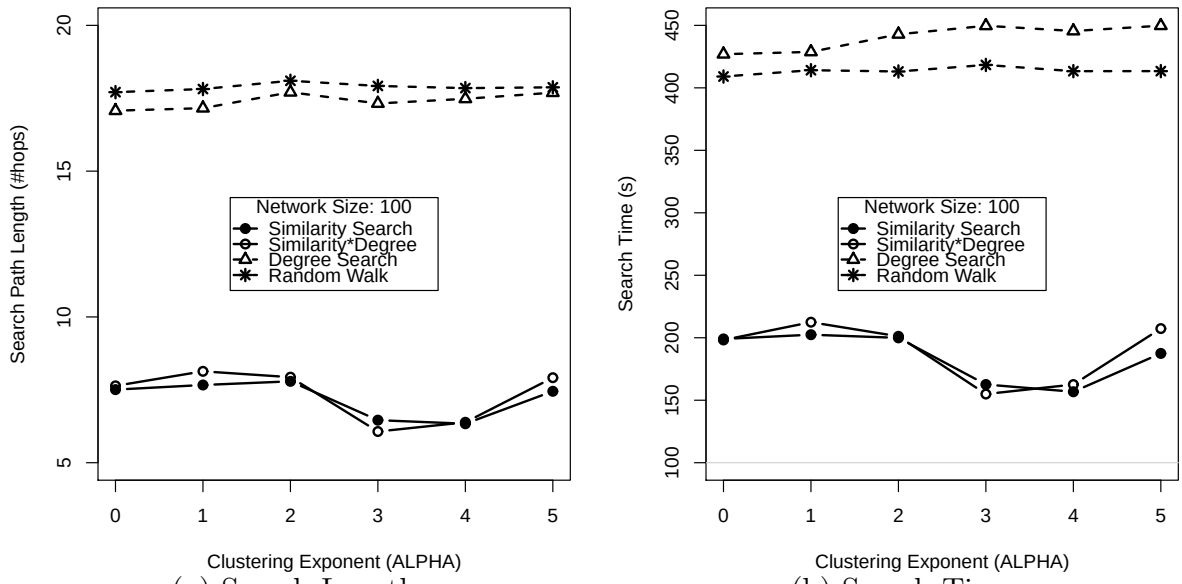
6.2 Rare Known-Item (Exact Match) Search

6.2.1 100-System Network

Figure 6.1 plots search performance in terms of effectiveness (recall and precision) across different network clustering levels $\alpha \in [0, 1, 2, 3, 4, 5]$ on the 100-system network. Overall, similarity search (SIM) and similarity*degree (SimDeg) methods performed very well in terms of effectiveness, showing a very large advantage in recall over degree (DEG) search and random-walk (RW) methods. For example, as shown in Figure 6.1 (a), SIM and SimDeg searches achieved above 0.9 recall at $\alpha = 3$ while DEG and RW searches only had recall values around 0.2. In all searches, precision was maintained at 1.0 because a document was retrieved only when it exactly matched a query (Figure 6.1 (b)).

In terms of efficiency, SIM and SimDeg searches also performed much better than

[2]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) Search Length (b) Search Time

Figure 6.2: Efficiency on 100-System Network

DEG and RW methods on the 100-system network. Figures 6.2 (a) and (b) show efficiency (search path length and search time respectively) vs. network clustering α. Whereas SIM and SimDeg methods only involved 5 systems and took less then 150 milliseconds to reach a recall of 0.9 at $\alpha = 3$, RW and DEG searches traversed $17 - 18$ systems (and more than 400 milliseconds) for a roughly 0.2 recall. The differences are large and statistically significant[3].

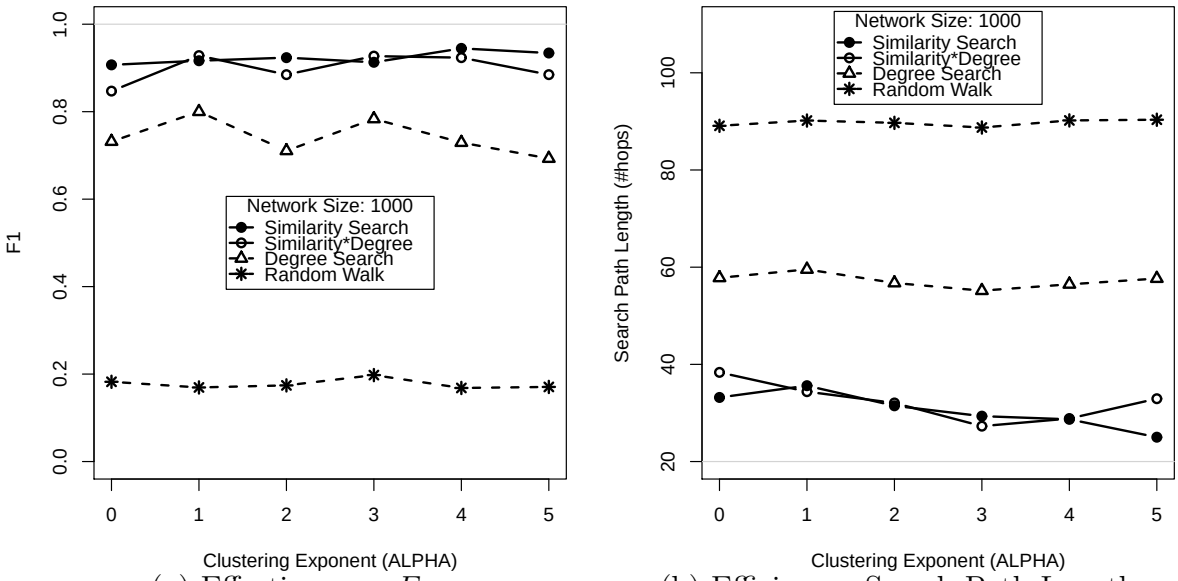
In Figures 6.1 and 6.2, the impact of network clustering (guided by α) on search performance is not clearly shown. As discussed in Section 3.2, among others, network structure is increasingly relevant in larger networks, where it becomes important to find a balance between strong ties for search guidance and weak ties for "jumps." In small networks of 100 systems, a balance of strong ties vs. weak ties is likely less crucial – in a small community, bridges among "remote" segments may not be essentially needed.

In Figure 6.2 (b), results on actual search time look consistent with the search

[3]In discussions that follow, reported differences are statistically significant unless stated otherwise.

path length plot shown in Figure 6.2 (a). Treating query processing of each system as one computational unit, we will use search path length as a surrogate for search time. In the following sections, the presentation on efficiency results will be concentrated on *search path length*. In addition, we will use a single F_1 metric, which combines precision and recall, to simplify discussions on effectiveness results in larger networks $N \in [10^3, 10^4, 10^5]$.

6.2.2 1,000-System Network



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

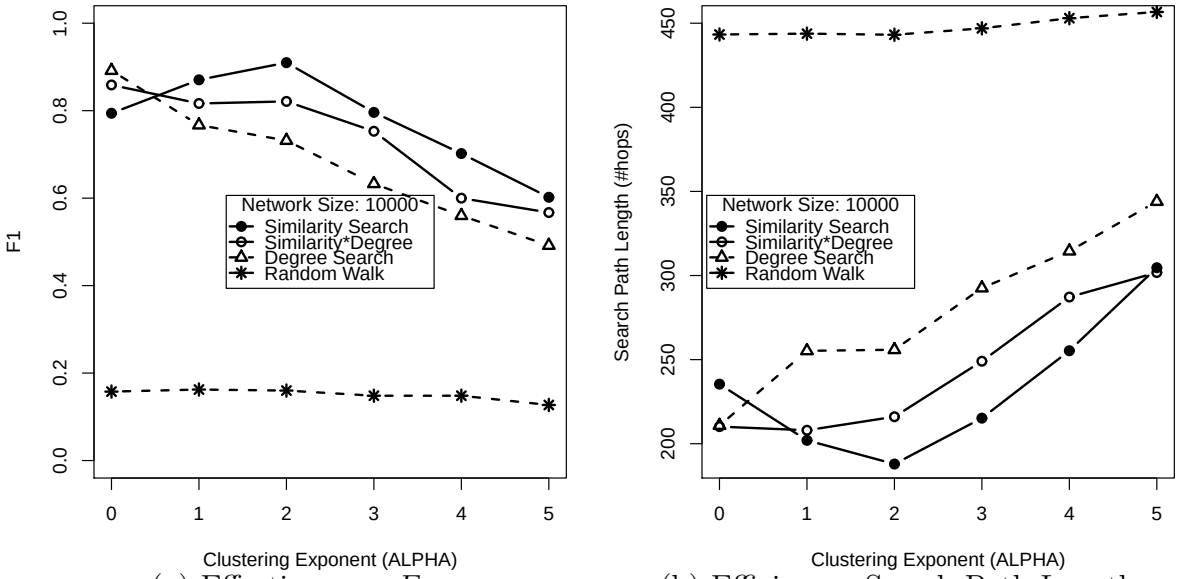
Figure 6.3: Performance on 1,000-System Network

When the network was extended to $1,000$ systems, SIM and SimDeg search methods continued to show large advantages on search performance. As shown in Figure 6.3 (a) and (b), 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. The DEG search appeared to perform much better than RW in the $1,000$-system network. Because queries used in the experiments were about web documents in the 100 most highly connected sites/systems, DEG search,

relying on connectivity information, was able to get out of less connected systems very quickly to reach targets in popular domains.

Based on results from the 1,000-system network, it is still unclear how network structure influenced search performance – visually, there is no obvious pattern of inflection in Figures 6.3 (a) and (b). In the following sections, we will discuss results from experiments on the 10,000- and 100,000-system networks, and present initial evidence, which appears to support the *Clustering Paradox*.

6.2.3 10,000-System Network



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

Figure 6.4: Performance on 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 6.4 (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

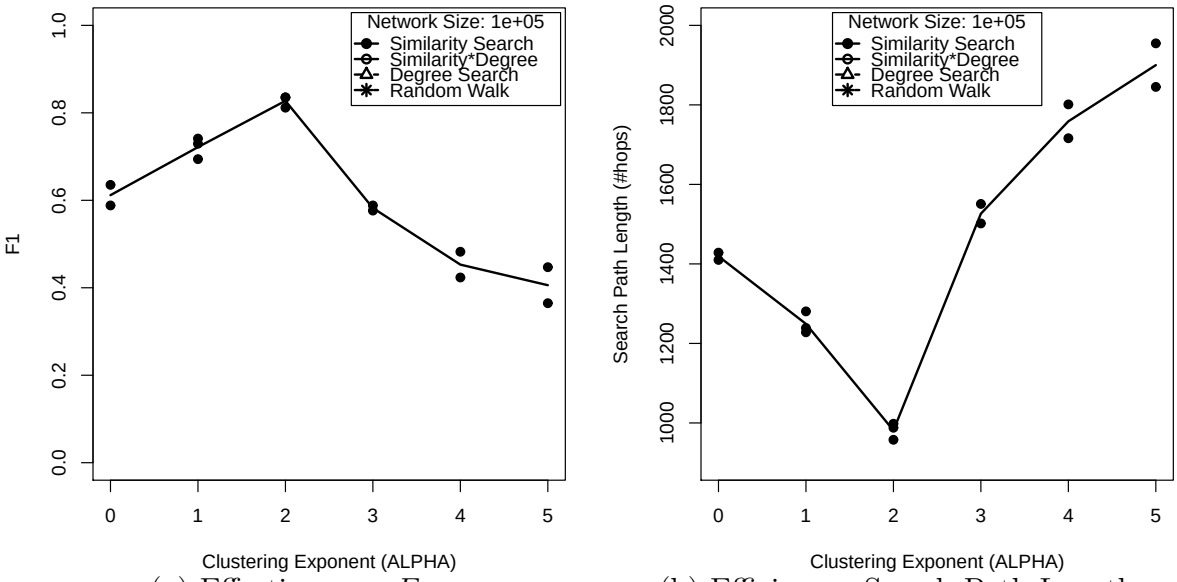
(weaker clustering) or increasing α (stronger clustering) led to degraded performances. The plots provide visual evidence about the *Clustering Paradox* in IR, in which neither under- nor over-clustering is desirable. Section 6.3 presents an in-depth statistical analysis of this phenomenon.

DEG search performances over clustering levels $\alpha \in [0, 1, .., 5]$ follow a very difference pattern. Interestingly, DEG search achieved its best performance at $\alpha = 0$, i.e., with no clustering in a random network. The SimDeg method, which combines similarity and degree information, appears to have mixed the performances of SIM and DEG methods in Figure 6.4 (a) and (b). It remains a question why DEG searches performed very well in random networks without clustering while any level of clustering in the study degraded DEG search performance.

6.2.4 100,000-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 in the largest network proposed, 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. As shown in Figures 6.5 (a) and (b), SIM search achieved its best effectiveness and efficiency also at $\alpha = 2$. Smaller α (weaker clustering) or larger α values (stronger clustering) led to noticeable 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



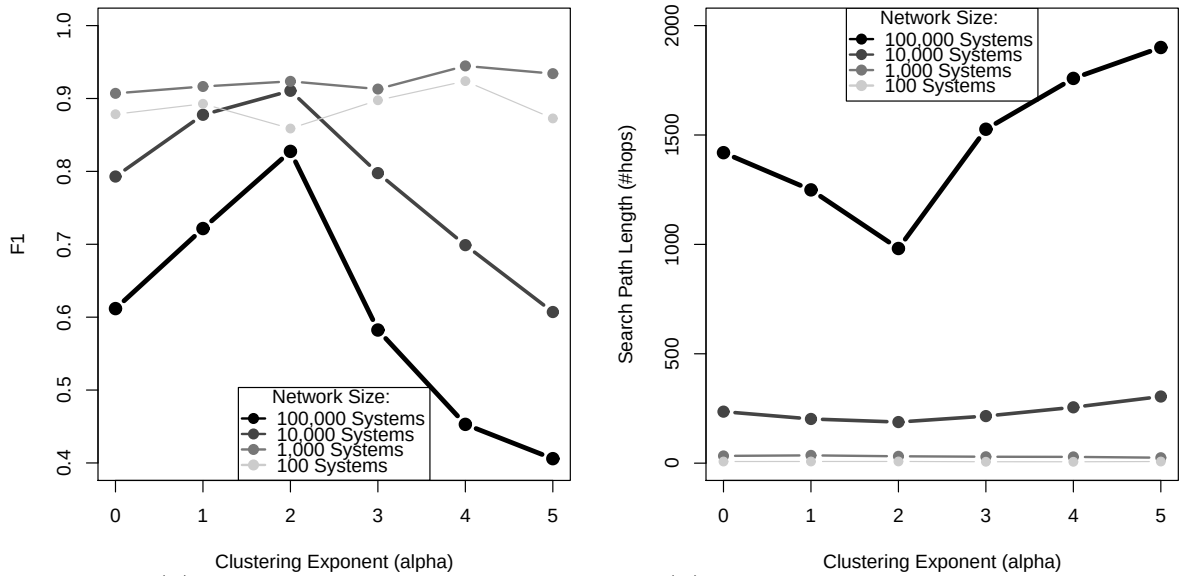
(a) Effectiveness: F_1 (b) Efficiency: Search Path Length

Figure 6.5: Performance on 100,000-System Network. Line is the average of individual data points at each α level.

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 6.3 and 6.4.

6.3 Clustering Paradox

Given that the Similarity Search (SIM) 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

Figure 6.6: Performance on All Network Sizes

Figure 6.6 shows SIM search performances over network clustering levels $\alpha \in [0, 1, 2, 3, 4, 5]$ of networks $N \in [10^2, 10^3, 10^4, 10^5]$ in terms of (a) effectiveness and (b) efficiency. Both sub-figures demonstrate that network structure (clustering) had an important impact on decentralized IR 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 dramatic in the network of $100,000$ systems (Figure 6.6 (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 an increase of search path length roughly from 1000 to 1550, by 550 hops (or 55%).

Statistical tests indicated that SIM search achieved significantly better results with a balanced level of network clustering (i.e., at $\alpha = 2$) than with over- or under-clustering. The significant differences appeared in both the $10,000$-system network and the $100,000$-system network. Results from the tests are shown in Tables 6.2, 6.3 ($10,000$-system network) and Tables 6.4, 6.5 ($100,000$-system network). We elaborate on the results below[4].

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

Table 6.2: SIM Search: Network Clustering on Effectiveness in Network 10,000

Table 6.2 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 $0 \rightarrow 1 \rightarrow 2$, i.e., from random/no clustering to some level of clustering, search effectiveness improved. When α continued to increase from $2 \rightarrow 3 \rightarrow 4 \rightarrow 5$, search effectiveness degraded.

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

Table 6.3: SIM Search: Network Clustering on Efficiency in Network 10,000

[4]Significance codes: *** p < 0.001, ** p < 0.01, * p < 0.05, . p < 0.1.

Similar patterns also appear in Table 6.3 on SIM search efficiency in the 10,000-system network. When α increased from $0 \rightarrow 5$, 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.

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

Table 6.4: SIM Search: Network Clustering on Effectiveness in Network 100,000

Table 6.4 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 $1 \rightarrow 2$ resulted in an F_1 increase of 0.03 and 14 hops shorter in search path length. 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.

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

Table 6.5: SIM Search: Network Clustering on Efficiency in Network 100,000

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. Although increasing α from $4 \rightarrow 5$ in the 10^5 network did not lead to *significant* performance degradation, the *no significance* is likely due to the fact that we only have a couple of data points on each clustering level[5]. The difference is likely significant when more experimental data are obtained.

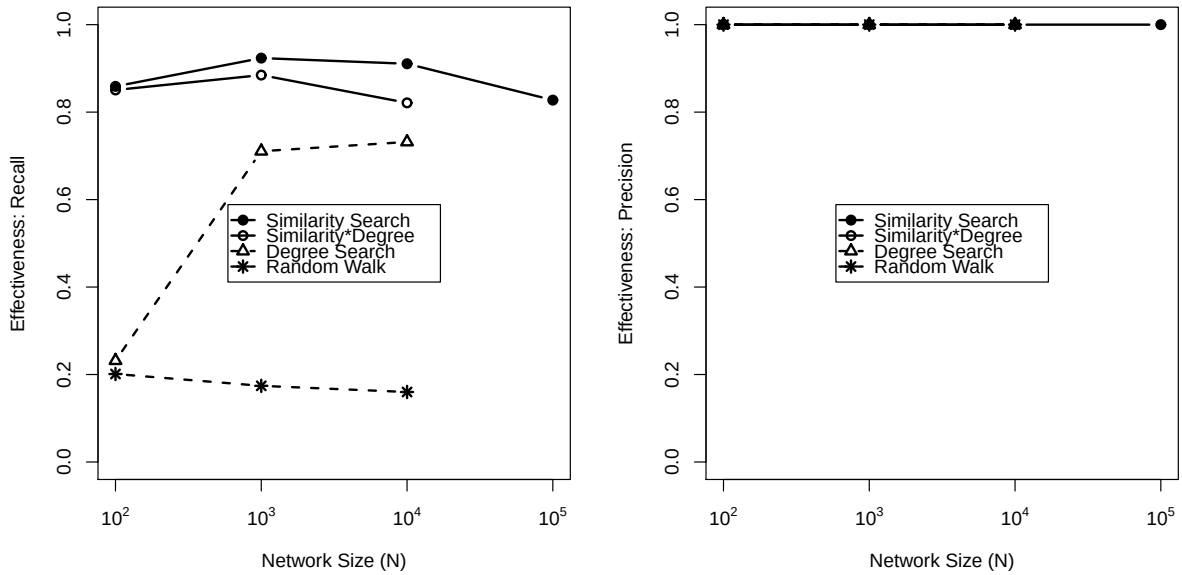
These tests support our first hypothesis about the *Clustering Paradox* – that there does exist a level of network clustering ($\alpha = 2$ in our experiments), below and above which search perform degrades. In other words, that specific level of clustering supports best search performance in terms of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

[5]Because it was time consuming to conduct experiments on the 10^5 network, especially under "bad" clustering conditions, we only had two experimental runs for $\alpha = 4$ and $\alpha = 5$ (each). Each run was conducted on 85 queries.

6.4 Scalability of Search

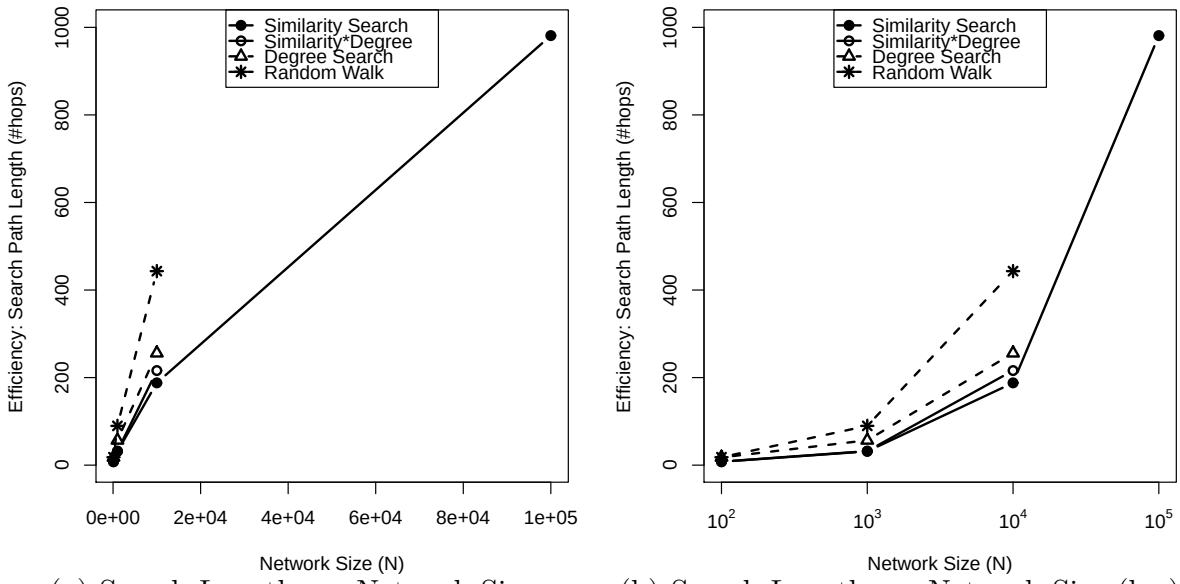
For each network size, we identified network clustering conditions under which superior performance was observed (i.e., at $\alpha = 2$ in the experiments). We plotted recall and precision vs. network size at $\alpha = 2$ in Figure 6.7. As discussed earlier, SIM and SimDeg searches consistently achieved 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) Precison vs. Network Size (log)

Figure 6.7: Scalability of Search Effectiveness at $\alpha = 2$

Figure 6.8 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. To better understand the scalability of SIM search and to predict how it could perform in even larger networks (e.g., a network of millions of nodes/systems), we conducted further analysis on the relationship of search path length to network size.



(a) Search Length vs. Network Size (b) Search Length vs. Network Size (log)

Figure 6.8: Scalability of Search Efficiency at $\alpha = 2$. The only difference between the two figures is that X axis (network size) in figure (b) 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 6.6, in which $L = 0.0125 \cdot log_{10}^{7}(N)$ has a nearly perfect $R^2 = 0.999$.

Search Path Length: $L \sim 0 + \beta \log_{10}^{7}(N)$, 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				

Table 6.6: SIM Search: Search Path length vs. Network size

Figure 6.9 shows actual data points on search path length L vs. network size N, together with values (dotted line) predicted by the regression model $L = 0.0125 \cdot log_{10}^{7} N$.

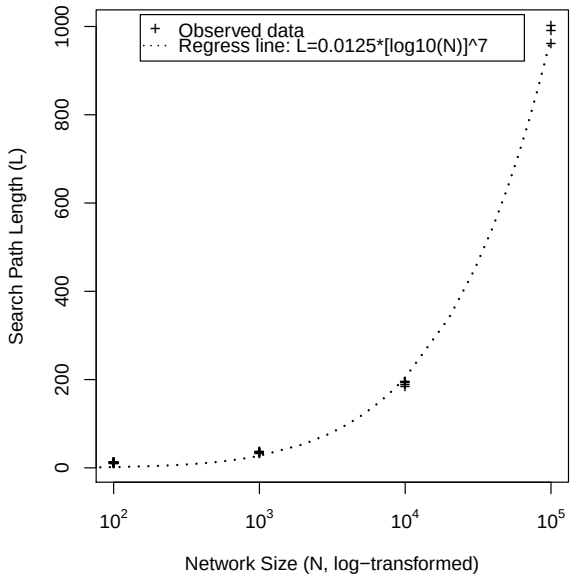


Figure 6.9: Scalability of SIM Search

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.

6.5 Scalability of Network Clustering

We showed that some specific level of network clustering is required for scalable searches. It is also important to understand how much effort is needed to construct and maintain such a network structure for effective and efficient search functions. If network clustering requires intensive computation of individual systems, then it will be challenging for the network community to swiftly evolve and adapt to dynamic changes over time.

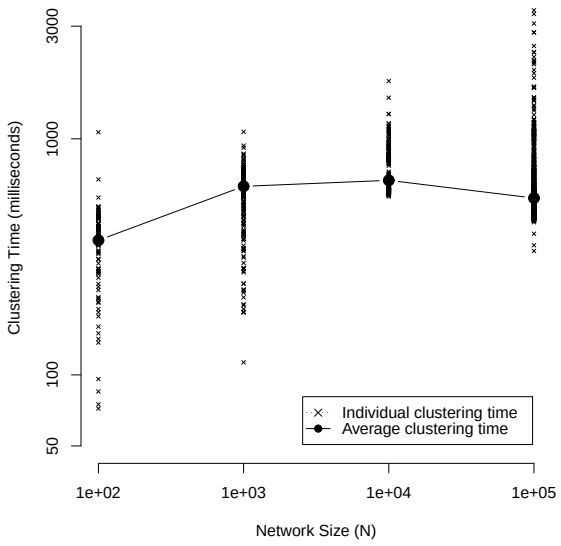
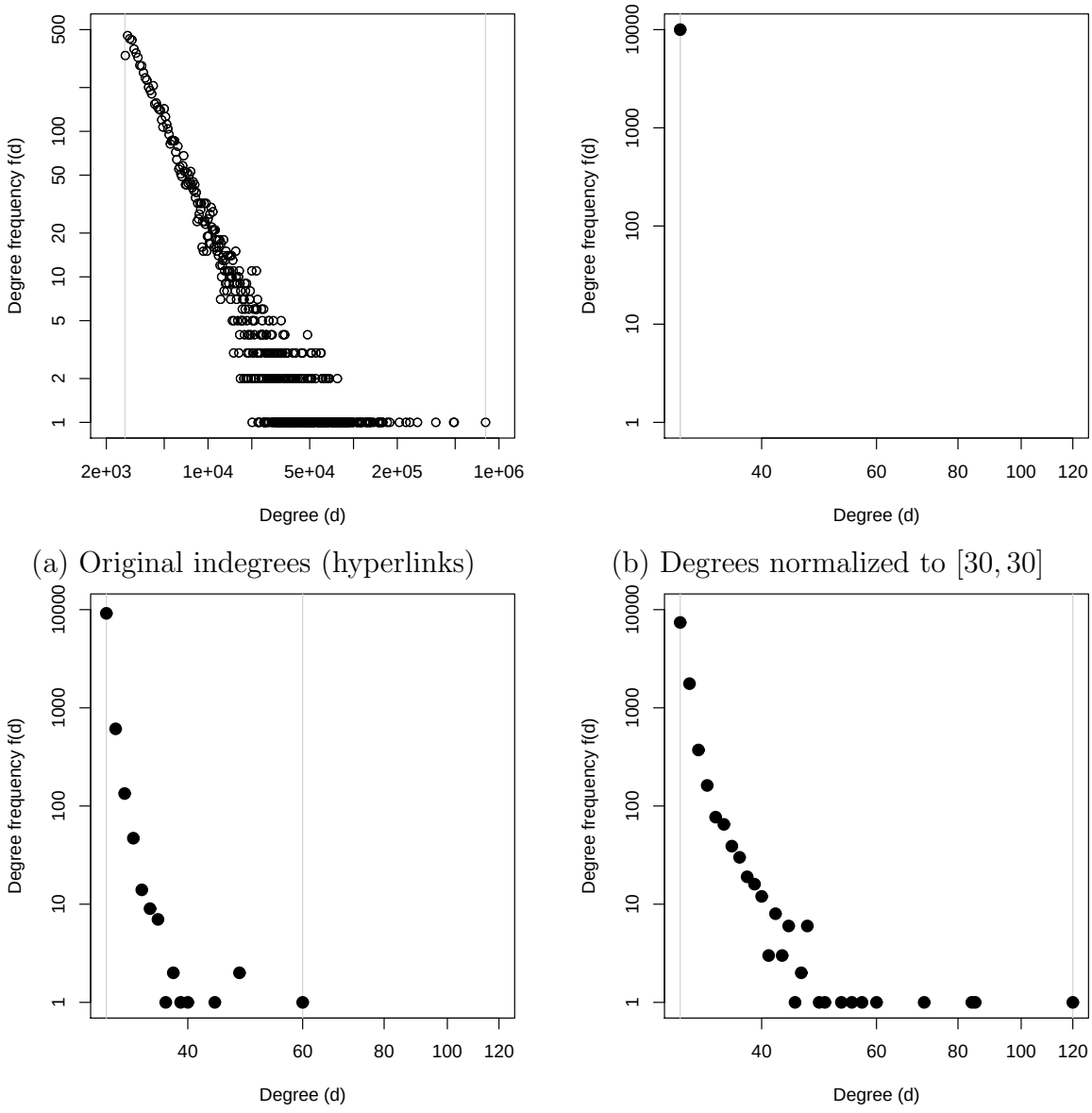


Figure 6.10: Scalability of Network Clustering

Our search methods relied on local indexes and a network structure self-organized by distributed systems in the network. Without global information and centralized control, network clustering was performed locally – distributed systems formed the network structure in terms of their limited opportunities to interact and individual preferences/constraints on building indexes for others.

This local mechanism for clustering demonstrated a high level of scalability. As shown in Figure 6.10, average clustering time τ_c remained relatively constant, < 1 sec, across all network size scales $N \in [10^2, 10^3, 10^4, 10^5]$. When there are changes in the network (e.g., 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. As shown by experimental results, this local mechanism supports very effective and efficient discovery of relevant information in the global space.

6.6 Impact of Degree Distribution



(a) Original indegrees (hyperlinks)

(b) Degrees normalized to $[30, 30]$

(c) Degrees normalized to $[30, 60]$

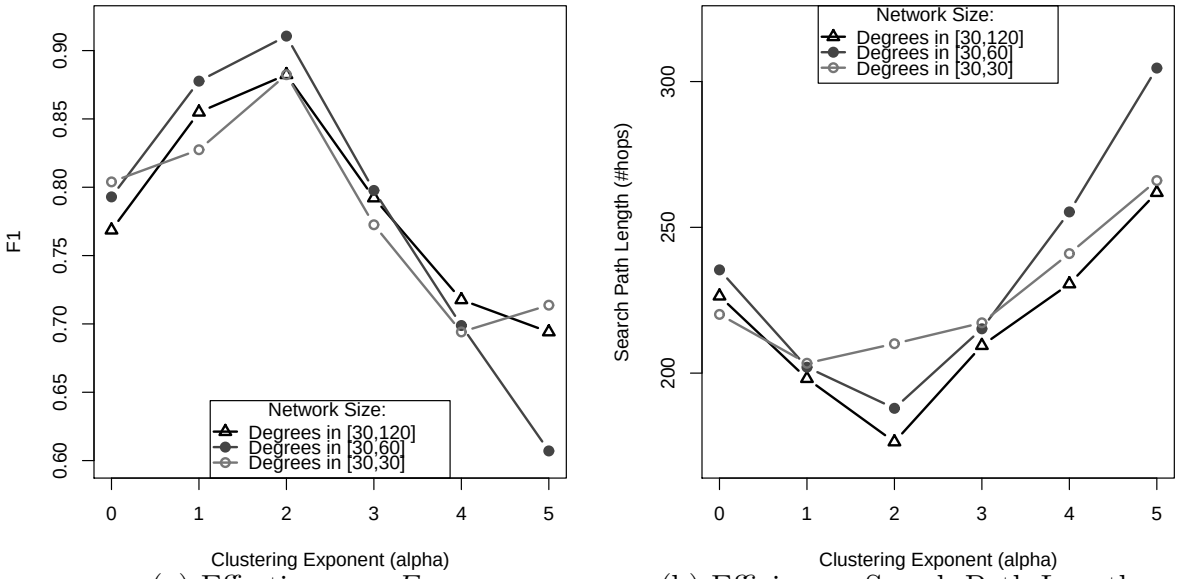
(d) Degrees normalized to $[30, 120]$

Figure 6.11: Degree Distribution and Normalization of $10,000$ Systems

The main experiments discussed in earlier sections were conducted on a degree (d_u, number of 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 shown in Figure 6.11 (a). We normalized all degrees to fit in the range of $[30, 60]$ using Equation 4.5 described in Section 4.2.3, resulting in the distribution shown in Figure 6.11 (c). These degrees were then used in experiments for network construction and clustering.

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, as shown in Figure 6.11 (b). Figure 6.11 (d) shows the degree distribution normalized to $[30, 120]$, in which degrees spread over larger values as compared to those $\in [30, 60]$ (Figure 6.11 (c)).



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

Figure 6.12: SIM Search Performance with Varied Degree Ranges

Experimental results with different degree ranges $[30, 30]$ and $[30, 120]$, in addition to main experiments on range $[30, 60]$, are shown in Figures 6.12 (a) and (b). While results mostly look consistent, those on range $[30, 30]$ look somewhat confounding. In Figure 6.12 (a), best effectiveness of SIM search with $d_u \in [30, 30]$ appeared at $\alpha = 2$. In Figure 6.12 (b), however, $\alpha = 2$ did not seem to produce best efficiency for that

degree range (search path length at $\alpha = 1$ looks shorter/better).

In order to better interpret the plots, we adopted a single measure that combined both effectiveness (F_1) and efficiency (search path length) for easier comparison. We refer to the new score as F_1 per 200 Hops, which is computed by: $F_{L200} = 200F_1/L$, where L is search path length. The combined score can be seen as a normalized effectiveness score given a fix time limit. Figure 6.13 shows search performances in terms of F_{L200} scores.

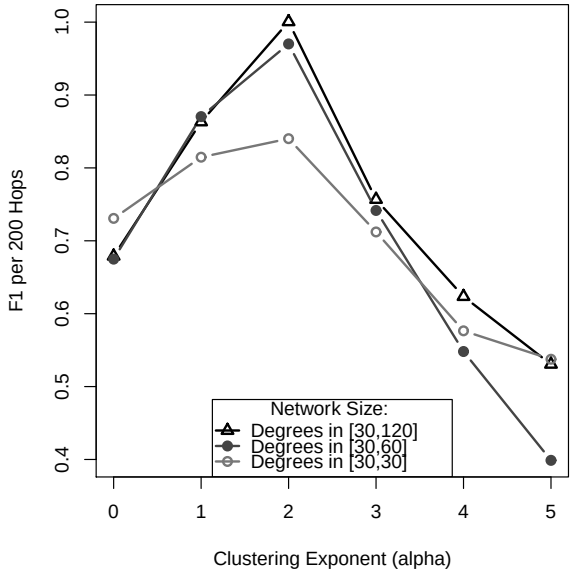


Figure 6.13: SIM Search Performance F_{L200} with Varied Degree Ranges

Comparison	Difference in F_1 per 200 Hops	Error	t value	$Pr(>\|t\|)$		R^2
$\alpha : 0 \rightarrow 1$	0.1842	0.01515	12.16	0.00026	***	0.974
$\alpha : 1 \rightarrow 2$	0.1374	0.02351	5.844	0.028	*	0.945
$\alpha : 2 \rightarrow 3$	-0.2438	0.03198	-7.621	0.017	*	0.967
$\alpha : 3 \rightarrow 4$	-0.1332	0.02422	-5.501	0.0053	**	0.883
$\alpha : 4 \rightarrow 5$	-0.09295	0.02839	-3.274	0.031	*	0.728

Table 6.7: SIM Search: Network Clustering on F_{L200} with $d_u \in [30, 120]$

As shown in Figure 6.13, best performances on the three different degree distributions $[30, 30]$, $[30, 60]$, and $[30, 120]$ all appeared at $\alpha = 2$. We tested performance difference (in terms of F_{L200}) of every two consecutive alpha levels. Table 6.7 shows test

Comparison	Difference in F_1 per 200 Hops	Error	t value	$Pr(>\|t\|)$		R^2
$\alpha : 0 \rightarrow 1$	0.08398	0.02618	3.207	0.033	*	0.72
$\alpha : 1 \rightarrow 2$	0.02542	0.02809	0.905	0.42		0.17
$\alpha : 2 \rightarrow 3$	-0.128	0.03078	-4.158	0.014	*	0.812
$\alpha : 3 \rightarrow 4$	-0.1357	0.02774	-4.891	0.0081	**	0.857
$\alpha : 4 \rightarrow 5$	-0.03898	0.02234	-1.745	0.16		0.432

Table 6.8: SIM Search: Network Clustering on F_{L200} with $d_u \in [30, 30]$

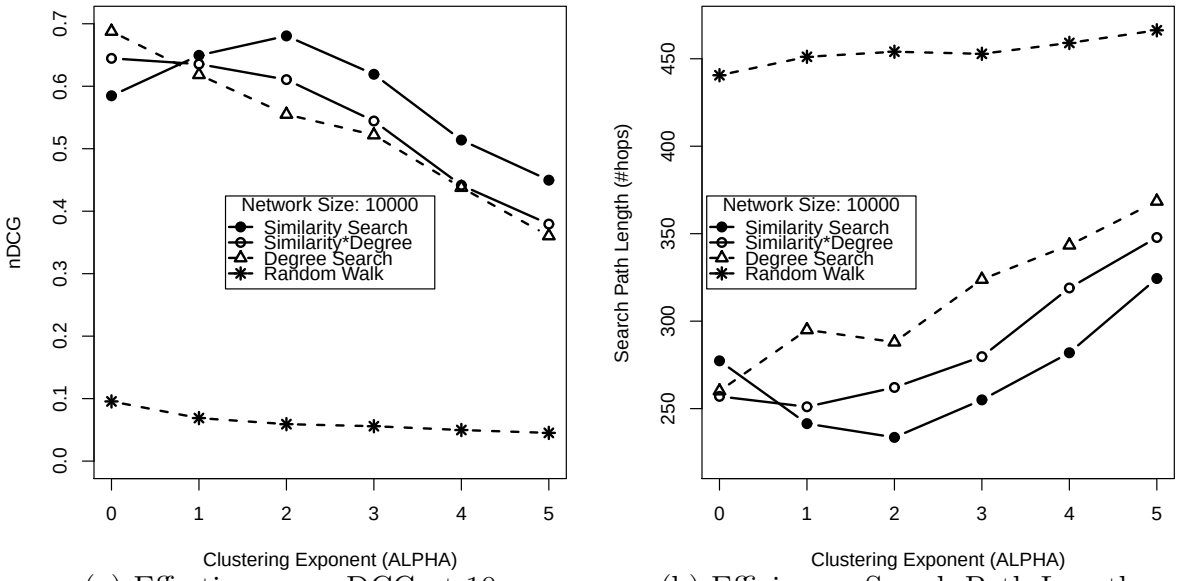
results for degree range $[30, 120]$, supporting the observation that optimized network clustering level for degrees $\in [30, 120]$ was at $\alpha = 2$.

Tests on degree range $[30, 30]$, as shown in Table 6.8, produced consistent results. Whereas the general trend looks similar to that of $[30, 120]$, results showed no significant difference between $\alpha = 1$ and 2. Hence, the inflection point is likely between 1 and 2. Overall, while search performance changes when degree distribution varies, evidence continues to support the existence of the *Clustering Paradox*.

6.7 Additional Experiments and Results

6.7.1 Relevance Search on ClueWeb09B

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 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, 2007).



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

Figure 6.14: 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 5.5).

Figure 6.14 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 with random/no clustering, i.e., at $\alpha = 0$, and degraded when there was stronger clustering.

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

Table 6.9: SIM Search: Network Clustering on Relevance Search Effectiveness

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

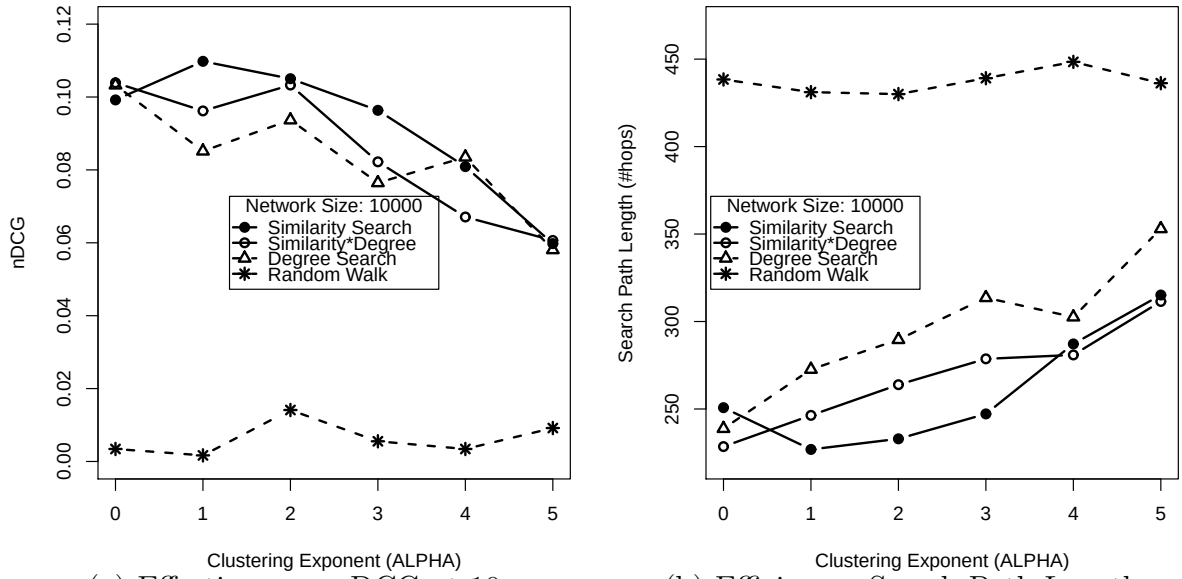
Table 6.10: SIM Search: Network Clustering on Relevance Search Efficiency

We analyzed SIM search performances over different values of $\alpha \in [0, 1, 2, 3, 4, 5]$. Table 6.9 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 $0 \rightarrow 1 \rightarrow 2$, i.e., from random/no clustering to some level of clustering, search effectiveness improved. When α continued to increase from $2 \rightarrow 3 \rightarrow 4 \rightarrow 5$, search effectiveness degraded. This trend resembles how F_1 changed over α values in exact match searches (compare to Table 6.2).

Similar patterns also appear in Table 6.10 on SIM search efficiency in the 10,000-system network. When α increased from $0 \rightarrow 5$, 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 6.3). This provides further evidence that the *Clustering Paradox* also existed in relevance searches (hypothesis 1).

6.7.2 Authority Search on ClueWeb09B

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*.



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

Figure 6.15: Authority Search Performance on 10,000-System Network

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 6.15 presents results from authority search experiments on the 10,000-system network. As shown in Figure 6.15 (a), search effectiveness was low in general – SIM, 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 6.15 (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. Visually, larger or smaller α values than 1 degraded both effectiveness and efficiency.

Comparison	Difference in $nDCG_{10}$	Error	t value	$Pr(>\|t\|)$		R^2
$\alpha: 0 \rightarrow 1$	0.01059	0.004723	2.242	0.055	.	0.386
$\alpha: 1 \rightarrow 2$	-0.004764	0.004747	-1.004	0.34		0.112
$\alpha: 2 \rightarrow 3$	-0.008683	0.002584	-3.361	0.0099	**	0.585
$\alpha: 3 \rightarrow 4$	-0.01544	0.00385	-4.011	0.0039	**	0.668
$\alpha: 4 \rightarrow 5$	-0.02114	0.004253	-4.97	0.0011	**	0.755

Table 6.11: SIM Search: Network Clustering on Authority Search Effectiveness

Comparison	Difference in Search Length	Error	t value	$Pr(>\|t\|)$		R^2
$\alpha: 0 \rightarrow 1$	-23.88	5.866	-4.071	0.0036	**	0.674
$\alpha: 1 \rightarrow 2$	6.063	5.423	1.118	0.3		0.135
$\alpha: 2 \rightarrow 3$	14.25	4.201	3.392	0.0095	**	0.59
$\alpha: 3 \rightarrow 4$	39.99	4.324	9.25	0.000015	***	0.914
$\alpha: 4 \rightarrow 5$	27.96	5.131	5.449	0.00061	***	0.788

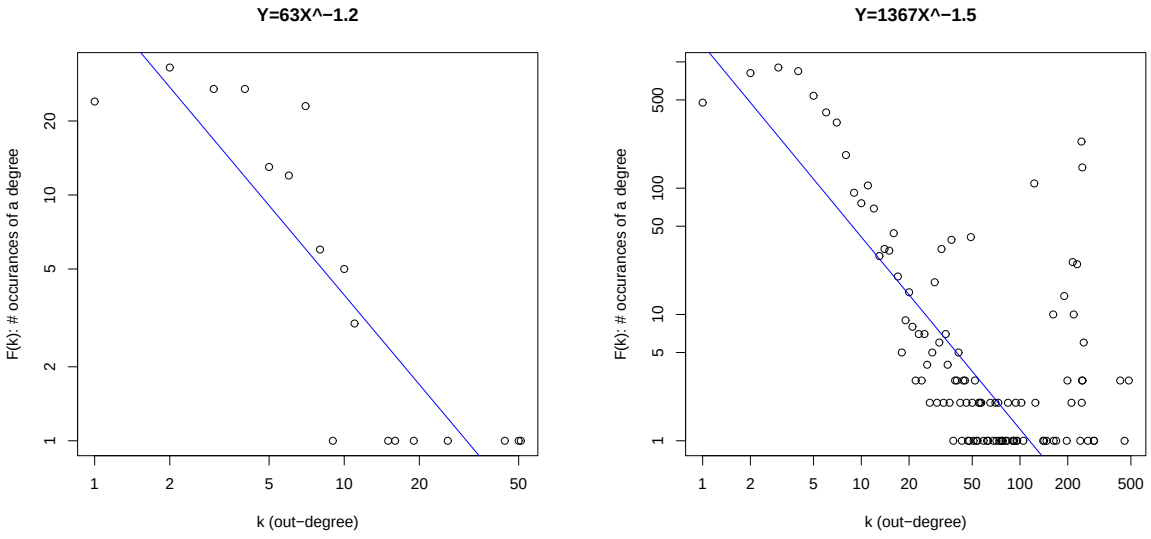
Table 6.12: SIM Search: Network Clustering on Authority Search Efficiency

To understand the performance inflection in authority searches, we tested SIM search performance difference between any two consecutive clustering levels of $\alpha \in [0, 1, 2, 3, 4, 5]$. Tables 6.11 and 6.12 show the test results on effectiveness and efficiency respectively. 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.

6.7.3 Experiments on TREC Genomics

Data Collection and Networks

We conducted relevant peer (expert) searches on the TREC Genomics 2004 collection. The task was to find an expert peer given a topic in a network of peers (representatives of scholars having document collections). To establish initial peer networks, we first chose six scholars in the medical informatics domain, i.e., associate editors of the Journal of the American Medical Informatics Association (JAMIA). We then identified their direct co-authors (1^{st} degree) who published 10 to 80 articles in the TREC collection, resulting in a small network of 181 peers. The network was later extended to the 2^{nd} degree (i.e., co-authors' co-authors) to total 5890 peers for experiments on a larger scale.



(a) 181-Peer Network (b) 5890-Peer Network

Figure 6.16: Genomics 2004 Data: Degree Distributions

Both networks had a diameter (the longest of all shortest pairwise paths) of 8. Degree distributions of the networks are shown in Figure 6.16 (a) and (b). For each peer, which represented a scholar, all articles (with titles and abstracts) authored or

co-authored by the scholar were loaded as the local information collection.

Relevant Peer Search

On the TREC Genomics 2004 collection, we constructed peer-to-peer networks by treating each unique scholar as a peer, who possessed a local collection of documents published by the scholar (author). The task involved finding a peer with relevant information in the network, given a query. Applications of this framework include, but are not limited to, distributed IR, P2P resource discovery, expert location in work settings, and reviewer finding in scholarly networks. However, we focused on the general decentralized search problem in large networked environments.

Relevant peers/agents were considered few, if not rare, given a particular information need. For experiments on the TREC Genomics 2004 collection, we considered those scholars whose topical similarity to a given query was ranked above the fifth percentile. Hence, for evaluation purposes, peers were sampled to estimate a threshold similarity score for each query, which was then used in experiments to judge whether a relevant peer had been found.

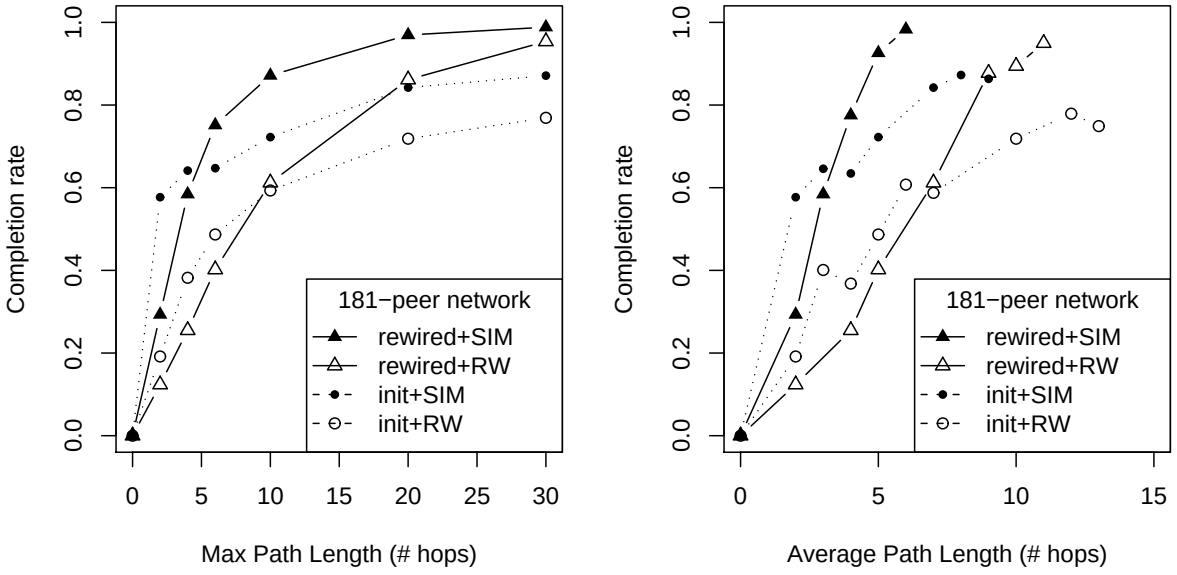
We retrieved citations to articles published in the Journal of the American Medical Informatics Association (JAMIA) in the Genomics track collection and used all (498) articles with titles and abstracts to simulate queries/submissions. For each submission, an agent that represented the editor in chief of JAMIA assigned it to one of the associate editors, who then began to forward the submission to a potential relevant agent/scholar through connected neighbors (e.g., co-authors).

Experimental Results

From experiments on the TREC Genomics 2004 data, we present effectiveness and efficiency results on initial and rewired networks of 181 and 5890 peers and focus on

the impact of network clustering on decentralized search performance.

Results on 181-Peer Network



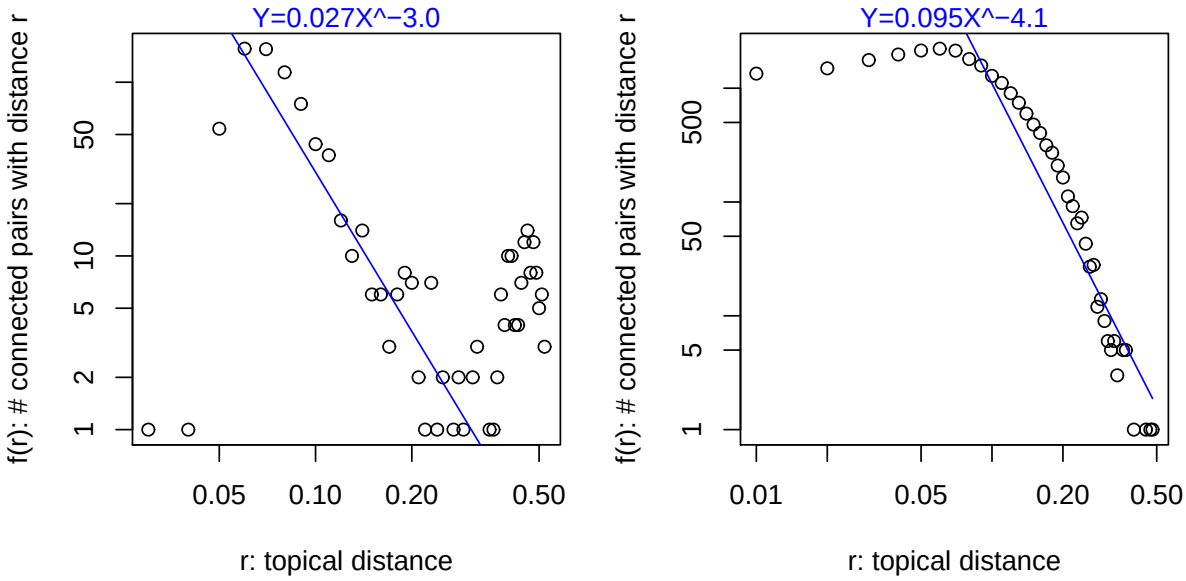
(a) Completion rate vs. Max Search Length (b) Completion rate vs. Average Search Length

Figure 6.17: Effectiveness vs. Efficiency on 181-Agent Network

Figure 6.17 shows experimental results on 181-peers networks. The X axis denotes the efficiency (search path length) while Y is effectiveness (completion rate). Solid points refer to the SIM search method. Dotted lines are results based on the initial co-authorship network. With the initial network (dotted lines), similarity-based SIM search consistently outperformed random walks (RW), especially within small search path lengths. For instance, within two hops, SIM search already achieved a completion rate of more than 50% while random-walk was still at 20%. Allowing for longer search path lengths helped both models but neither reached a completion rate higher than 90%, suggesting that there were particular characteristics of the initial network that disoriented some searches after a long path.

Clustering analysis, as plotted in Figure 6.18 (a) on log/log coordinates, showed

that the association between connectivity frequency and topical distance has a power-law region (in the middle) with irregularities. We believe that SIM search was well guided by the network in most instances (when routed through peers with regular clustering-guided connections) but was lost in others (disoriented in regions where irregular connections dominated).

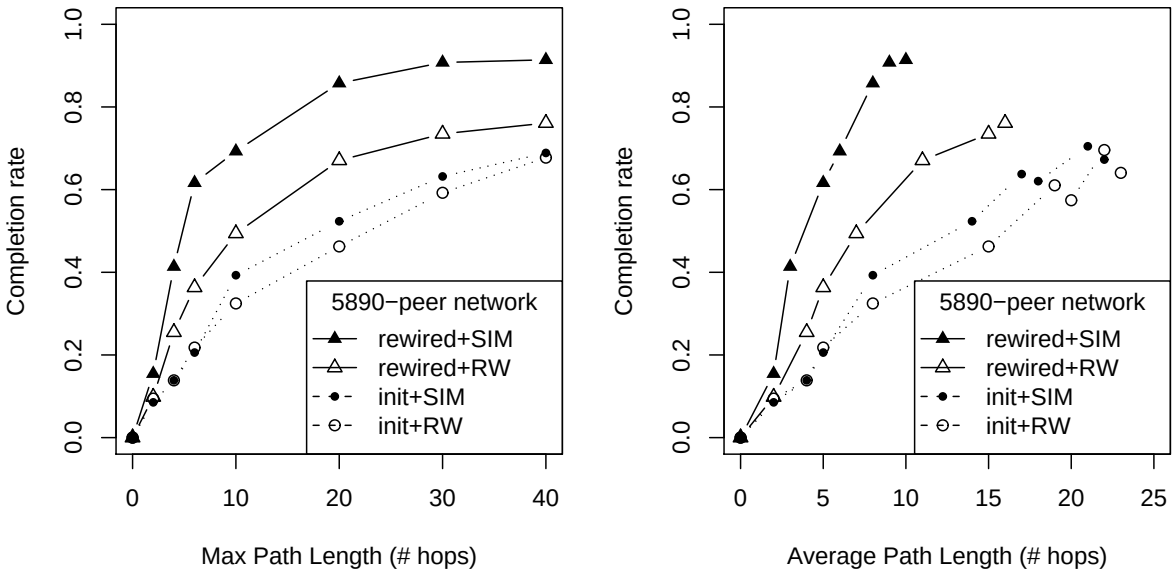


(a) 181-Agent Network

(b) 5890-Agent Network

Figure 6.18: Clustering of Initial Genomics Networks: Connectivity frequency (Y) vs. Topical distance (X). Compare to Figure 3.3.

To demonstrate potential utility of network clustering, we rewired the network (network clustering) based on the connectivity probability function described in Section 4.2.3. Experimental results with clustering exponent $\alpha = 3.0$ are shown as solid lines in Figure 6.17, in which proper network clustering better guided SIM search and further improved the results – a higher than 95% completion rate was already achieved at max search path length 20 (Figure 6.17 (a)) or average path length 5 (Figure 6.17 (b)).



(a) Completion rate vs. Max search length

(b) Completion rate vs. Average search length

Figure 6.19: Effectiveness vs. Efficiency on 5890-Agent Network

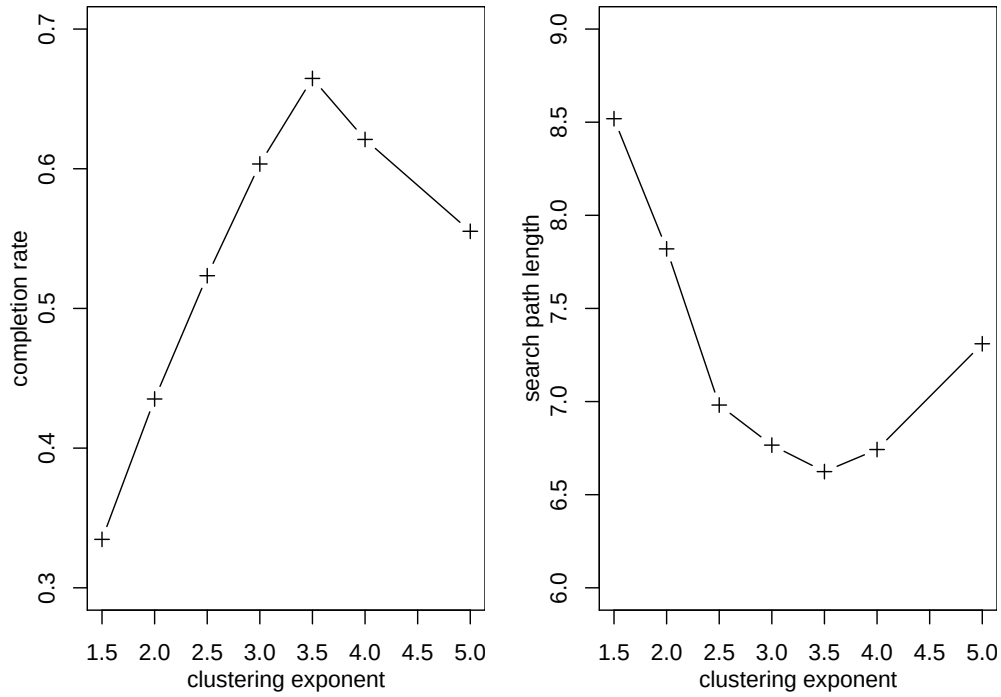
Results on 5890-Peer Network

On the initial 5890-peer network, experimental results indicated that SIM search had very limited advantage over *random walk*, as shown by dotted lines in Figures 6.19 (a) and (b). Further analysis revealed that the network was very weakly clustered. As shown in Figure 6.18 (b) on log/log coordinates, the correlation between connectivity and topical distance departed quite a bit from a power-law function (linear on log/log). There were many topically remote connections. Peers had many weak ties for a query to "jump" but insufficient strong ties to circulate the query within the boundary of a relevant neighborhood.

Again, we performed network clustering described in Section 4.2.3 to reconstruct/rewire the 5890-peer network. As shown by solid lines in Figure 6.19, given *clustering exponent* $\alpha = 4.0$, the *SIM* search method performed much better and achieved above 90% completion rate within a max search path length of 40 (Figure 6.19 (a)), or an average search path length of about 10 (Figure 6.19 (b)).

Impact of Clustering

In the results above, we have demonstrated that some level of network clustering improved decentralized search for relevant peers. It is unclear yet how much clustering is enough or how much is too much. Setting max search path length at 10, experiments on SIM search with various clustering exponent α values on the 5890-peer network produced results shown in Figures 6.20 (a) and (b).



(a) Effectiveness (b) Efficiency

Figure 6.20: Impact of Clustering Exponent α (X)

Figure 6.20 shows that the SIM search method achieved best performance, i.e., highest completion rate in (a) and shortest search path length in (b), at $\alpha \approx 3.5$. Both smaller and larger α values resulted in less optimal searches. As discussed, smaller α values produced less visible topical segments and more remote connections that disoriented searches. Larger α values, on the other hand, led to an over-clustered and fragmented network without sufficient *weak ties* for searches to move fast.

This result, obtained in a decentralized relevant peer searching context, is consistent with findings from *relevance search*, *authority search*, and *exact match* experiments on the ClueWeb09B collection. It continues to support hypothesis 1 regarding the *Clustering Paradox*, in which some balance between *strong ties* and *weak ties* should be maintained for effective and efficient searches.

6.8 Summary of Results

Experimental results have shown that relevant information can be found quickly not only in small networks (e.g., a network of 100 distributed systems) but also in networks of a larger scale (e.g., networks of 100,000 systems). Experiments in various settings have produced consistent results well aligned with the theory. We summarize major findings below in terms of hypotheses stated in Section 3.5.

6.8.1 Hypothesis 1: Clustering Paradox

> H1: There exists some level of network clustering, below and above which search performance degrades.

Yes, there was the *Clustering Paradox*. Best similarity (SIM) search performance was supported by $\alpha = 2$ in most experiments[6]. Stronger or weaker clustering degraded search performance. The *clustering paradox* appeared in all three levels of search tasks, namely, *exact match* (rare known item search), *relevance search*, and *authority search*. Additional results from experiments on the TREC Genomics 2004 collection were consistent to this finding.

6.8.2 Hypothesis 2: Scalability of Findability

> H2: With optimal network clustering, search time (search path length) is explained by a poly-logarithmic function of network size.

Yes, there was evidence on scalable searches. Search path length L of SIM search is poly-logarithmic to network size N at $\alpha = 2$: $L = 0.0125 \cdot log_{10}^{7}(N)$ in exact match experiments. The model was tested on data containing five network size levels $N \in$

[6]In authority searches, the inflection point was projected to be an α value between 1 and 2.

$[1, 10^2, 10^3, 10^4, 10^5]$ (more than 5 experimental runs on each network size) and produced an ideal fit $R^2 = 0.999$.

6.8.3 Hypothesis 3: Impact of Degree Distribution

H3: Hypotheses 1 and 2 remain true with different degree distributions.

Yes, we observed the clustering paradox in various degree distribution settings. In the $10,000$-system network, for example, the balanced level of network clustering for best search performance was at $\alpha = 2$ given degree range $[30, 60]$. With a varied distribution $\in [30, 90]$ or $\in [30, 120]$, an inflection point remained even though it appeared at a slightly different clustering level (H1 supported). The poly-logarithmic scalability function was established on degree distribution $\in [30, 60]$. H2 in the other degree settings requires further investigation. In future work, we plan to use a much wider range of degrees, which is more likely to resemble power-law characteristics in real networks but will require more computing power to simulate highly connected systems.

6.8.4 Hypothesis 4: Scalable Search Methods

H4: Search methods that utilize information about neighbors' degrees and relevance (similarity to a query) are among scalable algorithms stated in Hypotheses 1 and 2.

Yes, relevance (similarity) information was particularly useful to guide searches. The similarity search (SIM) method, among the four strategies proposed, consistently achieved best results. As discussed earlier, given $\alpha = 2$, search path length of SIM search is a poly-logarithmic relation to network size. Degree information was also helpful, especially because queries used in experiments were about web documents from highly popular web domains. DEG search, which utilized degree information, and SimDeg

search, which combined similarity and degree information, performed competitively in large networks.

Chapter 7

Conclusion

With the rapid growth of digital information, it becomes increasingly challenging for people to survive and navigate in its magnitude. It is crucial to study basic principles that support adaptive and scalable retrieval functions in large networked environments such as the Web, where information is distributed among dynamic systems. In this research, we aimed to address the scalability challenge facing classic information retrieval models and researched on a decentralized, organic view of information systems pertaining to search in large scale networks. The study 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.

7.1 Clustering Paradox

We conducted experiments on decentralized IR operations on various scales of information networks and analyzed effectiveness, efficiency, and scalability of proposed search methods. Results provided evidence about the *Clustering Paradox* in the IR context and showed network structure was crucial for retrieval performance. In an increasingly large, distributed environment, decentralized searches for relevant information

were able to function well only when systems interconnected in certain ways. Relying on partial indexes of distributed systems, some level of network clustering under local topical guidance supported very efficient and effective discovery of relevant information in large scale networks.

In main 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 allowed a balance between strong ties and weak ties. While strong ties aids in creating local segments useful to guide searches, weak ties provide opportunties 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, and exact match (rare known-item search). Additional experiments on another benchmark IR collection, namely, TREC Genomics 2004, supported this major finding.

7.2 Scalability of Findability

Examining the *Clustering Paradox* is crucial to understanding how search functions 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 $log^7(N)$, where N is the 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, larger than 2 discovered by Kleinberg (2000b) in search experiments on simplified network models. We mainly focused on network clustering for search performance in the experiments and believe that a smaller exponent can be expected when other variables in decentralized searches are taken into account.

7.3 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 connect given their opportunities and preferences. This organic mechanism potentially allows for a bottom-up approach to coping with dynamics in a fast growing information network.

Chapter 8

Implications and Limitations

In an open, dynamic information space such as the Web, people, information, and technologies are all mobile and changing entities. The classic view of "knowing" where information is and indexing "known" collections of information for later retrieval is hardly valid in these environments. Finding where relevant repositories are for the *live* retrieval of information is critical. Without global information, new methods have to rely on local intelligence of distributed systems and/or their delegates to collectively construct paths to desired information.

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

With the magnitude of information and the number of computing systems on the Internet, any level of centralization will be doomed to great challenges and potential failure. We believe that the fully decentralized view expressed in this study reflects a reality we cannot avoid in information retrieval research. While monolithic search

systems continue to struggle with scalability problems of today, the future of search likely requires a better infrastructure where all can participate.

With a focus on the impact of network structure on search performance, this dissertation has produced promising results on finding relevant information in large scale distributed environments. Findings, nonetheless, should be interpreted with caution because experiments were conducted under certain assumptions/conditions. The current research is limited in several aspects. We discuss future research directions in light of current limitations.

Network Dynamics and System Adaptation

In a dynamic networked information space, all can change and evolve. While users may have different information needs, contents of distributed systems in the network may appear, disappear, and change over time. Information that is relevant, valid and findable now may not be so in the future.

Network clustering requires systems/agents to connect to one another in terms of their similarities/preferences. In a dynamic environment, agents need to interact with others and understand changing settings. Learning provides an important means for agents to perceive their environment and act accordingly, critical to overall system utility and robustness.

In this research, we assumed that contents in distributed systems were relatively static and a network structure only needed to be built once to reflect the content distribution. Future studies will investigate how a network structure (clustering) can be dynamically maintained when systems/agents come and go with evolving information collections. We also plan to study the dynamics of search traffics and how an entire network can cope with individual system failures. Agent learning and adaptation will be a key focus in this research direction.

User and Relevance

This study relies on automatic simulations based on text queries and pre-established relevance judgment for the evaluation of distributed IR systems. It is well known in IR that the notion of *relevance* involves multiple dimensions beyond topicality. Relevance often depends on user's search contexts and can rarely be judged objectively using a pre-established relevance base. In the future, we hope to develop a user interface for the decentralized system and involve real users in the study of searching and evaluation. There might be new interface elements to be studied as well given that searches will be conducted in a different manner. Because many individual systems participate in the decision making for search, we expect such a system to provide more diversified results than those from classic, centralized models. The TREC Web track (the diversity task in particular) might be a good platform for result comparison in this regard.

Representation, Ranking, and Result Fusion

In this dissertation research, we limited retrieval algorithms to a set of classic methods, such as TF*IDF for information representation, Cosine similarity for relevance scoring, and a simple normalization function for result fusion. The underlying assumption was that every individual information collection (system), large or small, could be represented using a meta-document based on document frequency values. This assumption, however, is hardly valid for very large collections containing a diverse set of topics. For example, *en.wikipedia.org* contains information about nearly every major subject in the world. A single meta-document will not be able to represent such a big and diverse collection accurately. How to determine the granularity for large collection representation is an important question. Future work will also study other retrieval models such BM25 ranking and CORI result fusion in distributed network environments.

In this study, we used text contents of documents (e.g., web pages) to simulate

queries. On the Web, however, users usually issue very short queries, e.g., queries with only a couple of terms. In addition, existing experiments provided evidence that it was easier to find relevant information for some queries than for others. Query representation factors such as query length and model (e.g., binary vs. weighted) are worth further investigation.

Potential Barriers to Implementation

Although experiments show promises, much remains to be done before our model can be implemented to work in a real world environment. One additional important assumption in our experimental model was that systems/agents were cooperative and trustworthy. Decentralization in the reality, however, allows for individual participants to do independent decision making and exercise self interests. System behaviors, driven by their own objectives, may become very different from what is ideally expected.

Why would systems participate in decentralized search and contribute their computing power? There have to be benefits and/or incentives that motivate individuals to do so. Ideas can be borrowed from peer-to-peer applications, where individual computer systems share their resources in order to gain access to other resources. Besides incentives, we are yet to study why (and how to make sure) systems would behave in a contributive manner. There have been plenty of examples about free-riders in peer-to-peer networks, who take advantage of existing resources but have very little willingness to contribute. Others may offer contributions only to mislead and boost their own popularity.

Mechanisms have to be built to ensure better behaviors. Methods must also be implemented to detect harmful practices and guide beneficial interactions. Trust plays an important role in all this. Implementation of a decentralized search infrastructure will have to take into account issues of trust among uncooperative, untrustworthy, or

malicious systems by drawing on findings and inspirations in distributed trust management.

Finally, there is one crucial question concerning how much effort is needed for individuals and/or organizations to implement connections to a network when it is ready for participation. Just as the power of the Web relies on its growing population, the power of a decentralized search network is dependent on how well the technology can be adopted quickly. Only with a good magnitude of information and computing power can such a network be useful to people and continue to attract additional resources. To achieve this, the cost of establishing connections should be close to the level of adding hyperlinks to web pages, connecting to a peer-to-peer network, or simply joining an online social network.

Appendix A

Glossary

- **network** or **graph**: a data structure of a set of entities called **nodes** or **vertices**, which connect to each other through a set of pairwise **edges** (undirected) or **arcs** (directed), e.g., a network of web pages (nodes) connecting to each other through hyperlinks (arcs).

- **degree**: the number of *edges* or *arcs* a *node* has, e.g., the number of unqiue co-authors a scholar has in a co-authorship network.

- **peer-to-peer** (P2P) system: a distributed system consisting of interconnected nodes able to self-organize into network topologies with the purpose of sharing resources such as content, CPU cycles, storage and bandwidth, capable of adapting to failures and accommodating transient populations of nodes while maintaining acceptable connectivity and performance, without requiring the intermediation or support of global centralized server or authority.

- **peer**: an entity, often an independent information system or computer, in a *peer-to-peer* network, whose *edges* represent communication/interactions with other peers.

- **agent**: a computer system situated in some environment, and that is capable of *autonomous action* in this environment in order to meet its design objectives.

- **multi-agent** systems: a societal view of multiple agents in certain environment

with an emphasis on the collective capability, as oppose to the individual agent as the functional unit.

- **neighbor**: from a network or graph perspective, a *node* that the *current node* directly connects to, e.g., a web page directly linked from the current page, a peer that communicates with the current peer in a peer-to-peer network, or an agent that interacts with the current agent in multi-agent systems.

The multi-agent paradigm is often used to model peer-to-peer systems, in which the concepts *agent* and *peer* are equivalent.

Appendix B

Research Frameworks in Literature

	PROBLEMS				
FRAMEWORKS	Findability	Scalability	Robustness	Relevance	Recall
Complex Network					
Boguna2009	•	•			
Hu2009	•	•			
Simsek2008	•	•		•	
Kurumida2006	•	•			
Liben-Nowell2005	•	•			
Adamic2005	•				
Dodds2003	•				
Watts2002	•	•			
Kleinberg1999/2000/2006	•	•			
Watts1998	•	•			
Milgram1967/1969	•				
Peer-to-Peer IR					
Doulkeridis2008	•	•		•	•
Raftopoulou2008	•	•		•	•
Skobeltsyn2007	•	•		•	•
Lu2003/2004/2006/2007		•		•	•
Wang2006	•	•			

Amoretti2006	•	•			
Luu2006	•	•	•	•	•
Cooper2005		•			•
Bender2005	•	•		•	
Zeinalipour-Yazti2004	•	•	•		•
Tsoumakos2003	•	•	•		
Bawa2003		•	•	•	•
Li2003	•	•			
Lv2002	•	•	•	•	•
Adamic2001			•		•
Multi-Agent System IR					
Zhang2004/2006/2007	•	•	•	•	•
Ke2007				•	
Kim2006		•			
ZhangJ2005/2006	•			•	•
Mukhopadhyay2005	•		•	•	
Fu2005	•			•	
Yu2002/2003	•	•	•	•	•
Pereira2002	•				
Singh2001	•		•	•	
Menczer1998	•	•	•		•
Foner1997	•				
Distributed (Federated) Information Retrieval					
		•[?]		•	•
Link-based Ranking Methods					

				•	•
Collaborative Filtering					
				•	•

Table B.1: Research Problems and Frameworks

Appendix C

Research Results in Literature

Model	Data	N	N_{rel}	$\langle k \rangle$	$\langle l \rangle$	C	D	N_R	τ	Reference	Spatial	Degree
Abstract Models												
2D Lattice	Synthetic	4×10^8	1	5	n/a	0	2	4×10^8	120	Kleinberg2000	unif	unif
2D Lattice	Synthetic	4×10^6	1	5	n/a	0	2	4×10^6	70	Ke2009	unif	unif
		1×10^6	1					1×10^6	54			
		2.5×10^5	1					3×10^5	42			
		4×10^4	1					4×10^4	28			
		1×10^4	1					1×10^4	19			
3D Lattice	Synthetic	1×10^6	1	7	n/a	0	3	1×10^6	33	Ke2009	unif	unif
Hierarchical	Synthetic	$102,400$	1	99	6		1-13	$102,400$	7	Watts2002	n/a	unif
		$204,800$	1	99	6		1-13	$204,800$	7			
		$409,600$	1	99	6		1-13	$409,600$	7			

		1×10^8	1	99	6		1-13	1×10^8	7			
Hidden Space	Synthetic	1×10^5	1				2	1×10^5	55	Boguna2009	unif	power
1D Circle	Synthetic	1×10^3	10	n/a	3	n/a	1	100	12	Simsek2008	unif	power
1D Circle	Synthetic	1×10^3	10	n/a	3	n/a	1	100	22	Simsek2008	unif	Poiss
IR Experiments												
Geograph	Airports	500*	1	15	n/a	n/a	2	100	6	Boguna2009	geo	n/a
TFIDF+Cos	Citation	833	10*	16	n/a	n/a	n/a	83	15	Simsek2008	n/a	power
RefNet VSM	Coauthor	5,891	295	20	8	n/a	n/a	20	10	Ke2009b	n/a	power
RefNet VSM	Coauthor	181	9	20	8	n/a	n/a	20	5	Ke2009b	n/a	power
Hierarch SON	.GOV2sub	5,000	200*					250	28	Doulkeridis08		
Gradt+Rand	TREC-6	2,000	20	12		0.69		100	6	Raftopoulou08		
Hierarchical	.GOV2	25,000	200	3	5	0		125	10	Lu2007		
Hierarchical	TREC WT10g	2,500	50	3	5	0		50	3	Lu2007		
Agent view	TREC 123	100							n/a	Zhang 2007		
Agent view	TREC VLC	921							n/a	Zhang 2007		
PursuitLearn	Reuters	37	1	36	1	1	1	37	8	Ke 2007		unif
Hierarchical	TREC WT10g	2,500	50	3	5	0		50	4	Lu2006		

MINERVA	.GOV	20	4					5	1	Bender2005		
Org Hierarch	HP email	490	1	13	3.1			490	5	Adamic2005	n/a	power
BreakConnect		200	12*						n/a	Cooper2005		
Hamming dist	Enron	147	≥ 1	10*	2.5	0.096	n/a	74	10	ZhangJ2005		
ISM	Reuters	104	2*	8	≤ 4	n/a		52	5	Zeinalipour-		
										Yazti2004		
Agent View	TREC VLC	912	73					13	5	Zhang2004	n/a	
SETS seg.	CiteSeer	83,946	500*					168*	8	Bawa2003		
Hierarchical	TREC WT10g	11,485								Lu2003		
MARS Ref	Coauthor	4,933	287	n/a	n/a	n/a	n/a	17	10	Yu2003	n/a	n/a
Best degree	Gnutella								n/a	Adamic 2001		

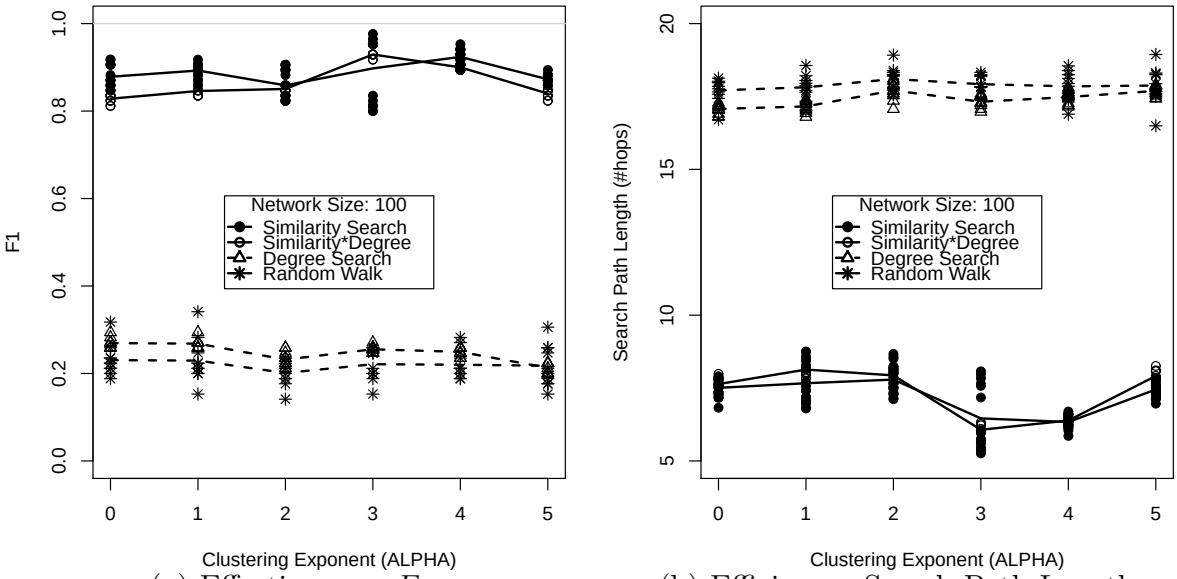
Table C.1: Research Results on Findability and Scalability. Symbols: 1) N: the number of nodes in the network; 2) N_{rel}: the number of relevant nodes (search targets) in the network; 3) $\langle k \rangle$: average number of connections or neighbors a node has; 4)$\langle l \rangle$: average path length between any two nodes in the network; 5) C: clustering coefficient, or the probability of one's neighbors directly connect to each other; 6) D: dimensionality of the model; 7) N_R: rarity, i.e., one target out of N_R peers on average; 8) τ: traversal time, or the number of hops traveled to find a target. '*' denotes estimates, no such data reported in paper.

Appendix D

Experimental Data Detail Plots

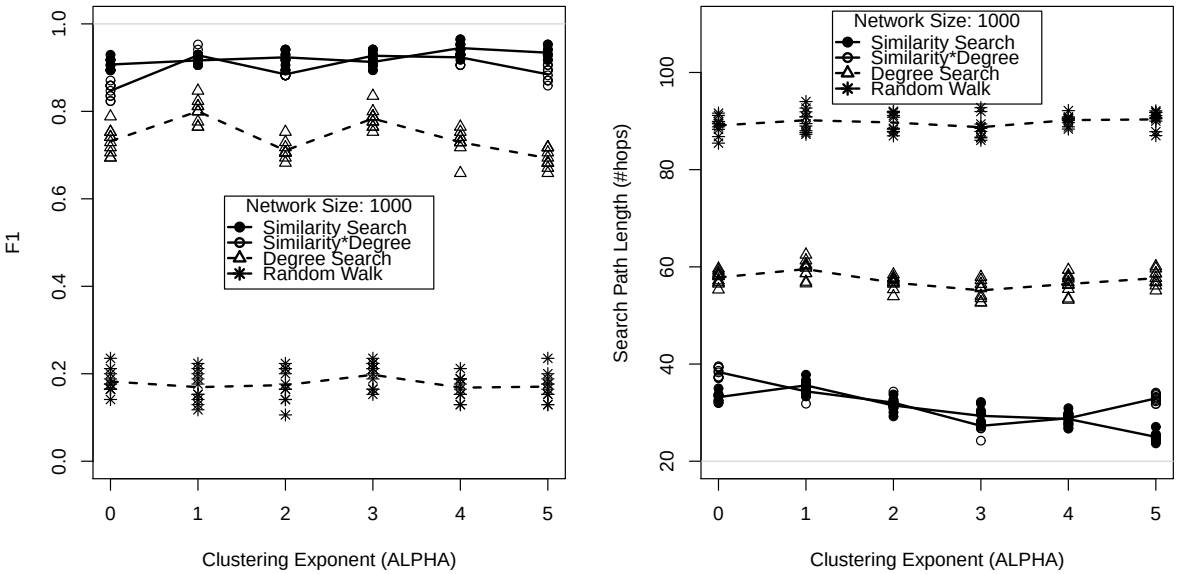
In Chapter 6 Experimental Results, plots are mainly based on aggregated data, e.g., average search path lengths and effectiveness scores of multiple experimental runs. Here we plot data from individual experiments to show how they vary at each X (α) level.

D.1 Exact Match Searches



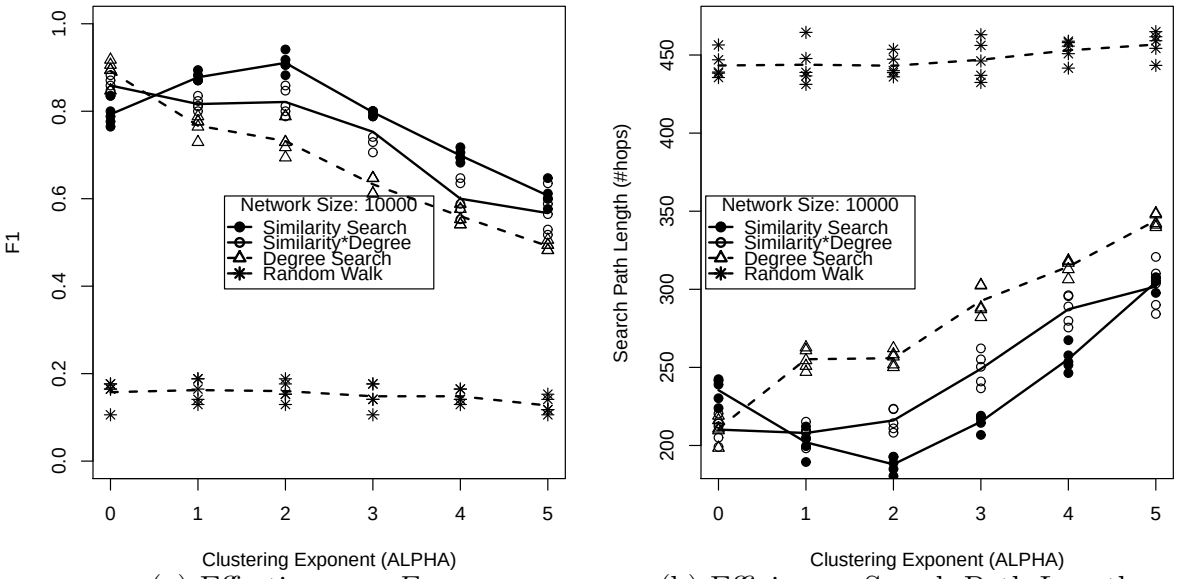
(a) Effectiveness: F_1 (b) Efficiency: Search Path Length

Figure D.1: Performance on 100-System Network



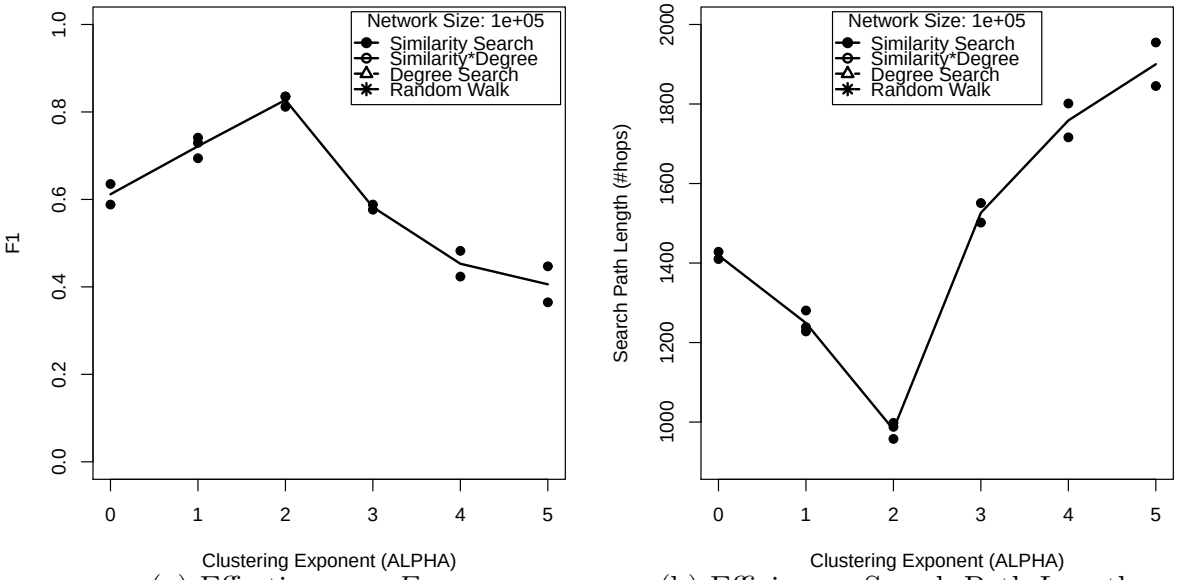
(a) Effectiveness: F_1 (b) Efficiency: Search Path Length

Figure D.2: Performance on 1,000-System Network



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

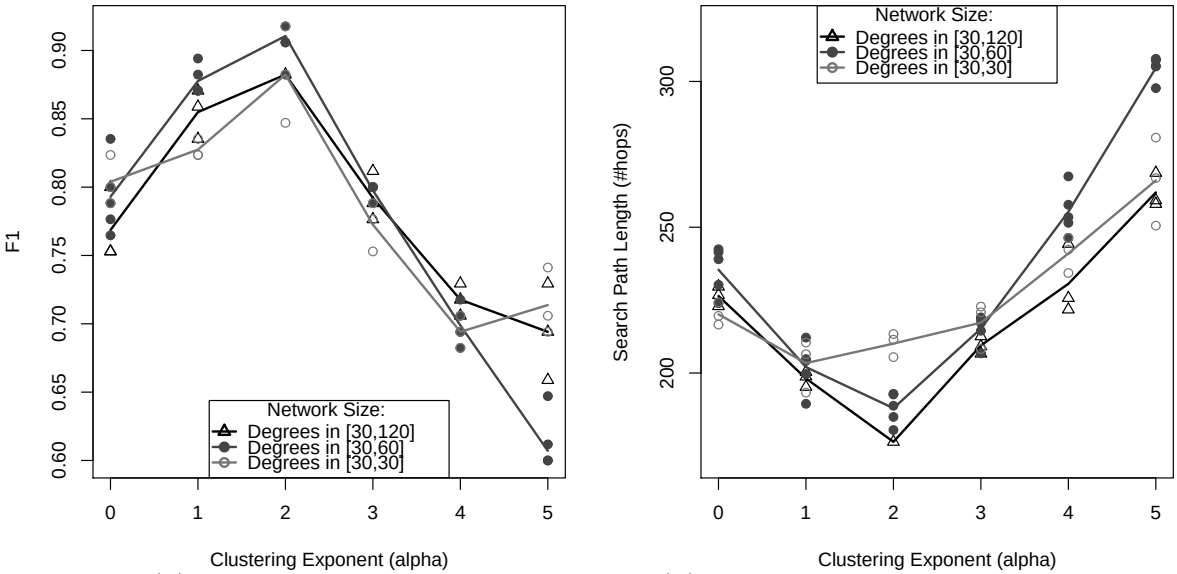
Figure D.3: Performance on 10,000-System Network



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

Figure D.4: Performance on 100,000-System Network

D.2 Impact of Degree Distribution



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

Figure D.5: SIM Search Performance with Varied Degree Ranges

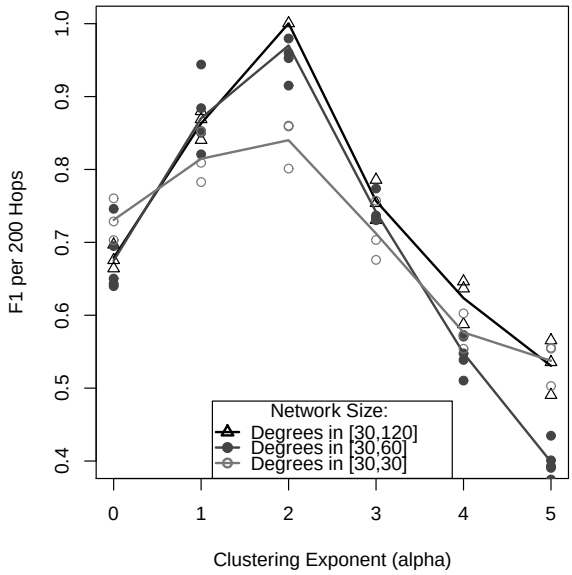
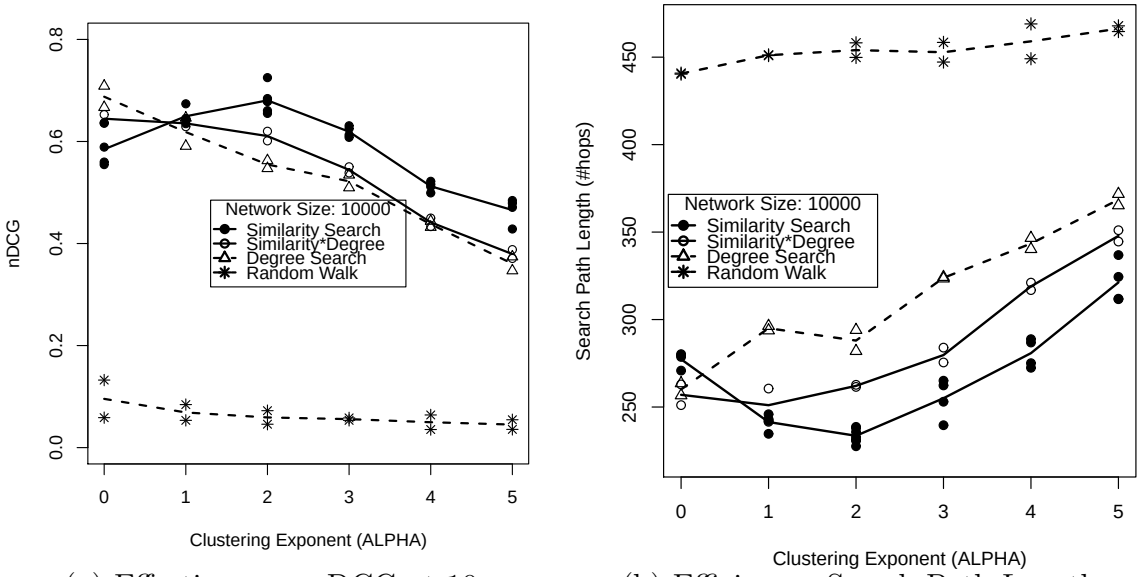


Figure D.6: SIM Search Performance F_{L200} with Varied Degree Ranges

D.3 Relevance Searches

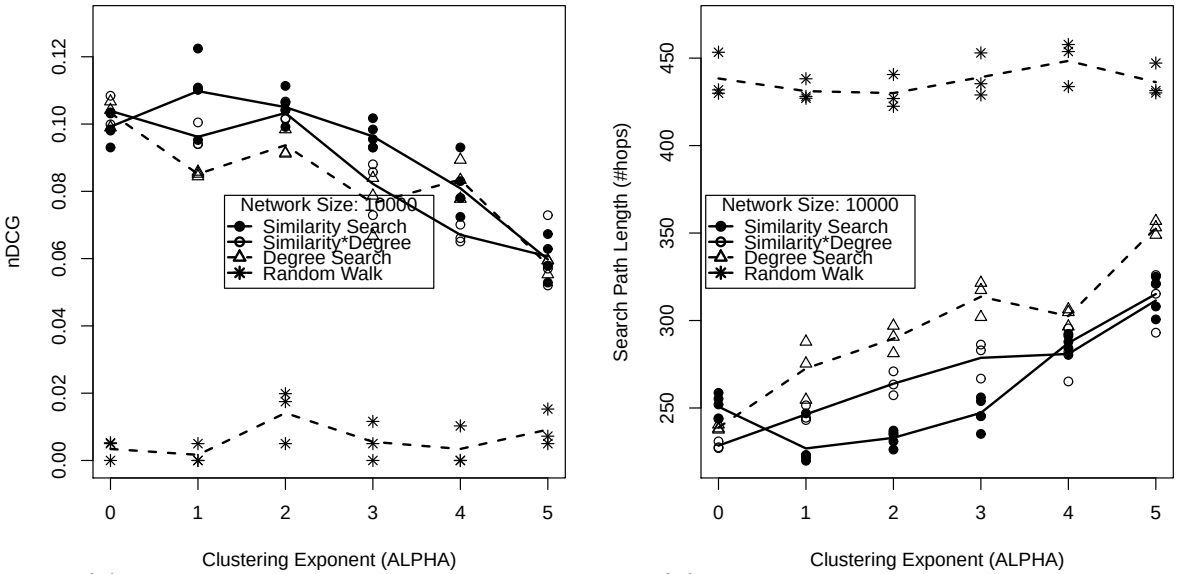


(a) Effectiveness: nDCG at 10

(b) Efficiency: Search Path Length

Figure D.7: Relevance Search Performance on 1,000-System Network

D.4 Authority Searches



(a) Effectiveness: nDCG at 10

(b) Efficiency: Search Path Length

Figure D.8: Authority Search Performance on 10,000-System Network

Appendix E

Additional Network Models

Experiments in this study mainly focused on the network model described in Section 5.2, information/documents were distributed among interconnected IR systems. Here we present additional network models that may worth investigation to better understand the impact of various network factors.

Based on the TREC data collections, two types of networks can be constructed, namely, document networks and agent networks. Document networks can be further broken down into: 1) *document network with global dimensions* (DG) (Section E), and 2) *document network with local dimensions* (DL) (Section E). The *agent network with local dimensions* (AN) (Section E) is what we used in experiments, where each agent/system hosted a collection of multiple documents and formed its neighborhood by *local* network clustering.

DG: Document Network with Global Dimensions

In the DG model, we will construct a document network with the assumption of global information. In other words, each document will be treated as an individual node that can be unambiguously represented by global VSM dimensions. The global dimensions can be derived by aggregating all documents and applying feature selection or LSI techniques (Deerwester et al., 1990; Yang, 2002). After documents are represented using the selected dimensions, connections between documents (single-document nodes) will be established based on the network (re)wiring method described in Section 4.2.3. Various combinations of power-law degree distribution exponent γ and clustering exponent

α will be studied. In this model, both queries and targets can be precisely defined. For example, a query can be constructed to find a document with specific dimensional values. This model is thus simplistic and similar to existing abstract models in complex network research (e.g., Kleinberg, 2000b; Watts et al., 2002). Nonetheless, the model will be examined based on real IR data rather than synthetic networks.

DL: Document Network with Local Dimensions

The DL model adds one layer of complexity to the DG model by removing the global dimensionality assumption. In other words, every node/agent will self-represent its (only one) document without common dimensions or any global information such as network-wide document frequency (DF) values. The relevance of a document to a query is measured using each agent's local information. Agents follow the same principles in Section 4.2.3 to connect to one other.

AN: Agent Network with Local Dimensions

The AN model, the main focus of the proposed research, is similar to the DL model. However, the AN model allows each agent to have multiple documents, making agent representation more challenging. Neither does the AN model assume global information – agents have to represent themselves using local information they have and evaluate relevance based on that. Using web data such as the ClueWeb09 collection, we can simply treat a web site as an agent and use hyperlinks between sites to construct the initial network. For a bibliographical dataset such as the TREC Genomics 2004, we can treat a scholar/author as a site/agent holding articles they have published while using collaboration data (e.g., co-authorship) to establish the initial network topology. Network clustering will then be performed based on the method described in Section 4.2.3.

Bibliography

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

Adamic, L. A., Lukose, R. M., Puniyani, A. R., and Huberman, B. A. (2001). Search in power-law networks. *Physical Review E*, 64(4):046135.

Akavipat, R., Wu, L.-S., Menczer, F., and Maguitman, A. (2006). Emerging semantic communities in peer web search. In *P2PIR '06: Proceedings of the international workshop on Information retrieval in peer-to-peer networks*, pages 1–8, New York, NY, USA. ACM.

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

Albert, R., Jeong, H., and Barabási, A.-L. (1999). Internet: Diameter of the world-wide web. *Nature*, 401(6749):130–131.

Allan, J., Aslam, J., Belkin, N., Buckley, C., Callan, J., Croft, B., Dumais, S., Fuhr, N., Harman, D., Harper, D. J., Hiemstra, D., Hofmann, T., Hovy, E., Kraaij, W., Lafferty, J., Lavrenko, V., Lewis, D., Liddy, L., Manmatha, R., McCallum, A., Ponte, J., Prager, J., Radev, D., Resnik, P., Robertson, S., Rosenfeld, R., Roukos, S., Sanderson, M., Schwartz, R., Singhal, A., Smeaton, A., Turtle, H., Voorhees, E., Weischedel, R., Xu, J., and Zhai, C. (2003). Challenges in information retrieval and language modeling: report of a workshop held at the center for intelligent information retrieval, university of massachusetts amherst, september 2002. *SIGIR Forum*, 37(1):31–47.

Amaral, L. A. N., Scala, A., Barthélémy, M., and Stanley, H. E. (2000). Classes of small-world networks. *Proceedings of the National Academy of Sciences of the United States of America*, 97(21):11149–11152.

Amoretti, M., Zanichelli, F., and Conte, G. (2006). Performance evaluation of advanced routing algorithms for unstructured peer-to-peer networks. In *valuetools '06: Proceedings of the 1st international conference on Performance evaluation methodolgies and tools*, page 50, New York, NY, USA. ACM.

Anderson, T. D. (2006). Studying human judgments of relevance: interactions in context. In *IIiX: Proceedings of the 1st international conference on Information interaction in context*, pages 6–14, New York, NY, USA. ACM.

Androutsellis-Theotokis, S. and Spinellis, D. (2004). A survey of peer-to-peer content distribution technologies. *ACM Computing Surveys*, 36(4):335–371.

Anthonisse, J. (1971). The rush in a directed graph. Technical Report BN9/71, Stichting Mahtematisch Centrum, Amsterdam.

Aslam, J. A. and Montague, M. (2001). Models for metasearch. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 276–284, New York, NY, USA. ACM.

Baeza-Yates, R., Boldi, P., and Castillo, C. (2006). Generalizing pagerank: damping functions for link-based ranking algorithms. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 308–315, New York, NY, USA. ACM.

Baeza-Yates, R., Castillo, C., Junqueira, F., Plachouras, V., and Silvestri, F. (2007). Challenges on distributed web retrieval. *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 6–20.

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

Bar-Yossef, Z. and Mashiach, L.-T. (2008). Local approximation of pagerank and reverse pagerank. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 865–866, New York, NY, USA. ACM.

Barabási, A.-L. (2009). Scale-free networks: A decade and beyond. *Science*, 325:412–413.

Barabási, A.-L. and Albert, R. (1999). Emergence of Scaling in Random Networks. *Science*, 286(5439):509–512.

Barry, C. L. and Schamber, L. (1998). Users' criteria for relevance evaluation: A cross-situational comparison. *Information Processing & Management*, 34(2/3):219–236.

Bates, M. J. (1989). The design of browsing and berrypicking techniques for the online search interface. *Online Review*, 13(5):407–424.

Bates, M. J. (1999). The invisible substrate of information science. *Journal of the American Society for Information Science*, 50(12):1043–1050.

Bates, M. J. (2006). Fundamental forms of information: Research articles. *Journal of the American Society for Information Science and Technology*, 57(8):1033–1045.

Batko, M., Dohnal, V., and Zezula, P. (2006a). M-grid: similarity searching in grid. In *P2PIR '06: Proceedings of the international workshop on Information retrieval in peer-to-peer networks*, pages 17–24, New York, NY, USA. ACM.

Batko, M., Novak, D., Falchi, F., and Zezula, P. (2006b). On scalability of the similarity search in the world of peers. In *InfoScale '06: Proceedings of the 1st international conference on Scalable information systems*, page 20, New York, NY, USA. ACM.

Baumgarten, C. (2000). Retrieving information from a distributed heterogeneous document collection. *Information Retrieval*, 3(3):253–271.

Bawa, M., Manku, G. S., and Raghavan, P. (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*, pages 306–313, New York, NY, USA. ACM.

Bekenstein, J. D. (2003). Information in the holographic universe. *Scientific American*, 289(2):58–65.

Belkin, N., Oddy, R., and Brooks, H. (1982). Ask for information retrieval: Part i. background and theory. *Journal of documentation*, 38(2):61.

Belkin, N. J. and Croft, W. B. (1987). *Retrieval techniques*, volume 22, pages 109–146. Elsevier Science Inc.

Belkin, N. J. and Croft, W. B. (1992). Information filtering and information retrieval: two sides of the same coin? *Communications of the ACM*, 35(12):29–38.

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

Bender, M., Michel, S., Triantafillou, P., Weikum, G., and Zimmer, C. (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*, pages 67–74, New York, NY, USA. ACM.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, 284(5):34–43.

Bernstam, E. V., Herskovic, J. R., Aphinyanaphongs, Y., Aliferis, C. F., Sriram, M. G., , and Hersh, W. R. (2006). Using citation data to improve retrieval from medline. *Journal of American Medical Informatics Assocication*, 13(1):96–105.

Bhavnani, S. K. (2005). Why is it difficult to find comprehensive information? implications of information scatter for search and design. *Journal of American Society for Information Science and Technology*, 56(9):989–1003.

Birukov, A., Blanzieri, E., and Giorgini, P. (2005). Implicit: an agent-based recommendation system for web search. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 618–624, New York, NY, USA. ACM.

Blair, D. C. and Maron, M. E. (1985). An evaluation of retrieval effectiveness for a full-text document-retrieval system. *Communications of the ACM*, 28(3):289–299.

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

Bowman, C. M., Danzig, P. B., Manber, U., and Schwartz, M. F. (1994). Scalable internet resource discovery: research problems and approaches. *Communications of the ACM*, 37(8):98–ff.

Brandes, U. (2001). A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25(2):163–177.

Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117.

Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., and Wiener, J. (2000). Graph structure in the web. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 33(1-6):309–320.

Brown, E. (2004). *Modern Information Retrieval*, chapter Parallel and Distributed IR, pages 229–256. Addison Wesley, New York.

Buckland, M. K. (1991). Information as thing. *Journal of the American Society for Information Science*, 42(5):351–360.

Callan, J. (2000). *Advances in Information Retrieval*, chapter Distributed Information Retrieval, pages 127–150. Springer US.

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

Callan, J. P., Lu, Z., and Croft, W. B. (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*, pages 21–28, New York, NY, USA. ACM.

Caverlee, J., Liu, L., and Bae, J. (2006). Distributed query sampling: a quality-conscious approach. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 340–347, New York, NY, USA. ACM.

Chakrabarti, S., Punera, K., and Subramanyam, M. (2002). Accelerated focused crawling through online relevance feedback. In *WWW '02: Proceedings of the 11th international conference on World Wide Web*, pages 148–159, New York, NY, USA. ACM.

Chalmers, M. (1999). Comparing information access approaches. *Journal of American Society for Information Science*, 50(12):1108–1118.

Chatman, E. (1996). The impoverished life-world of outsiders. *Journal of the American Society for Information Science*, 47(3):193–206.

Cho, J., Garcia-Molina, H., and Page, L. (1998). Efficient crawling through url ordering. In *Seventh International World-Wide Web Conference (WWW 1998)*, Brisbane, Australia.

Cleverdon, C. W. (1991). The significance of the cranfield tests on index languages. In *SIGIR '91: Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12, New York, NY, USA. ACM.

Cooper, B. F. and Garcia-Molina, H. (2005). Ad hoc, self-supervising peer-to-peer search networks. *ACM Transactions on Information Systems*, 23(2):169–200.

Craswell, N., Hawking, D., and Robertson, S. (2001). Effective site finding using link anchor information. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 250–257, New York, NY, USA. ACM.

Craswell, N. and Szummer, M. (2007). Random walks on the click graph. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 239–246, New York, NY, USA. ACM.

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

Croft, W. B. (2003). Salton award lecture - information retrieval and computer science: an evolving relationship. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 2–3, New York, NY, USA. ACM.

Cutting, D. R., Karger, D., Pedersen, J. O., and Tukey, J. W. (1992). Scatter/Gather: A cluster-based approach to browsing large document collections. In *SIGIR '92: Proceedings of the 15th annual international ACM SIGIR conference on research and development in information retrieval*, pages 318–329. ACM Press.

Davison, B. D. (2000). Topical locality in the web. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 272–279, New York, NY, USA. ACM.

Dean, J. and Ghemawat, S. (2008). Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113.

Dean, J. and Henzinger, M. R. (1999). Finding related pages on the world wide web. *Computer Networks*, 31(11–16):1467–1479.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407.

Dodds, P. S., Muhamad, R., and Watts, D. J. (2003). An Experimental Study of Search in Global Social Networks. *Science*, 301(5634):827–829.

Donato, D., Laura, L., Leonardi, S., and Millozzi, S. (2007). The web as a graph: How far we are. *ACM Transactions on Internet Technology*, 7(1):4.

Doulkeridis, C., Norvag, K., and Vazirgiannis, M. (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*, pages 35–42, New York, NY, USA. ACM.

Erdös, P. and Rényi, A. (1959). On random graphs. i. *Publicationes Mathematicae*, 6:290–297.

Farhoomand, A. F. and Drury, D. H. (2002). Managerial information overload. *Communications of the ACM*, 45(10):127–131.

Fetterly, D., Craswell, N., and Vinay, V. (2008). Search effectiveness with a breadth-first crawl. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 755–756, New York, NY, USA. ACM.

Fischer, G. and Nurzenski, A. (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*, pages 25–32, New York, NY, USA. ACM.

Flake, G. W., Lawrence, S., Giles, C. L., and Coetzee, F. M. (2002). Self-organization and identification of web communities. *IEEE Computer*, 35(3):66–71.

Foner, L. N. (1997). Yenta: a multi-agent, referral-based matchmaking system. In *AGENTS '97: Proceedings of the first international conference on Autonomous agents*, pages 301–307, New York, NY, USA. ACM.

Fox, C. J. (1983). *Information and Misinformation: An Investigation of the Notions of Information, Misinformation, Informing, and Misinforming*. Greenwood Press, Westport Conn.

Freeman, L. (1977). A set of measuring centrality based on betweenness. *Sociometry*, 40:35–41.

French, J. C., Powell, A. L., Callan, J., Viles, C. L., Emmitt, T., Prey, K. J., and Mou, Y. (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*, pages 238–245, New York, NY, USA. ACM.

French, J. C., Powell, A. L., Viles, C. L., Emmitt, T., and Prey, K. J. (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*, pages 121–129, New York, NY, USA. ACM.

Garfield, E. (1972). Citation analysis as a tool in journal evaluation. *Science*, 178:471–479.

Georgakopoulos, D. and Papazoglou, M. P., editors (2009). *Service-Oriented Computing*. The MIT Press, Cambridge, Massachusetts.

Gibson, D., Kleinberg, J., and Raghavan, P. (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*, pages 225–234, New York, NY, USA. ACM.

Girvan, M. and Newman, M. E. J. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(12):7821–7826.

Goel, S., Muhamad, R., and Watts, D. (2009). Social search in "small-world" experiments. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 701–710, New York, NY, USA. ACM.

Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70.

Granovetter, M. (1983). The strength of weak ties: A network theory revisited. *Sociological Theory*, 1:201–233.

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

Gravano, L., García-Molina, H., and Tomasic, A. (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*, pages 126–137, New York, NY, USA. ACM.

Gravano, L., García-Molina, H., and Tomasic, A. (1999). Gloss: text-source discovery over the internet. *ACM Transactions on Database Systems*, 24(2):229–264.

Guan, Z., Wang, C., Chen, C., Bu, J., and Wang, J. (2008). Guide focused crawler efficiently and effectively using on-line topical importance estimation. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 757–758, New York, NY, USA. ACM.

Gulli, A. and Signorini, A. (2005). The indexable web is more than 11.5 billion pages. In *WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 902–903, New York, NY, USA. ACM.

Han, J., Kamber, M., and Tung, A. L. H. (2001). *Spatial Clustering methods in data mining: a survey*. CRC, New York.

Hatcher, E., Gospodnetić, O., , and McCandless, M. (2010). *Lucene in Action*. Manning Publications, second edition edition.

Hawking, D. and Thomas, P. (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*, pages 75–82, New York, NY, USA. ACM.

He, B., Patel, M., Zhang, Z., and Chang, K. C.-C. (2007). Accessing the deep web. *Communications of the ACM*, 50(5):94–101.

Hearst, M. A., Karger, D. R., and Pedersen, J. O. (1995). Scatter/Gather as a tool for the navigation of retrieval results. In *Working Notes AAAI Fall Symp. AI Applications in Knowledge Navigation*.

Hearst, M. A. and Pedersen, J. O. (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*, pages 76–84, New York, NY, USA. ACM Press.

Heidemann, J., Pradkin, Y., Govindan, R., Papadopoulos, C., Bartlett, G., and Bannister, J. (2008). Census and survey of the visible internet. In *IMC '08: Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, pages 169–182, New York, NY, USA. ACM.

Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237, New York, NY, USA. ACM.

Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1):5–53.

Hersh, W., Bhuptiraju, R. T., Ross, L., Johnson, P., Cohe, A. M., and Fraemer, D. F. (2004). Trec 2004 genomics track overview. In *The Thirteenth Text Retrieval Conference: TREC 2004*, Gaithersburg, MD. National Institute of Standards and Technology.

Hu, Y. and Di, Z. (2008). Navigation in social networks: Inspired by kleinberg's model and human travel behavior research. *N/A on arxiv.org*.

Huang, Z., Chen, H., and Zeng, D. (2004). Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Transactions on Information Systems*, 22(1):116–142.

Huhns, M. N. (1998). Agent foundations for cooperative information systems. In Nwana, H. and Ndumu, D., editors, *In: Proc. s of the Third International Conference on the Practical Applications of Intelligent Agents and Multi-Agent Technology*, London.

Huhns, M. N., Singh, M. P., Burstein, M. H., Decker, K. S., Durfee, E. H., Finin, T. W., Gasser, L., Goradia, H. J., Jennings, N. R., Lakkaraju, K., Nakashima, H., Parunak, H. V. D., Rosenschein, J. S., Ruvinsky, A., Sukthankar, G., Swarup, S., Sycara, K. P., Tambe, M., Wagner, T., and Gutierrez, R. L. Z. (2005). Research directions for service-oriented multiagent systems. *IEEE Internet Computing*, 9(6):65–70.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323.

Jarvelin, K. (2007). An analysis of two approaches in information retrieval: From frameworks to study designs. *Journal of the American Society for Information Science*, 58(7):971–986.

Jarvelin, K. and Kekalainen, J. (2002). Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems*, 20(4):422–446.

Jennings, N. R. (2001). An agent-based approach for building complex software systems. *Communications of the ACM*, 44(4):35–41.

Jennings, N. R. and Wooldridge, M. J. (1998a). *Agent technology: foundations, applications, and markets*, chapter Applications of Intelligent Agents, pages 3–28. Springer-Verlag, Secaucus, NJ, USA.

Jennings, N. R. and Wooldridge, M. J. (1998b). *Agent technology: foundations, applications, and markets*. Springer-Verlag, Secaucus, NJ, USA.

Jin, R., Chai, J. Y., and Si, L. (2004). An automatic weighting scheme for collaborative filtering. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 337–344, New York, NY, USA. ACM Press.

Joachims, T., Granka, L., Pan, B., Hembrooke, H., and Gay, G. (2005). Accurately interpreting clickthrough data as implicit feedback. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 154–161, New York, NY, USA. ACM.

Kautz, H., Selman, B., and Shah, M. (1997a). Referral web: combining social networks and collaborative filtering. *Communications of the ACM*, 40(3):63–65.

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

Ke, W. and Mostafa, J. (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*, pages 49–56, Boston, USA.

Ke, W., Mostafa, J., and Fu, Y. (2007). Collaborative classifier agents: studying the impact of learning in distributed document classification. In *JCDL '07: Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, pages 428–437, New York, NY, USA. ACM.

Ke, W., Sugimoto, C. R., and Mostafa, J. (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*, Boston, MA. ACM Press.

Kim, B. M., Li, Q., and Howe, A. E. (2006). A decentralized CF approach based on cooperative agents. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 973–974, New York, NY, USA. ACM.

Kleinberg, J. (2000a). The small-world phenomenon: an algorithmic perspective. In *STOC '00: Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 163–170, New York, NY, USA. ACM.

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

Kleinberg, J. (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*, pages 210–211, New York, NY, USA. ACM.

Kleinberg, J. and Raghavan, P. (2005). Query incentive networks. In *FOCS '05: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 132–141, Washington, DC, USA. IEEE Computer Society.

Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632.

Kleinberg, J. M. (2000b). Navigation in a small world. *Nature*, 406(6798).

Kleinberg, J. M., Kumar, R., Raghavan, P., Rajagopalan, S., and Tomkins, A. S. (1999). The Web as a graph: Measurements, models and methods. *Lecture Notes in Computer Science*, 1627:1–17.

Konstan, J. A. (2004). Introduction to recommender systems: Algorithms and evaluation. *ACM Transactions on Information Systems*, 22(1):1–4.

Kurumida, Y., Ogata, T., Ono, H., Sadakane, K., and Yamashita, M. (2006). A generic search strategy for large-scale real-world networks. In *InfoScale '06: Proceedings of the 1st international conference on Scalable information systems*, page 2, New York, NY, USA. ACM.

Lambiotte, R., Blondel, V. D., de Kerchove, C., Huens, E., Prieur, C., Smoreda, Z., and Dooren, P. V. (2008). Geographical dispersal of mobile communication networks. *Physica A: Statistical Mechanics and its Applications*, 387(21):5317 – 5325.

Landauer, T., Foltz, P., and Laham, D. (1988). Introduction to latent semantic analysis. *Discourse Processes*, 25:259–284.

Li, C., Yu, B., and Sycara, K. (2007). An incentive mechanism for message relaying in unstructured peer-to-peer systems. In *AAMAS '07: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 1–8, New York, NY, USA. ACM.

Liben-Nowell, D., Novak, J., Kumar, R., Raghavan, P., and Tomkins, A. (2005). Geographic routing in social networks. *Proceedings of the National Academy of Sciences of the United States of America*, 102(33):11623–11628.

Lillis, D., Toolan, F., Collier, R., and Dunnion, J. (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*, pages 139–146, New York, NY, USA. ACM.

Lin, X., White, H. D., and Buzydlowski, J. (2003). Real-time author co-citation mapping for online searching. *Information Processing & Management*, 39(5):689–706.

Liu, K.-L., Santoso, A., Yu, C., and Meng, W. (2001). Discovering the representative of a search engine. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 577–579, New York, NY, USA. ACM.

Liu, Y., Gao, B., Liu, T.-Y., Zhang, Y., Ma, Z., He, S., and Li, H. (2008). Browserank: letting web users vote for page importance. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 451–458, New York, NY, USA. ACM.

Lu, J. (2007). Full-text federated search in peer-to-peer networks. *SIGIR Forum*, 41(1):121–121.

Lu, J. and Callan, J. (2003). Content-based retrieval in hybrid peer-to-peer networks. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 199–206, New York, NY, USA. ACM.

Lu, J. and Callan, J. (2004). Merging retrieval results in hierarchical peer-to-peer networks. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 472–473, New York, NY, USA. ACM.

Lu, J. and Callan, J. (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*, pages 332–339, New York, NY, USA. ACM.

Lu, J. and Callan, J. (2007). Content-based peer-to-peer network overlay for full-text federated search. In *8th RIAO Conference on Large-Scale Semantic Access to Content (RIAO '07)*.

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

Luu, T., Klemm, F., Podnar, I., Rajman, M., and Aberer, K. (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*, pages 41–48, New York, NY, USA. ACM.

Lv, Q., Cao, P., Cohen, E., Li, K., and Shenker, S. (2002a). Search and replication in unstructured peer-to-peer networks. In *ICS '02: Proceedings of the 16th international conference on Supercomputing*, pages 84–95, New York, NY, USA. ACM.

Lv, Q., Ratnasamy, S., and Shenker, S. (2002b). Can heterogeneity make gnutella scalable? In *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*, pages 94–103, London, UK. Springer-Verlag.

Maes, P. (1994). Agents that reduce work and information overload. *Communications of the ACM*, 37(7):30–40.

Manmatha, R., Rath, T., and Feng, F. (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*, pages 267–275, New York, NY, USA. ACM.

Marchionini, G. (1995). *Information seeking in electronic environments*. Cambridge University Press, Cambridge.

Marchionini, G. (2006). Exploratory search: from finding to understanding. *Communications of the ACM*, 49(4):41–46.

Marchionini, G. (2008). Human-information interaction research and development. *Library & Information Science Research*, 30(3):165 – 174.

Markoff, J. and Hansell, S. (2006). Hiding in plain sight, google seeks more power. *The New York Times*.

Melnik, S., Raghavan, S., Yang, B., and Garcia-Molina, H. (2001). Building a distributed full-text index for the web. *ACM Transactions on Information Systems*, 19(3):217–241.

Menczer, F. (2004). Lexical and semantic clustering by web links. *Journal of the American Society for Information Science and Technology*, 55(14):1261–1269.

Menczer, F. (2005). Mapping the semantics of web text and links. *Internet Computing, IEEE*, 9(3):27–36.

Menczer, F. and Belew, R. K. (1998). Adaptive information agents in distributed textual environments. In *AGENTS '98: Proceedings of the second international conference on Autonomous agents*, pages 157–164, New York, NY, USA. ACM Press.

Menczer, F., Pant, G., and Srinivasan, P. (2004). Topical web crawlers: Evaluating adaptive algorithms. *ACM Transactions on Internet Technology*, 4(4):378–419.

Meng, W., Yu, C., and Liu, K.-L. (2002). Building efficient and effective metasearch engines. *ACM Computing Surveys*, 34(1):48–89.

Milgram, S. (1967). Small-world problem. *Psychology Today*, 1(1):61–67.

Mooers, C. N. (1951). Zatocoding applied to mechanical organization of knowledge. *American Documentation*, 2(1):20–32. Copyright 1951 Wiley Periodicals, Inc., A Wiley Company.

Mooers, C. N. (1959). Mooers' law; or why some retrieval systems are used and others are not. *Zator Technical Bulletin*, 136.

Mooers, C. N. (1996). Mooers' law or why some retrieval systems are used and others are not. *Bulletin of the American Society for Information Science and Technology*, 23(1):22–23.

Morville, P. (2005). *Ambient Findability: What We Find Changes Who We Become*. O'Reilly Media, Inc.

Mostafa, J. (2005). Seeking better web searches. *Scientific American*, 292(2):66 – 73.

Mostafa, J., Mukhopadhyay, S., and Palakal, M. (2003). Simulation studies of different dimensions of users' interests and their impact on user modeling and information filtering. *Information Retrieval*, 6(2):199–223.

Mostafa, J., Mukhopadhyay, S., Palakal, M., and Lam, W. (1997). A multilevel approach to intelligent information filtering: model, system, and evaluation. *ACM Transactions on Information Systems*, 15(4):368–399.

Mukhopadhyay, S., Peng, S., Raje, R., Mostafa, J., and Palakal, M. (2005). Distributed multi-agent information filtering - a comparative study. *Journal of the American Society For Information Science and Technology*, 56(8):834–842.

Najork, M. A., Zaragoza, H., and Taylor, M. J. (2007). Hits on the web: how does it compare? In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 471–478, New York, NY, USA. ACM.

Netflix (2006). Netflix prize. http://www.netflixprize.com/index.

Newman, M. E. J. (2001a). Scientific collaboration networks. i. network construction and fundamental results. *Physical Review E*, 64(1):016131.

Newman, M. E. J. (2001b). Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality. *Physical Review E*, 64(1):016132.

Newman, M. E. J., Strogatz, S. H., and Watts, D. J. (2001). Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, 64:1–19. arXiv:cond-mat/0007235v2 [cond-mat.stat-mech].

Newman, M. E. J. and Watts, D. J. (1999a). Renormalization group analysis of the small-world network model. *Physics Letters A*, 263(4-6):341 – 346.

Newman, M. E. J. and Watts, D. J. (1999b). Scaling and percolation in the small-world network model. *Physical Review E*, 60(6):7332–7342.

Newman, M. E. J., Watts, D. J., and Strogatz, S. H. (2002). Random graph models of social networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(Suppl 1):2566–2572.

Nicolaisen, J. (2007). Citation analysis. *Annual Review of Information Science and Technology*, 43:609–641.

Nottelmann, H., Aberer, K., Callan, J., and Nejdl, W. (2006). The CIKM 2005 workshop on information retrieval in peer-to-peer networks. *SIGIR Forum*, 40(1):38–40.

Nwana, H. S. and Ndumu, D. T. (1998). *Agent technology: foundations, applications, and markets*, chapter A Brief Introduction to Software Agent Technology, pages 29–47. Springer-Verlag, Secaucus, NJ, USA.

O'Donovan, J. and Smyth, B. (2005). Trust in recommender systems. In *IUI '05: Proceedings of the 10th international conference on Intelligent user interfaces*, pages 167–174, New York, NY, USA. ACM.

Paepcke, A., Chang, C.-C. K., Winograd, T., and García-Molina, H. (1998). Interoperability for digital libraries worldwide. *Communications of the ACM*, 41(4):33–42.

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

Peng, S., Mukhopadhyay, S., Raje, R., and Palakal, M. (2001). A comparison between single-agent and multi-agent classification of documents. In *IPDPS '01: Proceedings of the 10th Heterogeneous Computing Workshop–HCW 2001 (Workshop 1)*, page 20090.2, Washington, DC, USA. IEEE Computer Society.

Pereira, F. B. and Costa, E. (2002). The influence of learning in the behavior of information retrieval adaptive agents. In *Proceedings of the Symposium of Applied Computing*.

Phan, T., Mohanraj, N., Powell, A., and French, J. (2000). Database selection using document and collection term frequencies. Technical report, Charlottesville, VA, USA.

Pirolli, P. and Card, S. K. (1998). Information foraging models of browsers for very large document spaces. In *AVI '98: Proceedings of the working conference on Advanced visual interfaces*, pages 83–93, New York, NY, USA. ACM.

Pirolli, P., Schank, P., Hearst, M., and Diehl, C. (1996). Scatter/Gather browsing communicates the topic structure of a very large text collection. In *CHI '96: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 213–220, New York, NY, USA. ACM.

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

Powell, A. L., French, J. C., Callan, J., Connell, M., and Viles, C. L. (2000). The impact of database selection on distributed searching. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 232–239, New York, NY, USA. ACM.

Raftopoulou, P. and Petrakis, E. G. (2008). A measure for cluster cohesion in semantic overlay networks. In *LSDS-IR '08: Proceeding of the 2008 ACM workshop on Large-Scale distributed systems for information retrieval*, pages 59–66, New York, NY, USA. ACM.

Rasmussen, E. M. (2003). Indexing and retrieval for the web. *Annual Review of Information Science and Technology*, 37(1):91–124.

Ratnasamy, S., Francis, P., Handley, M., Karp, R., and Schenker, S. (2001). A scalable content-addressable network. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 161–172, New York, NY, USA. ACM.

Ravasz, E. and Barabasi, A. L. (2003). Hierarchical organization in complex networks. *Physical Review E*, 67(2):026112.

Rijsbergen, C. J. V. (1979). *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA.

Robertson, S. (2008). The study of information retrieval: a long view. In *IIiX '08: Proceedings of the second international symposium on Information interaction in context*, pages 1–2, New York, NY, USA. ACM.

Robertson, S. E. (1997). The probability ranking principle in ir. In *Readings in information retrieval*, pages 281–286, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Ross, P. E. (2003). 5 commandments [technology laws and rules of thumb]. 40(12):30–35.

Ruthven, I. (2005). *Integrating approaches to relevance*, pages 61–80. Springer, Netherlands.

Sabater, J. and Sierra, C. (2002). Reputation and social network analysis in multi-agent systems. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 475–482, New York, NY, USA. ACM.

Salton, G. (1968). *Automatic Information Organization and Retrieval*. McGraw-Hill, New York.

Salton, G. (1986). Another look at automatic text-retrieval systems. *Communications of the ACM*, 29(7):648–656.

Salton, G., Wong, A., and Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

Saracevic, T. (1999). Information science. *Journal of the American Society for Information Science and Technology*, 50(12):1051–1063.

Saracevic, T. (2007). Relevance: A review of the literature and a framework for thinking on the notion in information science. part ii: nature and manifestations of relevance. *Journal of the American Society for Information Science and Technology*, 58(3):1915–1933.

Schafer, J. B., Frankowski, D., Herlocker, J., and Sen, S. (2007). *Collaborative filtering recommender systems*, pages 291–324. Springer, Heidelberg.

Shadbolt, N., Hall, W., and Berners-Lee, T. (2006). The semantic web revisited. *IEEE Intelligent Systems*, 21(3):96–101.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423 and 623–656.

Shapiro, C. and Varian, H. R. (1999). *Information rules : A strategic guide to the network economy*. Harvard Business School Press, Boston, Mass.

Shardanand, U. and Maes, P. (1995). Social information filtering: algorithms for automating "word of mouth". In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 210–217, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

Shneiderman, B. and Maes, P. (1997). Direct manipulation vs. interface agents. *interactions*, 4(6):42–61.

Shokouhi, M., Baillie, M., and Azzopardi, L. (2007). Updating collection representations for federated search. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 511–518, New York, NY, USA. ACM.

Shokouhi, M. and Zobel, J. (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*, pages 495–502, New York, NY, USA. ACM.

Shokouhi, M., Zobel, J., Scholer, F., and Tahaghoghi, S. M. M. (2006). Capturing collection size for distributed non-cooperative retrieval. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 316–323, New York, NY, USA. ACM.

Si, L. and Callan, J. (2003). 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*, pages 298–305, New York, NY, USA. ACM.

Si, L. and Callan, J. (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*, pages 83–90, New York, NY, USA. ACM.

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

Singh, M. P., Yu, B., and Venkatraman, M. (2001). Community-based service location. *Communications of the ACM*, 44(4):49–54.

Skobeltsyn, G., Luu, T., Zarko, I. P., Rajman, M., and Aberer, K. (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*, pages 679–686, New York, NY, USA. ACM.

Smyth, B., Balfe, E., Freyne, J., Briggs, P., Coyle, M., and Boydell, O. (2004). Exploiting query repetition and regularity in an adaptive community-based web search engine. *User Modeling and User-Adapted Interaction*, 14(5):382–423.

Sparck Jones, K. (1979). Search term relevance weighting given little relevance information. *Journal of Documentation*, 35:30–48.

Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., and Balakrishnan, H. (2001). Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 149–160, New York, NY, USA. ACM.

Tang, C., Dwarkadas, S., and Xu, Z. (2004). On scaling latent semantic indexing for large peer-to-peer systems. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 112–121, New York, NY, USA. ACM.

Tang, C., Xu, Z., and Dwarkadas, S. (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*, pages 175–186, New York, NY, USA. ACM.

Thomas, P. and Hawking, D. (2007). Evaluating sampling methods for uncooperative collections. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 503–510, New York, NY, USA. ACM.

Tsoumakos, D. (2003). A comparison of peer-to-peer search methods. In *Proceedings of the International Workshop on Web Databases*, pages 61–66.

Udupi, Y. B. and Singh, M. P. (2007). Information sharing among autonomous agents in referral networks. In *AP2PC 2007: Proceedings of the Sixth International Workshop on Agents and Peer-to-Peer Computing*.

Vakkari, P. (1999). *Task complexity, information types, search strategies and relevance: integrating studies on information seeking and retrieval*, pages 35–85. Taylor Graham Publishing, London, UK, UK.

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

Voorhees, E. M. and Harman, D. (1999). The text retrieval conference (trec): history and plans for trec-9. *SIGIR Forum*, 33(2):12–15.

Vouros, G. A. (2008). Searching and sharing information in networks of heterogeneous agents. In *AAMAS '08: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*, pages 1525–1528, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Wang, J., Pouwelse, J., Lagendijk, R. L., and Reinders, M. J. T. (2006). Distributed collaborative filtering for peer-to-peer file sharing systems. In *SAC '06: Proceedings of the 2006 ACM symposium on Applied computing*, pages 1026–1030, New York, NY, USA. ACM.

Wang, J., Reinders, M. J. T., Lagendijk, R. L., and Pouwelse, J. (2005). Self-organizing distributed collaborative filtering. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 659–660, New York, NY, USA. ACM.

Wang, Y., Yang, J.-M., Lai, W., Cai, R., Zhang, L., and Ma, W.-Y. (2008). Exploring traversal strategy for web forum crawling. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 459–466, New York, NY, USA. ACM.

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

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

Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393(6684).

White, H. D. and Mccain, K. W. (1998). Visualizing a discipline: An author co-citation analysis of information science. *Journal of the American Society for Information Science*, 49:1972–1995.

White, R. W., Bilenko, M., and Cucerzan, S. (2007a). Studying the use of popular destinations to enhance web search interaction. In *Proceedings of the 30th Annual ACM International Conference on Research and Development in Information Retrieval (SIGIR '07)*, pages 159–166, Amsterdam, The Netherlands.

White, R. W., Drucker, S. M., Marchionini, G., Hearst, M., and schraefel, m. c. (2007b). Exploratory search and hci: designing and evaluating interfaces to support exploratory search interaction. In *CHI '07: CHI '07 extended abstracts on Human factors in computing systems*, pages 2877–2880, New York, NY, USA. ACM.

Wong, S. K., Ziarko, W., Raghavan, V. V., and Wong, P. C. (1987). On modeling of information retrieval concepts in vector spaces. *ACM Trans. Database Syst.*, 12(2):299–321.

Yang, B. and Garcia-Molina, H. (2003). Ppay: micropayments for peer-to-peer systems. In *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*, pages 300–310, New York, NY, USA. ACM.

Yang, K. (2002). *Combining Text-, Link-, and Classification-based Retrieval Methods to Enhance Information Discovery on the Web.* PhD thesis, University of North Carolina at Chapel Hill.

Yang, K. (2005). Information retrieval on the web. *Annual Review of Information Science and Technology*, 39(1):33–80.

Yolum, P. and Singh, M. (2005). Engineering self-organizing referral networks for trustworthy service selection. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 35(3):396–407.

Young, P. (1987). *The Nature of Information.* Praeger Publishers, New York.

Yu, B., , and Singh, M. P. (2003). Incentive mechanisms for peer-to-peer systems. In *In Proceedings of the Second International Workshop on Agents and Peer-to-Peer Computing*, pages 77–88.

Yu, B. and Singh, M. P. (2003). Searching social networks. In *AAMAS '03: Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 65–72, New York, NY, USA. ACM.

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

Zeinalipour-Yazti, D., Kalogeraki, V., and Gunopulos, D. (2004). Information retrieval techniques for peer-to-peer networks. *Computing in Science and Engineering*, 6(4):20–26.

Zhang, H., Croft, W. B., Levine, B., and Lesser, V. (2004). A multi-agent approach for peer-to-peer based information retrieval system. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 456–463, Washington, DC, USA. IEEE Computer Society.

Zhang, H. and Lesser, V. (2005). A queueing theory based analysis of an agent control mechanism in peer-to-peer information retrieval systems. In *P2PIR '05: Proceedings of the 2005 ACM workshop on Information retrieval in peer-to-peer networks*, pages 17–24, New York, NY, USA. ACM.

Zhang, H. and Lesser, V. (2006). Multi-agent based peer-to-peer information retrieval systems with concurrent search sessions. In *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 305–312, New York, NY, USA. ACM.

Zhang, H. and Lesser, V. (2007). A reinforcement learning based distributed search algorithm for hierarchical peer-to-peer information retrieval systems. In *AAMAS '07: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 1–8, New York, NY, USA. ACM.

Zhang, J. and Ackerman, M. S. (2005). Searching for expertise in social networks: a simulation of potential strategies. In *GROUP '05: Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work*, pages 71–80, NY, USA. ACM.

Zhu, X. and Gauch, S. (2000). Incorporating quality metrics in centralized/distributed information retrieval on the world wide web. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 288–295, New York, NY, USA. ACM.

Zobel, J., Moffat, A., and Park, L. (2009). Against recall: Is it persistence, cardinality, density, coverage, or totality? *SIGIR Forum*, pages 3–15.