Visual Analysis of Distributed Search Traffic in a Peer-to-peer Network

Weimao Ke
College of Computing and Informatics
Drexel University, Philadelphia, PA 19104
Email: wk@drexel.edu, http://lincs.ischool.drexel.edu

***Abstract*—We study distributed searches in fully decentralized peer-to-peer networks. This papers reports on a study that aims to understand the robustness and load balancing of a distributed search network. We focus on a visual analysis of search traffic, load distribution, and edge density when the network experiences adversaries. The analysis reveals that the network tends to rely more heavily on an even smaller portion of nodes for searches when an increasing number of nodes become unavailable. In the meanwhile, nodes become more exploratory and tend to engage more of those previously unutilized. The trade-off between a more skewed traffic distribution and a greater edge density in related scenarios is important to the robustness of the network and requires further investigation.**

***Keywords*-Distributed search; Robustness; Load balancing; Efficiency; Peer-to-peer networks; Decentralization**

I. Introduction

Decentralization is the nature of many naturally, socially, and technologically grown structures that scale. The Web and the Internet operate in a rather decentralized manner without explicit global control. On the Internet, fundamental technologies such as routing and lookup operations are decentralized by design and are able to scale with the rapid growth of the network.

From the perspective of the Internet of everything, where any tiny gadgets and daily items can be attached to a highly interconnected digital world, the question of finding relevant pieces of information about and on these devices is staggeringly challenging. While centralization is likely to fail in the long term, decentralization represents the future of technological innovation; searching is an essential part of the trend.

Our research envisions a fully decentralized architecture in which individual search engines can interconnect and contribute to the collective power of finding relevant information, through distributed routing or network traversal. The goal of this research is to understand the general mechanisms by which a large number of distributed systems can work together to support scalable search and retrieval operations. It aims to explore alternative search engine architectures that can function, scale, and cope with the increasing magnitude and dynamics of networks such as the Web.

II. Related Work

Related challenges facing distributed or decentralized searches have been studied in areas of distributed (federated) information retrieval, peer-to-peer networks, multi-agent systems, and complex networks [16]. Classic distributed IR research has focused on distributed database content and characteristics discovery, database selection, and result fusion in a relatively small number of distributed, persistent information collections [9].

Peer-to-peer IR research often involves a larger number of distributed systems which dynamically join and leave the community. Related projects have employed techniques such as distributed hashing tables (DHTs) in structured P2P networks and semantic overlay networks (SONs) in unstructured networks for efficient discovery [6]. Agent-based modeling has proven to be a powerful tool in distributed information retrieval (IR) research [10]. The agent paradigm has been extensively used to model processes such as P2P search [24], intelligent crawling [20], and expert finding [15].

The central idea of peer segmentation or clustering in frameworks such as SONs to support efficient decentralized searches has also been studied in complex network research. In networks with small world properties, studies have demonstrated that globally relevant targets can be found efficiently through collaboration of distributed, local intelligence in large networks [3].

Our research has studied related decentralized/distributed information retrieval problems in light of network formation and clustering, emergent from interconnectivity among distributed systems. In a series of large-scale IR experiments we conducted, network clustering based on semantic overlay was found to be useful for decentralized searches, however, with qualifications [11], [12]. The best search efficiency and effectiveness were supported by a very specific level of network clustering. Any departure from that fine-tuned level, i.e. stronger or weaker clustering, degraded search performance significantly [13].

III. Search Scalability & Robustness

The small world phenomenon suggests that distributed searches can be conducted to find results within a short radius, regardless of the network size. However, 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.

A. Network Clustering and Searches

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 [1], [5], [19], [6]. Hence, overall, clustering was often regarded as beneficial whereas the potential *negative* impact of clustering (or over-clustering) on retrieval has often been overlooked.

Research on complex networks has found that a proper level of network clustering with some presence of remote connections has to be maintained for efficient searches [17], [23], [18], [21], [3]. Clustering reduces the number of "irrelevant" links and aids in creating topical segments useful for orienting searches. Without sufficient clustering, the network has too much randomness to guide efficient traversals because *weak ties* dominate. While searches may jump quickly from one place to another (hops) in the network space, there is no "gradient" to lead them toward targets. With very strong clustering, on the other hand, a network tends to be fragmented into local communities with abundant *strong ties* but few *weak ties* to bridge remote parts [7], [22]. Although searches might be able to move gradually toward targets, necessary "hops" become unavailable.

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 [7].

One key parameter widely used in complex network research for studying the impact of clustering is the *clustering exponent* α. [17] studied decentralized search in small world using a two dimensional model, in which peers had rich connections with immediate neighbors and sparse associations with remote ones. The probability p_r of connecting to a neighbor beyond the immediate neighborhood was proportional to $r^{-\alpha}$, where r was the search distance between the two in the dimensional space and α a constant called *clustering exponent*. It was shown that only when *clustering exponent* $\alpha = 2$, search time (i.e., search path length) was optimal and bounded by $c(\log N)^2$, where N was the network size and c was some constant [16]. More generally, when $\alpha = d$ on a d-dimension space, decentralized search is optimal. Further studies conducted on small world networks as well as in distributed IR have shown consistent results [18], [21], [12], [13].

B. Verified Scalability Model

Our research has focused on search efficiency and scalability with growing network sizes N and varied (distributed system) neighborhood size d distributions (degree distributions). We have studied and validated a scalability function which we discuss below.

Let L denote search path length, the number of hops (distributed systems) a search has to traverse the network to reach a desired target. According to [16] and several studies in distributed IR, when network clustering is optimal, a reasonable relationship between $\hat{L}$ (expected value of L based on relevance/similarity searches) and network size N (the number of distributed systems in the network) is:

$$\hat{L} = \beta' \cdot (\log_b N)^{\lambda} \tag{1}$$

where β' is a constant and b is the logarithmic base. λ is an exponent parameter to be identified with empirical data.

Assume the majority of distributed systems (hops) have a neighborhood size (number of interconnected systems) d_m. Let L_g denotes the ideal search path length given a (imagined) perfect, global index of all distributed systems. It can be shown that $L_g \propto \log_b N$ as well as $N \approx d_m^{L_g}$:

$$L_g \propto \log_b N \tag{2}$$

$$\approx \log_{d_m} N \tag{3}$$

Hence, we can replace $\log_b N$ with $\log_{d_m} N$ (i.e. using d_m as the logarithmic base) in Equation 1, which becomes:

$$\hat{L} = \beta \cdot (\log_{d_m} N)^{\lambda} \tag{4}$$

$$= \beta \cdot (\log N / \log d_m)^{\lambda} \tag{5}$$

where β is a constant and d_m the neighborhood size (degree) of majority distributed systems. To simulate real networks, a power-law function will be used for degree distribution $d \in [d_m, d_x]$, where d_m is the min degree (which the vast majority have in a power law) and d_x is the maximum value (which only a small number of nodes have).

C. Research Focus on Robustness

In a previous study, we validated the scalability model with large-scale experiments, identified the exponent λ, estimated the β coefficient with varied N and d_m settings, and predicted potential search efficiency in real-scale environments with millions to billions distributed systems [14].

In this study, we aim to better understand how distributed service nodes can continue to function efficiently in unstable environments. In particular, we are interested to know whether the system as a whole can stand and how well it performs when a significant number of members become unavailable and unresponsive due to individual failures or attacks.

IV. Simulation Framework and Algorithms

We have developed a decentralized search simulation framework based on multi-agent systems for finding relevant information in distributed settings. Each agent (node) represents an search (retrieval) system, which has its document collection and can connect to others to route queries. The simulation system was implemented in Java the multi-agent system JADE [2] and full-text search library Lucene [8].

In the simulation framework, each node builds an index on a local document collection and connects to a number of neighbors (a variable in this study) for help with unanswered queries. When a node receives a query/task, it first searches its local collection and, if the result is unsatisfactory, forwards the query to one of its neighbors most likely to have relevant information. The query routing continues until relevant results have been found or when it reaches a limit (i.e. max search path length). Further details on the simulation framework can be found in [13].

The subsections below elaborate on specific algorithms implemented in the framework for 1) information representation and weighting (to represent documents and queries), 2) neighbor (node) representation, 3) neighbor selection (search) method, and 4) a network interconnectivity (clustering) function.

A. Basic Functions

*1) TF*IDF Information Representation:* Each node processes information it individually has and produces a local term space, which is used to represent each information item using the classic TF*IDF (Term Frequency * Inverse Document Frequency) weighting scheme. Note that IDF values are based on the node's local collection.

*2) DF*INF Neighbor Representation:* A node uses a meta-document to represent each of its neighbors. The weight of term t in a meta-document is computed by: $W'(t) = df'(t) \cdot log(\frac{N'_b}{nf'(t)})$, where $df'(t)$ is the number of documents in the neighbor node (collection) containing term t, N'_b is the total number of the node's neighbors (meta-documents), and $nf'(t)$ is the number of neighbors containing the term t. We refer to this function as *DF*INF*, or Document Frequency * Inverse Neighbor Frequency.

3) Similarity Scoring Function: Given a query q, the similarity score of a document d matching the query is computed by: $\sum_{t \in q} W(t) \cdot coord(q,d) \cdot queryNorm(q)$, where $W(t)$ is the weight of term t given by the above TF*IDF or DF*INF, $coord(q,d)$ 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. This scoring function is used to compute query-document similarities as well as query-metadocument (query-neighbor) similarities.

B. Neighbor Selection (Search) Methods

We use the following strategies to decide which neighbors should be contacted for the query: 1) SIM (Similarity) Search which selects the neighbor with the highest similarity score, and 2) Sim*Deg (similarity times degree) which combines similarity and degree scores to determine the best neighbor.

C. Interconnectivity and Network Clustering

To interconnect nodes, the first step is to determine how many links (degree d_u) each distributed node 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. Based on the ClueWeb data, given the number of incoming hyperlinks d'_u of system/site u, the normalized degree is computed by:

$$d_u = d_m + \frac{(d_x - d_m) \cdot (d'_u - d'_m)}{d'_x - d'_m} \tag{6}$$

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

$$p_{uv} \propto r_{uv}^{-\alpha} \tag{7}$$

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

V. Experimental Design

A. Data Collection

We rely on the ClueWeb09 Category B collection created by the Language Technologies Institute at Carnegie Mellon University for IR experiments. The ClueWeb09 collection contains roughly 1 billion web pages and 8 billion outlinks crawled during January - February 2009. The Category B is a smaller subset containing the first crawl of 50 million English pages from 3 million sites with 454 million outlinks. The ClueWeb09 dataset has been adopted by several TREC tracks including Web track and Million Query track [4]. Additional details about the ClueWeb09 collection can be found at http://boston.lti.cs.cmu.edu/Data/clueweb09/.

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

B. Network Model and Sizes

Each node represents an IR system serving a collection of pages (documents). We assume that there is no global information about all document collections. Nor is there centralized control over individual nodes. Nodes have to represent

themselves using local information they have and evaluate relevance based on that. Using the ClueWeb09 collection, we treat a web site/domain as a distributed system/node and use hyperlinks between sites to construct the initial network.

We first construct a list of all web domains in the category B subset with at least one in-link in the provided web graph. We take the first 1000 web sites to construct the initial network and extend it to 10000 nodes/systems. Network clustering is performed using the method described in Section IV-C to establish individual node neighborhoods. We use an observed optimal clustering exponent $\alpha = 2$ in the experiments.

C. Search Task - Rare Item Search

Given the size of the web (and likewise the ClueWeb09 collection), it is nearly impossible to manually judge the relevance of every document and establish a complete relevance base. Hence, we primarily rely on existing evidence in data to do automatic relevance judgment. We use documents (with title and content) as queries for decentralized searches. From the first 1000 web domains constructed above, we select as queries 12 random web pages with at least 3 in-links. The final set of query documents include (all trecids with prefix *clueweb09-en000*): 1-42-03978, 1-73-04287, 1-90-26216, 2-73-04700, 2-91-14776, 3-27-30577, 3-30-28328, 3-51-10345, 3-55-31539, 4-61-19060, 4-72-24215, 4-92-04942.

The search task is to find the exact copy of a given document (query). Specifically, when a query document is assigned to an node, 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 node 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. The extreme rarity will pose a great challenge on the proposed decentralized search methods.

D. Degree Distribution: $[d_m, d_x]$

We will use the degree (in-degree) distribution of the ClueWeb09B hyperlink graph and normalize the distribution to fall within a range $[d_m, d_x]$. With different d_m and d_x values, the degree distribution will continue to follow a power-law pattern in which the majority of nodes have the degree of d_m. We use degree ranges $d_u \in [16, 64]$ and $[64, 128]$, to examine the impact of degree distribution (neighborhood size) on decentralized searches.

E. Fraction of Unavailable Nodes f

We simulate the number of nodes that become unavailable and cannot provide routing services to help route any query. In the experiments, we randomly select a number of service nodes (from 10% to 80% of the entire network) and make them unresponsive. However, we make sure the target nodes with relevant information to search queries are available so that all searches can be conducted successfully.

F. Evaluation

In previous research we focused on the evaluation of search effectiveness and efficiency, using classic metrics based on precision, recall, and search time. In this study, we shift our focus to understanding dynamics and utility of the network to support distributed searches. With the varied fraction of unavailable nodes in different network settings, we examine the overall network traffic and load balancing using the following methods: 1) edge density, which computes the ratio of the number of edges (paths of search requests) and the number of possible edges, 2) cumulative distributions of the number of search requests to each node, and 3) network visualization of searched nodes to explore the underlying network structured formed by the search traffic. While the first two approaches enable quantitative evaluation of the search network, network visualization enables more qualitative evaluation and preliminary observation that can be explored further.

G. Parameter Settings

The list below summarizes major variables discussed above. We use full combinations of these parameters in experiments, i.e., 1 (network size) $\times$ 2 (degree ranges) $\times$ 2 (search methods) $\times$ 5 (simulations of node unavailability).

- Network sizes $N = 10,000$ and max search path lengths $L_{max} = N$;
- Degree ranges $d \in$: $[16, 64]$ and $[64, 128]$;
- Search methods: Similarity (SIM) search and similarity*degree (SimDeg) search.
- Fraction of unavailable nodes: we vary the portion of unavailable nodes from 0 (all available), to 10%, 20%, 40%, and 80%.

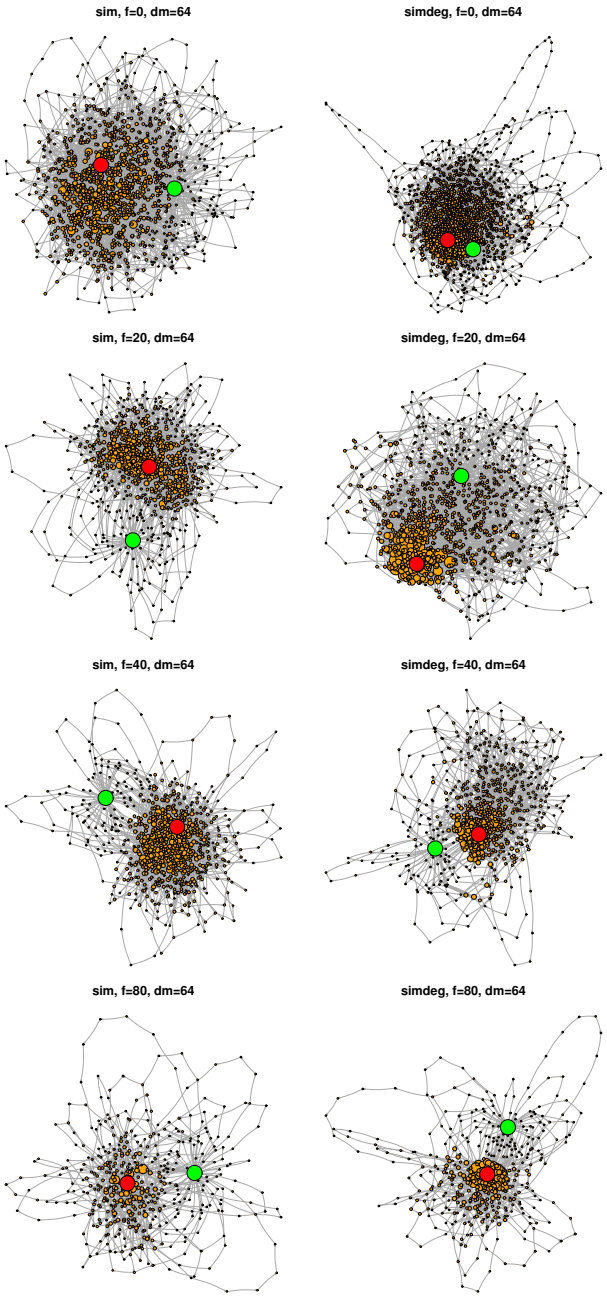
VI. Results

A. Network Visualization and Analysis

With the recorded chains of nodes engaged in the searches, we visualize the searched network under different parameter settings. As shown in Figure 1, each network is produced using the force-directed Fruchterman-Reingold layout. With the exception of green (query starting point) and red (target) nodes, the (orange) node size is proportionate to the number of search requests one has received whereas an arrow indicates the direction of requests.

Examination of the visualizations reveals that when all nodes are available (top plots in Figure 1 with $f = 0$), search requests are highly distributed among many network nodes. When f (fraction of unavailable nodes) increases, the network becomes increasingly sparse and there is a much smaller portion of nodes that are actively engaged in the searches (e.g. as indicated by a smaller number of larger yellow circles in plots closer to the bottom of Figure 1).

Comparing Similarity search (left) and Sim*Degree search (right) in Figure 1, it appears that Sim*Deg searches engage a smaller portion of nodes that are disproportionate in their load (circle size). This is likely due to the fact that Sim*Deg considers connectivity (out-degree) as a major factor in deciding



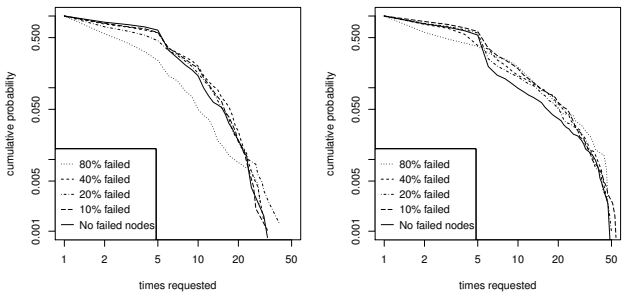
c) Similarity Search d) Sim*Degree Search

Fig. 1. Network visualization of searched nodes with Similarity Search (left column) and Sim*Degree Search (right column), with $d_m = 64$. From top to bottom, the fraction of unavailable nodes f increases from 0% to 80%. The green node is where queries are initially issued (routing starts) whereas the red node is where targets are (routing ends).

which neighbor nodes should be included in the search/routing process.

B. Cumulative Distribution of Traffic

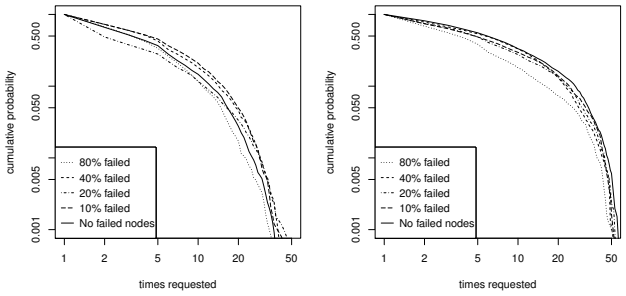
The visualizations seem to suggest that nodes are disproportionately engaged in search activities with an increasing number of unavailable nodes. We plot cumulative distributions of traffic (search requests to each node) given the varied



a. Similarity Search b. Sim*Degree Search

Fig. 2. Cumulative distribution of load (search requests), with $d_m = 64$.

parameters. Figure 2 plots the cumulative distribution for sim search and sim*degree search with $d_m = 64$, where each curve is on a specific f (fraction of unavailable nodes) setting.



a. Similarity Search b. Sim*Degree Search

Fig. 3. Cumulative distribution of load (search requests), with $d_m = 16$.

As Figure 2a shows, the distribution of $f = 80$ (80% unavailable) departs very much from that of $f = 0$ (all nodes available), and the distribution curve is flatter. This tells that with a large number of unavailable nodes, there is a much greater discrepancy among the heavily loaded nodes and others that are lightly engaged in searches. The same can be observed from distribution plots for Sim and Sim*Degree searches with $d_m = 16$.

The overall observation of these distribution plots is that when a large number of nodes become unavailable randomly, the network tends to rely more heavily on even fewer nodes for query routing. Even though our previous research showed that searches can be more efficient due to a downsized network, this is at the expense of more highly connected nodes and the load is more unevenly distributed.

C. Searched Network Density

We look at *edge density*, which is defined as the number of edges (i.e. search requests from one node to another) as the ratio to the number of all possible edges. We plot the the overall network edge density vs. the fraction of unavailable nodes f in Figure 4.

As Figure 4 shows, edge density increases with an increasing number of unavailable nodes for both Similarity and Sim*Degree searches. This is advantageous as it suggests that

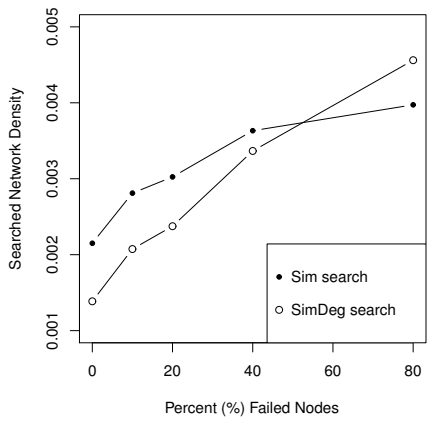


Fig. 4. Network Density vs. f Fraction of Unavailable Nodes

nodes are more exploratory and/or use a larger portion of other nodes that were not previously engaged in searches when the network shrinks in size due to node unavailability. The interplay of this impact on edge density and that on traffic distribution is important to the overall utility of the search network and should be studied further.

VII. Conclusion

In this paper we report on a visual analysis of search traffic, load distribution, and edge density of a network for distributed searches. We focus on the impact of node unavailability on the robustness and load balancing of the network. Results show that the network relies more heavily on an even smaller portion of nodes for searches when an increasing number of nodes become unavailable. Nodes also tend to be more exploratory and ultimately engage more of those previously unutilized.

In general, increased node unavailability (e.g. due to attacks) in the network leads to a more skewed traffic distribution and a greater edge density. Understanding the trade-off in related scenarios is important to the robustness of the network for distributed searching and requires further investigation.

Acknowledgment

This work was supported in part by the National Science Foundation under grant no. 1646955.

References

[1] M. Bawa, G. S. Manku, and P. Raghavan. 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, 2003. ACM.

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

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

[4] C. L. A. Clarke, N. Craswell, and I. Soboroff. Overview of the TREC 2009 Web Track. In *Proc. of TREC-2009*, 2009.

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

[6] C. Doulkeridis, K. Norvag, and M. Vazirgiannis. 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, 2008. ACM.

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

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

[9] D. Hawking and P. Thomas. 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, 2005. ACM.

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

[11] W. Ke and J. Mostafa. 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, July 23 2009.

[12] W. Ke and J. Mostafa. Scalability of findability: effective and efficient ir operations in large information networks. In *SIGIR'10: Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 74–81, 2010.

[13] W. Ke and J. Mostafa. Studying the clustering paradox and scalability of search in highly distributed environments. *ACM Transactions on Information Systems*, 31(2):8:1–8:36, May 2013.

[14] W. Ke and J. Mostafa. Scalability analysis of distributed search in large peer-to-peer networks. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 909–914, Washington, DC, 12/2016 2016.

[15] W. Ke, J. Mostafa, and Y. Fu. 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, 2007. ACM.

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

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

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

[19] J. Lu and J. Callan. 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, 2006. ACM.

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

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

[22] M. P. Singh, B. Yu, and M. Venkatraman. Community-based service location. *Communications of the ACM*, 44(4):49–54, 2001.

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

[24] H. Zhang and V. Lesser. 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, 2007. ACM.

AUTHOR'S BACKGROUND

Your Name	Title	Research Field	Personal Website
Weimao Ke	Associate Professor	Information Retrieval, Cloud Computing	https://lincs.cci.drexel.edu/