EVALUATION OF FIND-SIMILAR WITH SIMULATION AND NETWORK ANALYSIS

A Dissertation Presented

by

MARK D. SMUCKER

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 2008

Computer Science

EVALUATION OF FIND-SIMILAR WITH SIMULATION AND NETWORK ANALYSIS

A Dissertation Presented

by

MARK D. SMUCKER

Approved as to style and content by:

James Allan, Chair

W. Bruce Croft, Member

David D. Jensen, Member

Donald L. Fisher, Member

Andrew G. Barto, Department Chair
Computer Science

ACKNOWLEDGMENTS

First and foremost, my deepest thanks go to my wife, Anne. Her strength and help have made earning this Ph.D. possible. I very much thank Anne for being there for me throughout every moment. I thank my son Leo for the joy he has brought to Anne and me.

I thank my advisor, James Allan, for providing both intellectual and financial support for all of my time at the University of Massachusetts. James has allowed me a wide latitude in my research directions, which I greatly appreciate. I have enjoyed having James as my advisor and greatly appreciate the influence James has had on my life.

I thank Bruce Croft for his feedback and assistance during my time as a graduate student in the Center for Intelligent Information Retrieval (CIIR). While James was my advisor, Bruce's influence on the CIIR is enjoyed by all of the CIIR's students, and for that I am grateful.

Thanks go to my other dissertation committee members, David Jensen and Donald Fisher, for their time and helpful feedback.

One of the reasons I picked the University of Massachusetts for graduate school was the excellent staff of the CIIR and the computer science department. In particular, I thank very much Kate Moruzzi for all of her help. I also greatly appreciate the aid and advice of David Fisher. I thank Bob Armstrong for joining the CIIR and taking both GALE and Internano and many other tasks off my hands and thus allowing me the time to finish this degree. Thanks also to Jean Joyce, Andre Gauthier, and Glenn Stowell for making the CIIR a great place to be a graduate student. In the

computer science department, I would like to thank Sharon Mallory, Pauline Hollister, Leeanne Leclerc, and Claire Christopherson for being especially helpful and making the computer science department a great place to work. There are many more staff in the department, including CSCF, and while I cannot name them all, I would like to thank them all for their help.

I also thank my fellow graduate students for making the CIIR and the computer science department great places to work and do research. In particular, I thank Trevor Strohman for his friendship as well as for many excellent conversations about research, computer science, and life. I thank Vanessa Murdock and Hema Raghavan for always inviting me to lunch as well as for their help, support, and great conversation. Giridhar Kumaran and Ao Feng helped make the GALE project a success, and I am glad to have worked alongside them both. Thanks to Ben Carterette for his help and support. Thanks to Don Metzler for his help with the dependence models in Chapter 3 as well as for being a great fellow graduate student. Thanks to Fernando Diaz for his early support of this work. I thank my fellow CIIR graduate students for the chance to learn from them and be part of the leading edge of IR research: Nasreen Abdul Jaleel, Elif Aktolga, Niranjan Balasubramanian, Ron Bekkerman, Michael Bendersky, Marc Cartright, Henry Feild, Shaolei Feng, Jiwoon Jeon, Jinyoung Kim, Victor Lavrenko, Xiaoyan Li, Xiaoyong Liu, Natasha Mohanty, Ramesh Nallapati, Desislava Petkova, Jeremy Pickens, Toni Rath, Jangwon Seo, Chirag Shah, Courtney Wade, Xing Wei, Xiaobing Xue, Xing Yi, and Yun Zhou. Outside of the CIIR, I thank fellow students Lisa Friedland, Jerod Weinman, Michael Hay, and David Mimno for their support, conversation, and help.

I very much thank my parents, David and Carol, for their support during all of my years and especially for their help in achieving this degree; their aid was generous, and I feel very blessed to have them as my parents. I thank my sister Christine and her husband Andrew for visiting Anne and me and helping us when in need. I thank

my sister LynAnne and her husband Dagon. I thank my brothers-in-law, David and John, and their wives, Marie and Jess, for their visits and support.

I thank Craig Shankwitz for being a truly great friend for over 20 years.

I thank Rev. Lyle Seger and all the members of Wesley United Methodist Church for their help and support.

I thank Susan Dumais for the valuable advice and feedback I received both before graduate school and also as part of the Microsoft Fellowship interview process.

I thank David Kulp for co-supervising my synthesis project and for his help with the various fellowship applications that I submitted. I thank Mark Corner for the opportunity to teach CS197C and for his help with my job search.

My graduate education included two years at the University of Wisconsin-Madison. I thank Martin Reames and Suzan Stodder for being great friends and great office-mates. I also thank Paul Bradley for being a friend, for his help, and for staying in touch over the years. I thank Gil Gribb for being a model of independence and for many great times at the Rathskeller. I thank Todd Turnidge, Marc Shapiro, and Elton Glaser for their friendship. I thank Scot McCollum for putting up with me as a housemate, for his friendship, and for building me the most awesome commuter bike. I thank Una-May O'Reilly for friendship, advice, consistent enthusiasm, and for being a model of energy. I thank Una-May and Blake LeBaron for providing me an escape from graduate school as a dogsitter of the great Pawla and Barney. Finally, I thank Rik Belew for introducing me to information retrieval and for his guidance and advice.

My education as a researcher began as an undergraduate at Iowa State University. I thank Dan Ashlock, Ann Stanley, and Leigh Tesfatsion for welcoming me into their research project and letting me experience the excitement of discovery and the pleasure of a good collaboration. I thank David A. Lewis for his mentorship and the chance to be a member of his research team during all my years at Iowa State.

In between the University of Wisconsin-Madison and the University of Massachusetts Amherst, I received a valuable education in industry. I thank Nancy Benovich Gilby, Ted Smith, Michael Wolf, and Mike Moskowitz for their mentorship. I also thank Michael Wolf for his help with my graduate school applications. I thank Jesse Hull, Paul Perry, Steven Freedman, John Chandler-Pepelnjak, John McNulty, Jonathon Sheena, David Tiu, Jill Shay, Chris Berg, and Erik Snowberg.

In November of 2007, Jimmy Lin approached me with an idea to take the simulation I had developed in Chapter 3 and use it to examine PubMed. Jimmy's hypothesis was that even as initial retrieval performance decreases as measured by MAP, that with find-similar available in PubMed, overall performance would not decrease as quickly, for find-similar would act to compensate. The experiments of Chapter 3 had shown some evidence that find-similar worked in this way, but these experiments only looked at find-similar improving a single retrieval method. The result of our collaboration was a paper (Lin and Smucker, 2008). Some of the ideas we generated about simulation can now be found in the introduction to Chapter 3. Chapter 4 is based on my collaboration with Jimmy, but presents another, expanded analysis of our experiments. I very much thank Jimmy for asking me to work with him, and I also thank him for his permission to include portions of our work in this dissertation.

Portions of the remaining chapters contain expanded and revised versions of previously published material (Smucker and Allan, 2006, 2007a,b). I thank Diane Kelly for her feedback on the SIGIR version of Chapter 3. I thank the anonymous reviewers of these papers for their helpful feedback.

I am certain that I have forgotten to thank many people. I apologize if your name does not appear here, and I do thank you very much for your help.

All of my education has been at public institutions and much of my employment has come from government research grants. I very much thank the U.S. taxpayers for their support of education and science.

This work was supported in part by the Center for Intelligent Information Retrieval, in part by the Defense Advanced Research Projects Agency (DARPA) under contract number HR0011-06-C-0023, in part by SPAWARSYSCEN-SD grant number N66001-02-1-8903, and in part by NSF Nanotech # DMI-0531171. Any opinions, findings and conclusions or recommendations expressed in this material are the author's and do not necessarily reflect those of the sponsor.

ABSTRACT

EVALUATION OF FIND-SIMILAR WITH SIMULATION AND NETWORK ANALYSIS

SEPTEMBER 2008

MARK D. SMUCKER

B.S. Physics, IOWA STATE UNIVERSITY

B.S. Computer Science, IOWA STATE UNIVERSITY

M.S., UNIVERSITY OF WISCONSIN–MADISON

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor James Allan

Every day, people use information retrieval (IR) systems to find documents that satisfy their information needs. Even though IR has revolutionized the way people find information, IR systems can still fail to satisfy people's information needs. In this dissertation, we show how the addition of a simple user interaction mechanism, *find-similar*, can improve retrieval quality by making it easier for users to navigate from relevant documents to other relevant documents. Find-similar allows a user to request documents similar to a given document. In the first part of the dissertation, we measure find-similar's retrieval potential through simulation of a user's behavior with hypothetical user interfaces. We show that find-similar has the potential to improve the retrieval quality of a state-of-the-art IR system by 23% and

match the performance of relevance feedback. As part of a case study that first shows how find-similar can help PubMed users find relevant documents, we then show how find-similar responds to varying initial conditions and acts to compensate for poor retrieval quality. In the second part of the dissertation, we characterize find-similar in the absence of a particular user interface by measuring the quality of the document networks formed by find-similar's document-to-document similarity measure. Find-similar effectively creates links between documents that allow the user to navigate documents by similarity. We show that find-similar's similarity measure affects the navigability of the document network and how a query-biased similarity measure can improve find-similar. We develop measures of network navigability and show that find-similar should make the World Wide Web more navigable. Taken together, the simulation of find-similar and the measurement of the navigability of document networks shows how find-similar as a simple user interaction mechanism can improve a user's ability to find relevant documents.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

Search users turn to information retrieval (IR) systems when they have some information need to satisfy.[1] IR systems can be seen as involving three major layers. The top layer is the interface that provides the means for the user to interact with the IR system. The middle layer consists of the various algorithms used by the IR system. Chief among these algorithms is usually an algorithm to rank documents in response to a user's queries. Finally, the bottom layer is composed of data structures and other systems-like components designed to allow retrieval algorithms to be efficient and scalable.

While we can think of an IR system in terms of these layers, the divisions between the layers are not sharply defined. The interface depends on the algorithms which depend on the system level components. Similarly, we only are concerned with algorithms that can effectively utilize the input gathered from the user.

Understanding the meaning of a text document is an unsolved problem in IR and artificial intelligence (AI). Humans are the only known entities capable of reading comprehension. While both IR and AI researchers make continual progress towards machine intelligence capable of reading comprehension, human intelligence is still many orders of magnitude greater than machine intelligence and this is likely to be the case for the foreseeable future.

---

[1]Information retrieval research covers a wide range of information seeking and organizing needs. Text retrieval focuses on helping people find documents that are relevant to their information need. In this dissertation, all references to IR will be to text retrieval systems.

In what we might call the "human replacement" view of information retrieval, the IR system is supposed to act like an automated human that can understand the needs of the user and also understand all of the documents that the system stores. This artificially intelligent IR system is able to then quickly discern the documents that the user will consider relevant.

The human-replacement view stresses the creation of intelligence. In contrast, we take the position that IR systems should work to utilize the wealth of intelligence available in the human user. We see this as a *tool-centric* view of IR.

This perspective is not new in IR. An early example of this sentiment comes from Doyle (1962) who wrote that "humans should be capable of doing a better job of searching than machine, if confronted with well-organized material." Doyle gave keyword in context (KWIC) indexes an example of a way to better organize material. KWIC indexes arrange text phrases such that as one scans down a page, the phrases are all horizontally aligned with the same keyword in the middle of the line of text.

IR has improved and evolved a tremendous amount since 1962. Even so, IR, in the form of search engines that return a ranked list of documents to the user, is still primarily a tool that provides sophisticated best-match results given the user's natural language query.

An IR system's user interface can been seen to be composed of many tools. Some tools are better described as user interface features or interactive elements while other tools are well described as interaction mechanisms. Interface tools can be divided into three categories:

1. Tools to help find documents, i.e. search tools.

2. Tools to enable faster comprehension of documents.

3. Tools to help organize documents.

While users typically use tools for finding and understanding documents during a search, and organization tools for the saved documents after a search, the various types of tools can be useful at different times.

Some of these tools are used in concert with each other. For example, the *query box* in a standard interface is one tool and it gets used with the *results list* that displays the results of the query. There can be many different implementations of each tool. A few other example tools include the query language, term highlighting, term search within a document, spelling suggestions, document summaries, and interactive query expansion.

In this dissertation, we focus our research on one interaction mechanism, *find-similar*. We examine both the potential of find-similar as well as show how to improve its performance. Find-similar is a search tool that allows a user to request documents similar to a given document. Much in the spirit of tools promoted by Doyle, find-similar provides an advanced organization of the documents that aims to improve users' ability to find relevant documents.

1.1 Find-Similar

The typical search scenario supported by today's text retrieval systems allows a user to enter a query and receive a ranked list of documents. The IR system attempts to understand what the user's information need is based on the query and likewise attempts to provide documents that will best satisfy the user's information need.

In this dissertation, our work is motivated by the many user tasks that require finding more than one relevant document. Examples include:

- Literature searches: For example, when a scholar needs to fully review the literature in a research area.
- Legal discovery: Lawyers need to uncover all past relevant cases in order to make strong arguments.

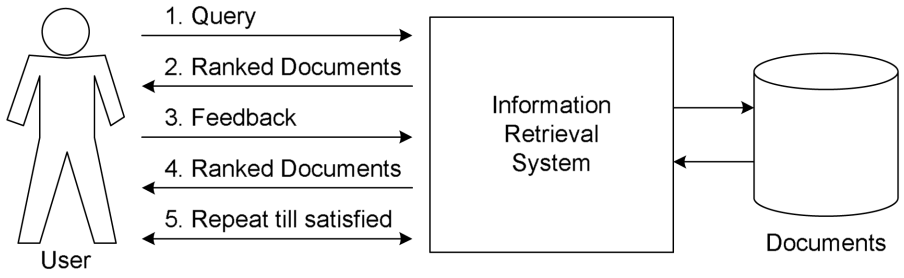


Figure 1.1. The traditional scenario for interacting with an information retrieval system to find relevant documents via relevance feedback.

- Medical conditions / treatments: When someone becomes ill, the doctor, patient, and family often need to find as much information as possible regarding the illness.

Within this broad problem of finding multiple relevant documents, we focus on one specific problem: once a user has found a relevant document, how should the user proceed to find other relevant documents? The user could find this document via a typical search or the user could already know of the document. The classic solution to this problem is what is known as *relevance feedback*.

Relevance feedback has the user provide feedback on the results to the IR system. The IR system can collect feedback in many ways, but a typical approach is multiple-item relevance feedback whereby the user judges the top 5 or 10 documents as relevant or non-relevant and submits these judgments to the IR system. The IR system uses the judgments to craft a new query and returns a new set of results. The aim is for this feedback loop to continue until the user's information need is satisfied. Figure 1.1 shows this process. Interaction techniques like relevance feedback focus on helping the user after the initial query rather than on improving the initial retrieval.

While relevance feedback is known to be a powerful technique for improving retrieval quality (Ruthven and Lalmas (2003) provide an extensive review of over 30 years of relevance feedback research), it has seen little adoption by search systems. A

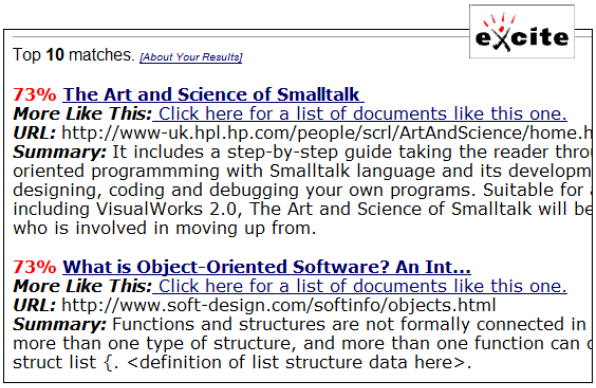


Figure 1.2. The Excite search system (circa 1996) provided a find-similar link next to each search result labeled "More Like This: Click here for a list of documents like this one."

feedback-like technique that has seen adoption is an interaction mechanism we term *find-similar*.

Find-similar allows a user to request a list of documents similar to a given document. As a user interface feature, find-similar is typically instantiated as a button or link next to each result in the list of search results. For example, the Excite search engine (circa 1996) labeled their find-similar link "More Like This: Click here for a list of documents like this one" as shown in Figure 1.2. As such, find-similar provides a way for users to navigate from one document to another and supports the search techniques commonly employed by users (Bates, 1989).

While not all people have experience using find-similar, significant evidence exists that many users utilize this interaction mechanism. Spink et al. (2000, 2001) analyzed samples of Excite's query logs and reported that between 5 and 9.7 percent of the queries came from the use of the "more like this" find-similar feature. Lin et al. (2007) have reported that for the U.S. National Library of Medicine's search engine, PubMed, 18.5% of non-trivial search sessions involve clicks on articles suggested by PubMed's find-similar, which PubMed refers to as *related articles*.

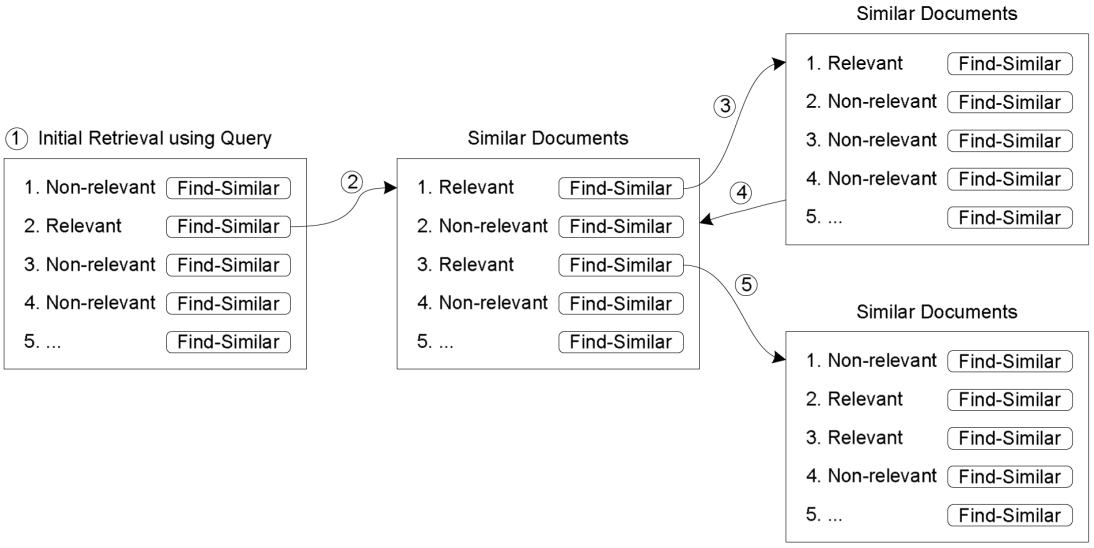


Figure 1.3. This figure shows an example of find-similar use. The user starts by entering a query and getting a ranked list of documents (1). The user examines the documents and finds the first one to be non-relevant. The second document is relevant and the user decides to apply find-similar to that document (2). The system produces a ranked list of documents that are similar to the requested document. The user continues (3) until reaching a point where the list of documents has few relevant documents and the user clicks the "back button" in the interface to go back to the previous ranked list (4). The user continues searching (5) via find-similar until finished.

This dissertation focuses on find-similar's use as a search tool. To use find-similar as a search tool, a user will apply find-similar to a relevant document to find more relevant documents, and so forth. A user can either start with an initial query and apply find-similar to individual results, or a user can start with a known relevant document found via other means and apply find-similar to that document. Figure 1.3 shows an example of find-similar use starting from an initial query.

Find-similar can provide other forms of similarity to the user besides content similarity. For example, Figure 1.4 shows a page from CiteSeer (Bollacker et al., 1998), which is a research paper repository and search system. Many of the links provided by CiteSeer can be considered find-similar links. Some of the links are to documents with similar content while others are to documents with similar citations.

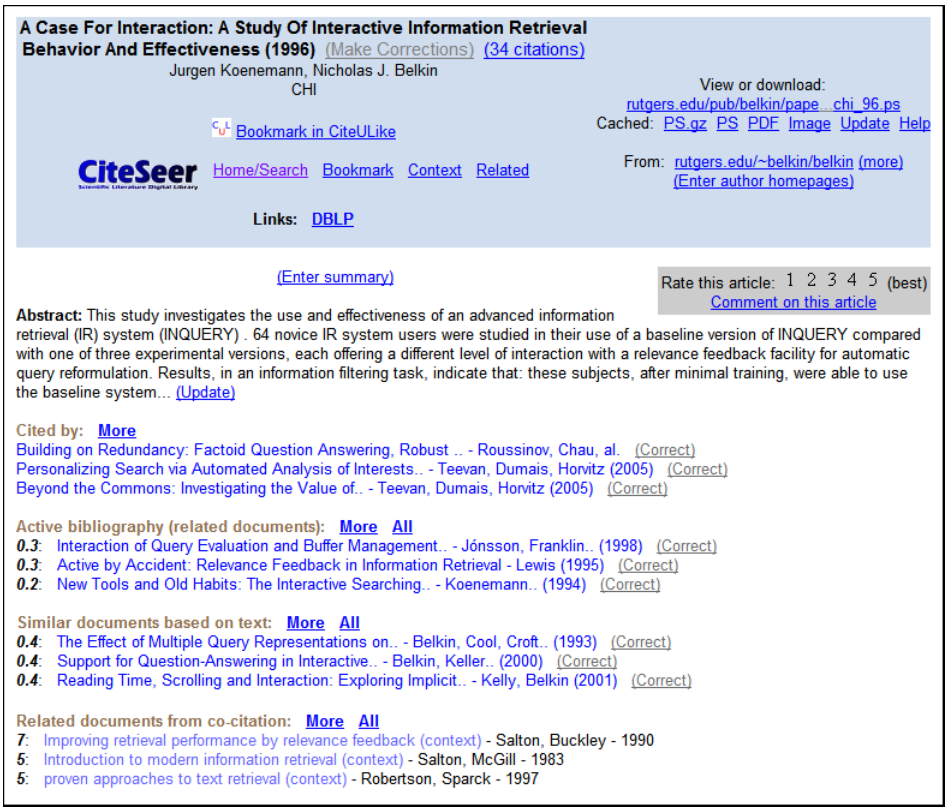


Figure 1.4. The CiteSeer research paper search system provides many types of similarity links besides content similarity, e.g. papers that cite this paper and co-citation similarity. This web page is also an example of using find-similar to find other relevant documents when a document, rather than a query is the starting point.

A similar search system, Google Scholar,[2] provides links to documents written by the author of a paper, which allows the user to navigate along another dimension of similarity.

Using find-similar to navigate via similarity has significant potential to improve retrieval quality. Figure 1.5 shows an example of the power of navigating via similarity for the TREC ad-hoc query number 334, "export controls cryptography." This example uses query likelihood to perform the initial retrieval and regular document-to-document similarity (details given in Chapter 3). For this query, the initial retrieval

[2] `http://scholar.google.com/`

pulls up three relevant documents into the top of the results at ranks 1, 2, and 4. In total, there are nine known relevant documents for query 334. The initial query finds the other relevant documents at ranks 79, 2465, 13564, 23831, 37874, and the last document is not retrieved at all since it doesn't contain any of the three query terms. In the figure, this last document is given a rank of 528,155, which is the number of documents in the collection. Without find-similar or some other mechanism, the documents at ranks 2465 and greater are effectively "out of reach" of the user. If the user uses find-similar to request similar documents for the relevant document at rank 4, the user will then find relevant documents at ranks 3, 6, 8, and 11 that were all at ranks greater than 1000 for the initial retrieval. Find-similar makes these documents much easier to reach. For example, document R5 now goes from a distance of 2465 to a distance of 7 $(4 + 3)$. Using find-similar, the hardest to reach relevant document is 46 documents away from the initial query and the remaining relevant documents are found within 15 documents. This is a dramatic improvement from only finding 3 relevant documents in the initial retrieval.

In a broad sense, find-similar aims to add links to documents such that the time for a user to get from relevant document to relevant document is minimized. We next describe the work in this dissertation that addresses how to measure and improve the performance of find-similar.

1.2 Dissertation Overview

While find-similar has existed for many years and many search systems have provided find-similar functionality to users, little research existed regarding find-similar before the work presented in this dissertation. We add to this existing research and further fill the gap between practice and research knowledge of find-similar.

The work in this dissertation is best viewed as falling into three categories: measurement, performance improvement, and applicability to different domains.

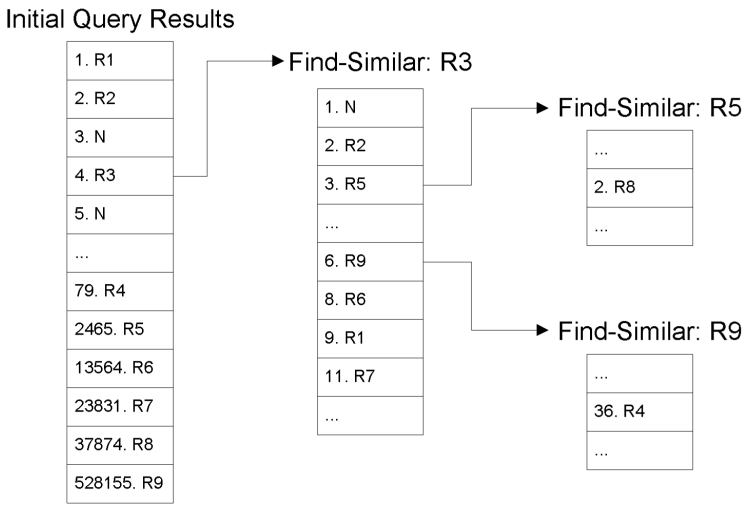


Figure 1.5. This figure shows the shortest paths via find-similar from the initial query to the nine relevant documents for TREC query 334, "export controls cryptography." Non-relevant documents are represented by an "N" or omitted to save space. The relevant document "R3" at rank 4 in the initial query results makes the majority of relevant documents much easier to reach.

We look at two ways to measure find-similar. The first, as presented in Chapter 3, performs a simulation of user behavior given an interface that incorporates find-similar. Here we measure find-similar's potential to improve retrieval quality as compared to a state-of-the-art retrieval system as well as compared to relevance feedback. In Chapter 4, we use this simulation methodology to investigate the effect of different initial conditions on find-similar's performance. In Chapter 5, we present a method that does away with the user interface and focuses its measurement on the document network formed by find-similar's document-to-document similarity measure.

To improve find-similar, we can provide user interface support for find-similar as well as change the document-to-document similarity measure. In Chapter 3, we investigate the need for an interface to help the user avoid the reexamination of documents while using find-similar. In Chapters 3, 4, 5, and 6, we look at various types of document-to-document similarity. We look at content similarity both with

and without query-biasing in Chapters 3 and 5. As part of Chapter 4, we compare our language modeling document-to-document similarity to PubMed's actual document-to-document similarity. In Chapter 6 we treat the World Wide Web's hyperlinks as a form of document-to-document similarity and measure the Web's navigability both with and without additional content similarity links.

To understand if find-similar is applicable across different domains, we investigate its performance with newswire and government documents in Chapters 3 and 5, abstracts of biomedical texts in Chapter 4, and with Web pages in Chapter 6.

We review related work in Chapter 2 and conclude the dissertation in Chapter 7.

1.3 Contributions

In this dissertation, we make the following contributions:

1. We show that find-similar has the potential to produce a 23% improvement over a non-interactive state-of-the-art baseline as measured by mean average precision. This performance matches relevance feedback. (Chapter 3)

2. By simulating simple and plausible user browsing patterns, we show that find-similar's performance is significantly affected by the browsing pattern. In particular, if carelessly used when results are already good and not in need of much improvement, find-similar can degrade these results. (Chapter 3)

3. We find that find-similar benefits from user interface support to avoid the reexamination of documents. Without support to avoid the reexamination of documents, find-similar only benefits the poorest performing topics. (Chapter 3)

4. Poor initial retrievals can come from complex information needs, the retrieval method, or novice users. As part of a case study of PubMed, we show how

find-similar compensates for poor initial retrievals. This work also broadens the applicability of find-similar beyond the newswire and government documents of Chapter 3 to biomedical abstracts. (Chapter 4)

5. We find that poorer retrieval systems are helped more by find-similar while the more difficult topics are not helped as much as the easier topics. (Chapter 4)

6. We find that find-similar's performance can be improved by using a query-biased, document-to-document content similarity rather than a similarity measure that simply uses the document as a query. (Chapters 3 and 5)

7. We create a novel and well defined method to evaluate the ability of document-to-document similarity measures to cluster relevant documents. We show that both local and global measures of clustering are needed. (Chapter 5)

8. We show that the query-biased similarity that performed better under simulation also clusters relevant documents better than a regular similarity that treats a document as a query. The query-biased similarity produces a relative gain in the global measure of clustering by 45% while also producing a relative gain of 38% in a local measure of clustering (precision at rank 5). (Chapter 5)

9. We show that to a limited extent, the cluster hypothesis is true on the web when the document-to-document similarity measure is the distance to navigate from one document to another using hyperlinks. We found that the automatic addition of content similarity hyperlinks to web pages can significantly increase the number of relevant documents reachable from a given relevant document. We quantify this increase in navigability using the method of Chapter 5 and show that find-similar produced an absolute gain in global navigability of 13.8% while at the same time increasing the local navigability of the web. (Chapter 6)

CHAPTER 2

RELATED WORK

In this chapter we review related work and describe its relation to the work in the dissertation. We start by addressing our approach to improving IR performance as well as giving more evidence for user tasks that require multiple relevant documents.

As discussed in Chapter 1, the well-known technique of relevance feedback is the classic solution to helping users find multiple relevant documents. The problem faced by relevance feedback is that it has not seen adoption by search systems. Find-similar is a feedback-like interaction mechanism that has seen adoption. While there are likely many reasons for find-similar's adoption, in Section 2.3, we discuss one possible reason: find-similar supports existing user behavior.

In Section 2.4 we review other research on find-similar or find-similar like systems. Since find-similar is a feedback-like search tool, we briefly touch on other attempts to improve and understand the issues with relevance feedback in Section 2.5.

Inherent in find-similar is that it provides a means for users to navigate from one document to another. Anytime a system provides such a navigation mechanism, we can think of the transition from document to document as a *link* between the two documents. Hypertext systems explicitly provide links between documents and most often these links are inline with the text of a document. The web is the most prevalent example of a hypertext system today. Find-similar automatically creates links between documents given find-similar's document-to-document similarity measure. In Section 2.6 we discuss automatic hypertext construction and the evaluation of hypertext.

Hypertext systems and find-similar create networks of documents. In the latter half of the dissertation, we apply network analysis measures to evaluate find-similar. We conclude this chapter in Section 2.7 with a look at related network analysis research.

2.1 A Tool-Centric View of Information Retrieval

One of our interests in studying find-similar is that it is an example of simple user interaction mechanism that has seen adoption by search systems. In this section, we describe the work of Bates (1990), who provides an alternate and detailed perspective on the automation of IR that shares much with our belief in a tool-centric view of IR.

Bates defines four levels of search activity: moves, tactics, stratagems, and strategies. Moves are basic operational steps such as entering search terms or examining a document. Tactics describe a sequence of one or more moves designed to help the user find relevant material faster. For example, a tactic could be to broaden a query to increase recall (Bates, 1979). Stratagems include the chaining behaviors of Ellis (1989), for example. Finally, strategies are broad plans for how a user will find relevant information.

Bates also defines fives levels of system automation: no system involvement, displays possible activities, executes actions on command, monitors search and recommends actions, and executes automatically. Bates claims that IR research aims to execute automatically all levels of search activity. This is similar to our notion of the AI-view of IR. Bates advocates research on ways to help the user with tactics and stratagems. Find-similar falls within Bates' recommended area of research as it helps the user execute chaining based on content, which would be a stratagem in Bates' classification.

2.2 Need for Multiple Relevant Documents

A key motivation for find-similar is that multiple relevant documents are needed by some users to satisfy their information needs. While we know that some tasks such as a literature review by default require many relevant documents, there is evidence that other less obvious tasks have similar requirements.

Bhavnani (2005) has analyzed the distribution of health care facts across relevant web pages and web sites. In general, Bhavnani found that facts are distributed such that very few pages contain many facts while many pages contain a few facts. The result of this is that users must find many relevant pages to collect all the relevant facts. In an earlier work, Bhavnani et al. (2003) estimated that users on average would have to visit 25 pages to find a complete set of 12 melanoma risk concepts.

Web search is well known to be a high precision activity for most users. In contrast to this, Rose and Levinson (2004) found that in a sample of web queries, 22.7% of all searches are undirected informational, e.g. "color blindness" or "jfk jr." and another 5.0% are searches for advice. For each of these types of searches, it is reasonable to expect users to need multiple relevant documents to satisfy their information needs.

2.3 User Behavior in Support of Find-Similar

One possible reason for find-similar's adoption by search systems is that find-similar supports existing user behavior. In this section, we review work that has either studied user behavior directly or work that has created predictive models of user behavior.

Bates (1989) presents a model of user searching behavior she named *berrypicking*. In the berrypicking model, the user's query is taken to be evolving over the search and the information that the user needs to find is scattered across many documents. The berrypicking model is in contrast to iterative relevance feedback where a single query is refined with positive and negative examples of relevant documents. While

this dissertation's experiments involve fixed sets of relevant documents, it is possible that the TREC assessors' notion of relevance shifts over the process of judging documents for relevance. Even with a fixed notions of relevance, a set of documents may contain many different relevant subtopics or aspects that are difficult for a single query representation to capture. Find-similar in contrast to relevance feedback is an appropriate tool to support a berrypicking search model. Find-similar only uses the current document to determine the other most similar documents and thus allows users to possibly visit different aspects of a topic with greater ease than does relevance feedback.

Ellis (1989) in a study of social scientists found that one of their primary search strategies was what Ellis termed *chaining*. Ellis broke chaining activities into forward and backward chaining. Forward chaining involves using a citation index to determine documents that have cited the current document. Backward chaining involves following the citations in the current document's bibliography. While find-similar as studied in this dissertation does not explore citations as a notion of similarity, find-similar does support forward and backward chaining based on the content of a document. Meho and Tibbo (2003) provide an updated review of Ellis' work and further refine the information seeking behaviors Ellis identified. Computer interfaces have been specifically designed to support the navigation of citations (Mackinlay et al., 1995).

There is other evidence that navigating from document to document is an important part of the search process for users. Teevan et al. (2004) described a search technique whereby users take small steps towards their goal without specifying their entire information need at the beginning of the search. They named this search technique *orienteering*. For example, Teevan et al. described the behavior of a user searching for a professor's phone number. Rather than try to use a search engine to find the information, the user navigated to the professor's departmental homepage to find a list of faculty. Teevan et al. found that orienteering seemed to be cogni-

tively easier for users compared to attempting to fully describe their search target as a query. Orienteering also provided users with a sense of location and gave them a context to better understand the answers they found. Cutrell et al. (2006) reported that users searching their personal information collections particularly liked a feature that allowed them to take a single search result and use it to modify their query.

Pirolli (1997) has proposed a model of search behavior called *information foraging*. The model basically posits that as humans evolved, we became very skilled in foraging for food and other resources. When foraging, one wants to maximize total gain over time even though resources are not found uniformly in nature but instead are often found in clumps. Given the distribution of resources, foraging theory addresses when should an individual stay and consume the current patch of resources and when should an individual begin searching for the next patch of resources. Information foraging models have been shown to effectively model user behavior in many information seeking environments (Pirolli and Card, 1998, 1999; Pirolli, 2007). While we do not explicitly utilize information foraging theory, the breadth-like browsing pattern introduced in Chapter 3 shares key attributes with information foraging theory. Namely, the breadth-like browser delays exploration until the results' quality degrades beyond a certain point and the browser always selects links given the highest *information scent* (they are links to relevant documents).

2.3.1 Relevance Feedback as Find-Similar

It appears that some users attempt to use relevance feedback systems designed for judgments on multiple documents in a manner resembling find-similar. Croft reports that users will often use a single document, which may be unrelated to the query, for relevance feedback and effectively be "browsing using feedback" (Croft, 1995). Hancock-Beaulieu et al. studied 58 user sessions that used interactive query expansion (IQE) via a relevance feedback interface (Hancock-Beaulieu et al., 1995). Of the

58 sessions, 17 used only a single document for feedback. Algorithms designed for multiple-document feedback may not always work well for single document feedback. Only 3 of the 17 sessions were successful at finding additional relevant material. Hancock-Beaulieu et al. hypothesized that IQE with too few relevant items may not work correctly. Similarly, Harper and Kelly (2006) created an IR system that allowed users to place documents into virtual piles. For each pile, the user could ask the IR system to produce a list of documents similar to the documents in the pile. As part of a user study, Harper and Kelly found that over 80% the similarity searches involved piles with 3 or fewer documents. From Figure 4 of Harper and Kelly's paper, we estimate that over 50% of the similarity searches involved piles containing a single document. While Harper and Kelly did not find that users performed better with the piles, they did find that users preferred the piles system.

2.4 Other Studies of Find-Similar or Find-Similar Like Tools

In this section we review work that has examined find-similar or has studied tools very closely related to find-similar. All of these systems provide a means for the user to go from an existing document to other similar documents.

The I^{3}R system worked by a "quality-in quality-out" principle that strove to help users build queries that more accurately reflected their information needs (Croft and Thompson, 1987; Croft et al., 1989; Thompson and Croft, 1989). I^{3}R was a rich IR system that had as one of its many components a *browsing expert* that allowed a user to browse by following links from a document, author, or index term. I^{3}R supported many different types of similarity. For example, the nearest neighbors of a document could be based on citations or content or other information from a knowledge base. Two aspects of I^{3}R's browsing expert are significantly different than our study of find-similar. First, I^{3}R acquires knowledge as the user browses and can use this knowledge to update the query or suggest a search strategy. Our version of

find-similar makes no attempt to utilize knowledge about seen documents. A second difference is that I^{3}R makes browsing suggestions to the user. For us, find-similar merely returns a ranked list of similar documents with no recommendation of which documents may be relevant or not to the user. Another more minor difference is that I^{3}R the system draws a display showing nodes and links indicating browsing choices. One of the functions of the display is as a map to prevent users from becoming lost in their browsing. Such a display may be of use to interfaces incorporating find-similar, but find-similar does not require such a display. Find-similar offers a subset of the richer browsing capabilities of I^{3}R, and our study of find-similar can be seen as giving support for the utility of the browsing capabilities in I^{3}R.

Wilbur and Coffee (1994) studied several aspects of find-similar and their research is the most similar to our work in Chapter 3. They found that on average, a single relevant document used as a query does not perform as well as the original query, but that relevant documents similar to the query will do better than the query. They also used a set of browsing patterns and found that a method they called *parallel neighborhood searching* performed better than the other patterns. This method attempts to search the find-similar lists of all discovered relevant documents to the same depth. This browsing pattern is likely too complex for a user to follow. They suggested that a system could hide the complexity by showing the user one document at a time to judge, but such a system no longer supports similarity browsing or traditional lists of results.

Melucci (1999) provides another evaluation of find-similar. Melucci's evaluation sees find-similar as a subpart of a larger hypertext retrieval system. The evaluation does not make mention of hypothetical interfaces as we do in Chapter 3, but like us and Wilbur and Coffee (1994), Melucci does investigate different browsing patterns, which he calls search tactics. Melucci's search tactics address comparisons to relevance feedback and the combination of find-similar and the initially retrieved list. Unlike

our work, Melucci appears to only perform a single iteration of relevance feedback and appears to only apply find-similar to a single document. In the find-similar search tactic employed by Melucci, if no relevant documents are found in the top ranked documents, find-similar may be applied to a non-relevant document. Melucci does not make mention of handling the reexamination of documents problem, which we address in Chapter 3. Melucci finds that browsing via his find-similar implementation was worse than relevance feedback, which is not surprising since his implementation only applies find-similar once. Melucci's results do suggest that find-similar may be of help to the user when the initial results are poor, which is something we confirm in Chapters 3 and 4.

As described in Chapter 1, the Excite search engine at one time contained a feature that allowed a user to click on a link that read "More Like This: Click here for a list of documents like this one." for each result in the ranked list. Spink et al. (2000, 2001) have analyzed samples of Excite's query logs and reported that between 5 and 9.7 percent of the queries came from the use of the "more like this" find-similar feature. There is little evidence that users repeatedly used the find-similar feature to browse by similarity. Even though some web users may need multiple relevant documents (see Section 2.2), most web users are precision oriented and thus it is not surprising that find-similar found limited use by Excite web users.

Campbell (2000) investigated a system that allowed users to search for relevant objects by browsing to similar objects. Campbell described his system as an *ostensive* browsing system. Campbell's system was unique in that the path a user took to reach an object influenced what objects were considered to be similar to that object. Campbell found some evidence that the *document in context* (DIC) model performed better than other models. The DIC model weights the more recently examined objects higher than objects further back on the user's exploratory path. In our work, find-similar allows ostensive browsing and implements the DIC model in the extreme.

Only the immediate document and possibly the user's query is used for determining the similar documents. Campbell's system also utilized a display to show the path taken by the user through the similarity space (cf. I^3R).

Strasberg et al. (2000) investigated different document-to-document similarity measures for biomedical articles. They built a collection of 186 articles. From these 186 articles, one was chosen at random as a "seed" article. Then 90 articles were chosen at random, and for each of these articles, an expert judged if the article would be one he would like to see in a list of "related articles." The various similarity measures were judged based on their ability to rank the 90 judged articles. They found little difference among their similarity measures although techniques using only title words or MESH terms performed worse at higher levels of recall. In a second experiment, they had an expert collect 20 related documents. They then selected 80 unrelated documents. Performance was measured for each of the 20 documents to rank the other 19 documents highly. This type of performance measurement is similar to the precision at rank 5 (P5) measure of the cluster hypothesis (Voorhees, 1985) but ignores the importance of a global measure of the cluster hypothesis as we discuss in Chapter 5. On this second collection, they found that tf-idf vector space ranking outperformed a simpler binary weighted Dice coefficient of similarity. Our studies involve much large document collections (on the order of 500,000 documents or more) and large numbers of user search topics (50 to 150).

Haveliwala et al. (2002) studied find-similar on the web. Their work looked at measuring the quality of document-to-document similarity measures, the development of better performing similarity measures, and the fast calculation of the similar documents. They first developed a methodology using the Open Directory Project (ODP)[1] hierarchy of websites to measure the quality of their document-to-document

[1] `http://www.dmoz.org/`

similarity methods. While all of our measures of document-to-document similarity involve either the improvement of search effectiveness or looking at the resulting navigability of document networks where navigability is concerned with getting from one relevant document to other relevant documents, Haveliwala et al. use the ODP as a surrogate or oracle of what is similar to what. Menczer (2005) created a similar method to determine what he calls the *semantic similarity* of two web pages. They also investigated several document-to-document similarity methods. They found using anchor text (the text surrounding links to a given page) very useful and weighting the text based on distance from the anchor the best. Our investigations focus on the value of query-biasing the similarity method as well as on comparing content similarity to existing web hyperlinks. To compute similarity quickly, they used very simple term weighting with the Jaccard coefficient and a hashing mechanism.

Several approaches to find-similar utilize the link structure of the web to determine similar pages (Dean and Henzinger, 1999a,b; Jeh and Widom, 2002; Thelwall and Wilkinson, 2004; Lin et al., 2006). These approaches are related to the link prediction problem in social networks (Liben-Nowell and Kleinberg, 2003). Dean and Henzinger (1999b) evaluated their methods by asking users to judge the usefulness of the pages found similar to a given page. This notion of usefulness is separate from any given search task that a user may have. Our focus in this dissertation is on helping users find relevant documents given an information need. Additionally, many of the pages used by Dean and Henzinger were top-level or homepages of large websites, e.g. `www.ebay.com`, `www.babynames.com`, and `www.rei-outlet.com`. Searching for similar websites will tend to be a different task than finding other relevant content, which is our focus.

Takaki et al. (2004) use a patent application document as a query to find existing patents for the purpose of being able to invalidate the patent application. In an attempt to improve the document-to-document similarity measure, they break up the document into subtopics rather than use the whole document. For their application,

the subtopics were the various sections of the patent application. By down-weighting the preamble of the application, they were able to improve performance. While like find-similar, in their study there is no notion of navigating from one relevant document to another. The patent application is the only query issued and the resulting ranked list of documents is judged for quality.

Perhaps the best known form of find-similar is Amazon.com's item-to-item collaborative filtering (Linden et al., 2003). In this application, users can navigate by a similarity that has been computed based on the sets of items purchased by users and other data. Linden et al. (2003) do not report any evaluation of Amazon.com's ability to help users find relevant content.

Some systems provide a list of similar documents based on what the user is currently viewing or doing (Rhodes and Starner, 1996; Dumais et al., 2004). These systems are effectively versions of find-similar, for if a user selects a document from the list of similar documents, the user's context will shift to that new document and the list of similar documents will change automatically.

While Campbell (2000) looked at annotated images, Birbeck et al. (2006) and Joho et al. (2007) have continued in the ostensive framework and have studied an interface that allows ostensive browsing of the top ranking sentences from a query's top 30 results. The interface allows the browsing to be conducted by hovering the mouse pointer over a set of sentences and this then pulls up a cascaded menu of three more top ranking sentences similar to the selection under the mouse. This is an innovative interaction technique similar to the technique of fluid links of Zellweger et al. (1998). They found evidence that such a system would benefit the user searching for relevant documents. Our work differs from their work in that we allow the user to find similar documents across the entire collection, which is necessary for increased recall. We do not restrict the user to the top ranked documents.

Lin et al. (2007) took a preliminary look at the properties of the document network formed by the find-similar feature of PubMed and analyzed logs of PubMed's find-similar feature. They took 46 topics from the 2005 Genomics track and looked at the precision at 5. They found that topics that had more relevant documents tended to have higher precision at 5. One variant of the related articles feature in PubMed shows 5 articles in a panel on the same page as a currently viewed document. Another analysis they performed was to take the network formed by looking at the relevant documents and the 5 related articles and measure how many connected components existed in this network. They again found that as the number of relevant documents increased so did the number of components in a linear fashion. From their preliminary investigation, they saw no reason to conclude that the related articles feature should not be helpful.

In the next piece of analysis, they looked at a week of PubMed's log files. They first eliminated sessions of only a single page view since those are presumed to be bots and systems that provide direct access to MEDLINE. They ended up with 1.9M sessions. 18.5% of these sessions also had a click on a link to a related article. Ignoring short sessions, they found that approximately 5% of the page views were a result of clicks on related articles. They also found that users' most likely next action after using the related articles feature is to use the related articles feature again.

Huggett and Lanir (2007) conducted a user study in which users found more relevant documents using an interface that provided find-similar over an interface without find-similar. Huggett and Lanir's study used small newswire collections of 2000 documents and limited test subjects to two minutes for each search. In this dissertation, we use much larger document collections and utilize batch experiments.

2.4.1 Other Approaches Related to Find-Similar

Lieberman (1995) created a software agent named Letizia that watches the user browse the web and tries to infer the user's information need. Letizia actively crawls the web from the user's current location. When the user does not know where next to browse, the user can ask Letizia for suggestions. Olston and Chi (2003) created ScentTrails to bridge the gap between searching and browsing. ScentTrails highlights links on a web page that lead to pages containing a user's search terms. This highlighting allows the user to combine the visible browsing cues on the current page and the search information hidden on the following pages. In a user study, subjects completed tasks faster using ScentTrails than with browsing or searching alone.

Another set of research has focused on helping the user better process ranked retrieval results. This work is related to but different from relevance feedback and find-similar, both of which are applied to the entire collection of documents and not restricted to the set of top ranked results. For example, Leuski (2000) created a software agent to guide users in their exploration of the top results. Other approaches involve presenting the results grouped by an online clustering of the results or by predetermined categories (Hearst and Pedersen, 1996; Eguchi, 1999; Iwayama, 2000; Chen and Dumais, 2000). These approaches are different from find-similar in that while the user gets to see documents grouped by similarity, the user does not get to request more documents similar to a document.

2.5 Making Relevance Feedback Work

One possible reason for search systems' adoption of find-similar is its appearance as an easily understood and simple to use form of relevance feedback. Of the large body of relevance feedback research (Ruthven and Lalmas, 2003), Aalbersberg's *incremental feedback* is an illustrative example of simplifying relevance feedback (Aalbersberg, 1992). With incremental feedback, the user is shown one result at a time

(cf. Wilbur and Coffee). To see more results, the user must judge the relevance of the presented item. In batch experiments, Aalbersberg found that incremental feedback worked better than Rocchio, Ide Regular, and Ide Dec-Hi. For these other approaches, Aalbersberg used an iteration size of 15 documents. Incremental feedback improved a 3-point[2] average precision by an average of 32% across four test collections while the other methods improved by 18% over retrieval without feedback. Aalbersberg found that as the iteration size decreased to the limit of 1 (incremental feedback), the performance of Ide Regular improved. While incremental feedback builds a model of relevant documents one document at a time, each use of find-similar involves a single document without any accumulation of documents or model of relevance.

Designing good interface support for relevance feedback takes a considerable amount of work (Belkin et al., 2001). Koenemann and Belkin (1996) studied the use of relevance feedback for a filtering task. Users were most successful using relevance feedback when the feedback interface was *penetrable* as compared to *transparent* or *opaque*. The penetrable interface showed the user what terms were added to a query based on feedback and also allowed the user to add or subtract terms from those suggested via relevance feedback. The transparent interface showed the terms but did not allow them to be directly manipulated. The opaque version completely hid how the relevance feedback mechanism worked. Of note is that users much preferred the penetrable interface. Users of the opaque interface wanted to "see and control" the process. Find-similar leans towards giving the user direct control although our use of query-biased similarity is opaque.

Beaulieu (1997) describes a VT100 character-based interface and two graphical users interfaces (GUIs) for Okapi. The VT100 interface supported automatic query expansion. Similar to Koenemann and Belkin, Beaulieu found that for the VT100

[2]For a 3-point average, the standard recall points are 25%, 50%, and 75%.

interface, users would have preferred to know what terms were added to their queries. Without knowing the added terms, users had to restart a search session to redirect the search. The first GUI created supported interactive query expansion and allowed users to select terms suggested by the relevance feedback component for addition to their queries. As opposed to Koenemann and Belkin's results, take-up of relevance feedback and its effectiveness dropped from 31% and 50% respectively to 11% and 31%. With a redesign, a new GUI raised the take-up and effectiveness to 21% and 56%. The design of an interface can greatly affect the utility of a retrieval feature.

2.5.1 Implicit Relevance Feedback

Implicit feedback describes systems that attempt to watch the user and infer what the user is interested in without obtaining explicit relevance judgments from the user as relevance feedback would traditionally do. A goal of implicit feedback is to obtain retrieval quality improvements from relevance feedback without the user's need to do anything except search for documents. Kelly and Teevan (2003) provide an overview and bibliography of implicit feedback research. As studied here, find-similar has no component of implicit relevance feedback. In all variants that we concern ourselves with, find-similar simply provides a list of documents similar to a given document. Previously viewed documents, and documents that have previously been the query document for find-similar, do not change the list of similar documents that find-similar produces.

2.5.2 Poison Pills

A series of relevance feedback experiments were conducted during the Reliable Information Access (RIA) workshop organized by NIST in 2003. One of the striking results was that some relevant documents hurt relevance feedback when added to the set of documents judged relevant by the user (Warren and Liu, 2004). A further analysis identified that for TREC 6, 7, and 8 ad-hoc retrieval, 5.3% of the relevant

documents perform poorly for single document feedback; Terra and Warren (2005) called these document "poison pills." Find-similar can be viewed as a way of allowing the user to easily correct for mistakes made by the retrieval system with single document feedback. If the resulting list is poor, the user simply returns to the previous list and continues the search. With more traditional multi-document relevance feedback, the user would have to be very careful to notice that retrieval results have suffered because of the addition of a relevant document to the set of judged relevant documents.

2.6 Automatic Hypertext

In many systems, users can browse documents via hyperlinks. If a collection lacks hyperlinks, they can be automatically generated (Allan, 1997). Find-similar effectively adds a hyperlink from a document to those most similar to it. For hypertext systems like the web, researchers have created programs to assist the user with finding relevant pages via browsing (Lieberman, 1995; Olston and Chi, 2003). In contrast to these approaches, find-similar does not observe the user to determine what the user considers relevant, and find-similar does not offer any assistance in choosing where to browse. Wilkinson and Smeaton (1999) provides a nice short survey of automatic hyperlink creation research.

Bodner et al. (2001) provide an overview of a large body of their work that looks at various ways to augment text retrieval systems with automatic hyperlinks and other browsing interaction mechanisms. A significant finding from this research was that hypertext enabled IR systems tended to increase the recall performance of the IR system for users. In general they found that the increased recall came as a result of users viewing more documents in the hypertext enabled systems. The details of their work most relevant to find-similar can be found the papers of Golovchinsky and Chignell (1993); Golovchinsky (1997); Bodner and Chignell (1998, 1999).

2.6.1 Evaluation of Hypertext Quality

Botafogo et al. (1992) detail a property of hypertext they term *compactness* as a measure of the quality of a hypertext. Compactness is basically a normalized all pairs shortest paths measure, much like the efficiency measure of Latora and Marchiori (2001) that we use in Chapter 5 to measure the global navigability of a document network. In contrast to Latora and Marchiori's measure, the compactness measure requires a maximum possible distance between nodes in the network for the normalization. This requirement makes compactness unable to handle hypertext networks that have documents with no path between them (an infinite distance). In this dissertation, we concern ourselves with the distribution of relevant documents and the ease of navigating between them. Botafogo et al.'s work does not consider relevance.

Blustein et al. (1997) present methods for evaluating the quality of hypertext links. They construct hyperlinks via tf-idf measures of similarity. They only place links between the top 1, 2, ..., 5 most similar documents. Then they evaluate the shortest path distance between nodes. Whereas they make the link weight 1 for all links, we set the link weight equal to the rank of the target document in the ranked list of similar documents as an approximation of the cost for a user to find and traverse a link. They then measure the correlation between the distance between documents on the hypertext graph and the tf-idf measure. Of note, they only compute the correlation for documents that have a path between them in the hypertext.

2.7 Related Network Analysis Work

In Chapter 5, our focus shifts to measuring the document networks formed by different document-to-document similarity measures. Newman (2003) provides a review of the large body of research on networks and various graph theoretic measures of

network properties. Costa et al. (2007) provides a survey of many of these network measures.

Chapter 5 discusses the well-known cluster hypothesis, which essentially says that relevant documents will tend to be more similar to each other than to non-relevant documents. Given a document-to-document similarity measure, we can view the similarity relationships between documents in a graph theoretic sense. Documents are nodes and links are placed between nodes based on the given similarity measure. IR similarity measures tend to produce a numeric measure of similarity and some measures are asymmetric. Given this nature of IR similarity measures, an obvious document network for IR would be a directed network with weights for each link between nodes.

Given the hypothesis' name, the *cluster* hypothesis, the *clustering coefficient* network measures appear to be obvious choices for measuring the cluster hypothesis. These measures exist to characterize the extent to which the nodes in a network are clustered. Many of these measures are designed only for undirected, unweighted networks. Cluster measures typically average over all nodes in the network a measurement that captures the extent to which a node's neighbors are themselves neighbors. In the world of undirected, unweighted networks, one can see that these cluster measures may say a network is perfectly clustered even if some nodes cannot be reached from other nodes. In other words, a network can be clustered in groups of fully connected subgraphs and be perfectly clustered. Even when these measures are extended to weighted networks (but not to directed networks) (Barthélemya et al., 2005; Kalna and Higham, 2006), these measures still aim to capture the same characteristic of networks, which is different from the notion of clustering in the cluster hypothesis.

The cluster hypothesis says that for various subsets of the graph, where each subset is a set of relevant documents, that the nodes in these subsets are *closer* to each other than to the other nodes in the network. As we will explain in Chapter 5, we need

a measure that captures a sense of the distance between these relevant documents as well as a measure that captures the local or neighborhood quality of a relevant document.

In a similar fashion to our analysis in Chapter 5, Lawrie (2003) performed a shortest-paths and reachability analysis of hierarchical summaries of search results to compare the quality of various summarization methods.

Since users of find-similar navigate a document network formed by find-similar's document-to-document similarity measure, this dissertation would seem to hold many connections to work that has looked at the navigation of social networks. Perhaps the most famous of the social network navigation works is that of Travers and Milgram (1969) that measured the number of people needed to reach a known target person via personal "first name" relationships only. An important difference between this work and ours is that for the IR tasks we concern ourselves with, our users do not know a priori their destination. Travers and Milgram (1969) gave participants a description of the target that included "his name, address, occupation and place of employment," ... "his college and year of graduation, his military service dates, and his wife's maiden name and hometown." In network terms, participants had to forward the research packet (message) to a neighboring node where the network was people and the links represented knowing someone on a first-name basis. Participants picked a neighboring node that they thought would most likely get the message to the target. When we know the destination in IR, we call this *known item search*, which we do not examine in this dissertation. We and our retrieval system users do not know what the set of relevant documents is for a given information need until the documents have been found as part of search. To design our IR systems, we collect known sets of relevant documents and measure the quality of IR system using these test sets, but in operation these sets are unknown.

Another way in which social network navigation research differs from our work is that we are interested in constructing inherently navigable networks rather than in only studying navigation on an existing network. We want to create similarity measures and interfaces that combined make the cluster hypothesis true. If a user has found a relevant document, we want it to be trivial for the user to find the other relevant documents. We do not create algorithms to guide the user's navigation of a network as has been done for other networks (Şimşek and Jensen, 2005). If we knew how best to guide the user through the network, we would utilize this information to improve the initial ranking of documents for the user. For our work in Chapter 3, we assume that the state-of-the-art baseline retrieval system has already exploited all reasonable information in its ranking of documents.

CHAPTER 3

USING SIMULATION TO EVALUATE THE POTENTIAL OF FIND-SIMILAR

In this chapter, we investigate the potential of find-similar to improve retrieval quality. Our investigation utilizes hypothetical user interfaces and a simulation of user behavior over these interfaces. We vary find-similar by examining two slightly different hypothetical user interfaces, two simulated user browsing patterns, and two methods for defining document-to-document similarity. We compare find-similar to a traditionally styled relevance feedback system and to a state-of-the-art baseline non-interactive retrieval.

Find-similar with a query-biased similarity, an interface that helps the user avoid reexamining documents, and a breadth-like browsing pattern achieved a 23% increase in the arithmetic mean average precision and a 66% increase in the geometric mean average precision over our baseline retrieval. This performance matched that of the traditionally styled iterative relevance feedback technique.

3.1 Introduction

At a very high level, information retrieval (IR) is about satisfying a user's information needs. IR systems help people better find, understand, and organize information. We cast the broad problem of satisfying a user's information needs, into the narrower but still difficult problem of improving the order in which a user examines documents while using a specific retrieval system. Only some documents are relevant to a user's information needs and our goal is to help the user find the relevant documents while minimizing the number of non-relevant documents that are examined.

The Cranfield experiments (Cleverdon, 1967) established the dominant IR experimental paradigm. The Cranfield methodology involves the creation of a test collection, which consists of:

- A set of documents. The documents can be anything but are typically newswire, government documents, or web pages.
- A set of topics, queries, or other statements of users' information needs.
- For each topic, a set of relevance judgments. These judgments, typically made by the user who created the topic, record whether the user considers a document to be relevant or non-relevant.

More recently, the Text Retrieval Conference (TREC) (Voorhees and Harman, 2005) has come to epitomize the Cranfield style of IR evaluation. TREC is organized around many different tracks. Each track looks at different retrieval tasks. Like the Cranfield experiments, the *ad-hoc* track studied the ability of systems to rank documents in response to a wide variety of topics. All of the experiments in this dissertation use TREC test collections.

A TREC topic typically consists of a title, description, and narrative. Figure 3.1 shows an example of a typical TREC topic for the TREC 6 ad-hoc track. The title is a few word summary of the topic. The description is often a one or two sentences describing the information need. The narrative is a paragraph length explanation that often notes specifics of what will and will not be considered relevant by the user. In this dissertation, we use the title of a topic as representative the short keyword queries typically entered by users of search engines (Spink et al., 2001).

A retrieval system must convert the user's topic into a representation suitable for retrieval by the system. Then the retrieval system ranks document from most likely to be relevant to least likely. This ranked list is then scored by various metrics based on the known relevance judgments.

TREC Topic 334

Title: Export Controls Cryptography

Description: Determine the usefulness and effectiveness of continuing to maintain export controls on encryption software.

Narrative: Relevant documents will argue for or against continuing to make encryption software subject to export controls. Since 1993 quality encryption software to ensure the secrecy of communications has been available, but the U.S. Government has considered such software to be subject to the same export controls as munitions, and has sought to restrict the export of encryption software unless it contains a device which could allow the U.S. to read the underlying messages. Business interests say that this will make it impossible for U.S. producers to compete in the international market.

Figure 3.1. Example topic.

3.1.1 Information Retrieval Metrics

Many metrics exist to evaluate the quality of ranked retrievals, but we will focus our discussion on three: precision, recall, and average precision.

Precision at rank N is the fraction of documents that are relevant in the top N documents of a ranked retrieval:

$$Prec(N) = \frac{1}{N} \sum_{i=1}^{N} Relevant(D_i) \tag{3.1}$$

where D_i is the document at rank i and the binary function *Relevant* has a value of 1 when D_i is relevant and 0 when D_i is non-relevant.

Recall at rank N is the fraction of all known relevant documents seen at rank N. In other words, if there are known to be 100 relevant documents and at rank 20 we've seen 4 relevant documents, the recall at rank 20 is $4/100 = 0.04$.

We compute the *average precision* (AP) by averaging the precision at rank N given all the ranks at which relevant documents occur.

$$AP = \frac{1}{|R|} \sum_{d \in R} Prec(Rank(d)) \tag{3.2}$$

where R is the set of relevant documents, $Rank$ returns the rank of document d, and $Prec$ is as given in Equation 3.1.

Average precision captures both a notion of precision and recall. A high average precision is not obtainable unless most relevant documents are retrieved. Unretrieved relevant documents contribute a precision of 0 to the average precision.

3.1.2 Cranfield-Style Evaluation and Interactive IR

A huge advantage of the Cranfield-style of IR evaluation is that it allows for rapid, low-cost evaluation of different ranking methods. To obtain statistically significant measurements with user studies would be time consuming and cost-prohibitive given the added noise of human subjects experimentation.

At first glance, the Cranfield-style of IR evaluation would appear to be unable to accommodate interactive IR systems such as an IR system that makes find-similar available to the user. In fact, Cranfield-style evaluation can be seen as a form of automated usability testing (Ivory and Hearst, 2001) that simulates user behavior given a hypothetical user interface.

A Cranfield-style evaluation presupposes a hypothetical user interface consisting of some sort of search box that accepts a query and when submitted by the user produces a ranked listing of results. This hypothetical interface is in abstract form the predominant interface used today by IR systems.

The simulated user behavior results from a simple model of the user. The Cranfield "user model" has the simulated user enter a query and then has the simulated user examine each search result in rank order at some constant rate. The data that results from this simulation is the order in which the simulated user examines documents. To evaluate the quality of a system, IR metrics such as average precision are applied to the ordered list of examined documents.

Seen as automated usability testing, the Cranfield-style of IR evaluation can be applied to interactive IR systems with the definition of two components:

1. Hypothetical user interface.

2. User model / Simulated user behavior.

The hypothetical user interface defines the operations that the simulated user can take. The user model determines which operations to take given the current state of the user interface.

The simulated user examines documents in some given order. As these documents are examined, they are added to a ranked list. This ranked list can be evaluated in the same manner as the traditional ranked lists of a Cranfield-style evaluation.

We are not the first to evaluate interactive IR systems in this manner. Simulation of user behavior has been part of IR research since at least the work of Oddy (1977) who used, like us, prerecorded relevance judgments to simulate users of an IR system. Our evaluation methodology directly follows the example set by Aalbersberg (1992), who defined a hypothetical user interface for his *incremental feedback* and then simulated user behavior with this interface. The order in which the simulated user examined documents determined the order of the documents in the ranked lists used for evaluation. Dunlop (1997) presented an evaluation of various interfaces and advocated the measurement of time to find relevant documents. More recently, White et al. (2004) have argued that simulation studies can be used to find better ways to implement the algorithms behind interface features before investing in user studies. Other discussions of our simulation methodology can be found in Smucker and Allan (2006) and Lin and Smucker (2008).

To evaluate find-similar, we will define two hypothetical user interfaces and two models of user behavior as well as investigate two measures of document-to-document

similarity. To compare find-similar to relevance feedback, we also define a hypothetical user interface for relevance feedback and simulate user behavior over this interface.

3.2 Methods and materials

We first give an overview of our experimental design. Next we describe how we retrieved documents for find-similar, the baseline, and an implementation of iterative relevance feedback. We then explain how we created a query model for a document to which a user has applied find-similar. We then discuss our hypothetical user interfaces and two models of user behavior (browsing patterns) used for evaluation of find-similar. We finish by describing the test collection and the evaluation methodology.

3.2.1 Experimental Design

Our basic experiment design involves the test of two treatments to a baseline retrieval. The first treatment is find-similar and the second treatment is relevance feedback. For our investigation of find-similar, we utilize a factorial design to understand the impact of three variables on find-similar's performance. The three variables are the reexamination of documents, the browsing pattern, and document-to-document similarity. We examine 2 settings for each of these variables. Table 3.1 shows the $2 \times 2 \times 2$ factorial design for the find-similar runs. As is common for IR experiments, our experiment blocks are the set of 150 topics from our test collection.

3.2.2 Retrieval methods

We used both the language modeling approach to information retrieval (Ponte and Croft, 1998) and its combination with the inference network approach (Metzler and Croft, 2004) as implemented in the Lemur (Lemur, 2003) and Indri (Strohman et al., 2005) retrieval systems.

Language modeling represents documents and queries as probabilistic models. We used multinomials as our probabilistic models of text. A multinomial has a probability

Browsing Pattern	Reex. Docs.	Similarity
greedy	allow	regular
greedy	allow	query-biased
greedy	avoid	regular
greedy	avoid	query-biased
breadth-like	allow	regular
breadth-like	allow	query-biased
breadth-like	avoid	regular
breadth-like	avoid	query-biased

Table 3.1. Factorial design for find-similar runs. "Reex. Docs." stands for "reexamination of documents" (see Section 3.2.4). Not shown are the 150 experimental blocks (150 topics).

for each word in the collection and these probabilities sum to 1. For a given piece of text T, we write the probability of the word w given the model M_T of the text as $P(w|M_T)$.

The maximum likelihood estimated (MLE) model of text estimates the probability of a word as the count of that word divided by the total number of words in the text. As such, the probability of a word w given a text T is: $P(w|M_T) = T(w)/|T|$, where $T(w)$ is the count of word w in the text T and $|T| = \sum_w T(w)$ is the text's length.

For find-similar, we ranked documents using the Kullback-Leibler divergence of the query model M_Q with the document model M_D:

$$D_{KL}(M_Q||M_D) = \sum_w P(w|M_Q) \log \frac{P(w|M_Q)}{P(w|M_D)} \tag{3.3}$$

where $0 \log 0 = 0$, and the query model is a model of the document to which find-similar is being applied. We detail the two ways we constructed query models for find-similar in section 3.2.3.

To avoid zero probabilities and better estimate the document models, we calculated the document models using Dirichlet prior smoothing (Zhai and Lafferty, 2001):

$$P(w|M_D) = \frac{D(w) + mP(w|C)}{|D| + m}$$

where $P(w|C)$ is the MLE model of the collection, and m is the Dirichlet prior smoothing parameter.

The inference network approach by Metzler and Croft (2004) takes the probability estimates from language modeling and uses them as part of the Bayesian inference network model of Turtle and Croft (1991). The inference network provides a formal method for combination of evidence, and is easily accessed by users via a structured query language.

For our baseline, we used Metzler et al.'s method (2005) that combines Metzler and Croft's (2005) dependence models with Lavrenko and Croft's (2001) relevance models. This method can be seen as using a precision enhancing retrieval method (dependence models) with a pseudo-relevance feedback technique (relevance models). Unlike Metzler et al., we used only the existing collection for query expansion with relevance models and did not use any external collections for expansion.

The dependence model uses the Indri query language to combine three types of evidence. The first is the standard bag-of-words unigram model as used by language modeling. The second type captures the sequential ordering of the terms in the query. The third uses the close proximity of query terms as evidence. Figure 3.2 shows the Indri query produced by Metzler and Croft's dependence models for TREC topic 301, "international organized crime."

To perform the baseline retrieval, first the dependence model Q of the query is run. Then a relevance model is created from the top k ranked documents. The relevance model M_R is calculated as:

$$P(w|M_R) = \sum_{i=1}^{k} P(D_i|Q)P(w|D_i)$$

where $P(D_i|Q) = P(Q|D_i)/\sum_{j=1}^{k} P(Q|D_j)$, and $P(Q|D_i)$ is the Indri belief that document model D_i is relevant to the query Q. Finally, the dependence model and

```
#weight(
  0.8 #combine( international organized crime )
  0.1 #combine(
    #1( organized crime )
    #1( international organized )
    #1( international organized crime ) )
  0.1 #combine(
    #uw8( organized crime )
    #uw8( international crime )
    #uw8( international organized )
    #uw12( international organized crime ) ) )
```

Figure 3.2. TREC topic 301, "international organized crime," converted to an Indri query by Metzler and Croft's dependence models. This query gives a weight of 0.8 to the unigram model of the topic. The ordered phrases, `#1`, have a weight of 0.1 as well as the unordered windows, `#uwN`. Not shown here is the unigram relevance model that provides a pseudo-relevance feedback component when combined with the dependence model query for our baseline run.

the relevance model are combined to create the final baseline query using Indri's #weight operator.

The baseline is also used as the initial retrieval for both find-similar and iterative relevance feedback.

Our implementation of iterative relevance feedback is akin to that used by Rocchio (1971). We mix in a model of the relevant documents with the original baseline query model using Indri's #weight operator. We tried weights of 0.0, 0.3, 0.5, and 0.7 for the original query and found 0.3 to work best. The model of relevant documents is calculated as:

$$P(w|M_R) = \frac{1}{k} \sum_{i=1}^{k} P(w|D_i)$$

where k is the number of documents the user has judged to be relevant. An alternative is for us to replace the pseudo feedback component of the baseline query model with

Parameter	Value
Dirichlet smoothing for unigram terms, m	1500
Dirichlet smoothing for ordered and unordered windows, m	2000
Weight of unigram model in dependence model	0.8
Weight of ordered windows model in dependence model	0.1
Weight of unordered windows model in dependence model	0.1
Number of pseudo feedback documents for relevance model	10
Weight of dependence model when mixed with pseudo relevance model	0.3
Max. terms in pseudo feedback relevance model	25
Max. terms in find-similar document models	50
Max. terms in iterative feedback relevance model	50
Weight of initial query when mixed with iterative feedback relevance model	0.3

Table 3.2. Retrieval parameters.

the real relevance model as provided by the user's judgments, but we have not yet investigated this variant.

We used the same parameter settings that Metzler et al. derived from training on the TREC 2004 Robust track data and that they used for the 2005 Robust track (Metzler et al., 2005). The 2004 Robust track includes the same 150 topics we used for evaluation (topics 301-450) in its 250 topics. Table 3.2 shows the retrieval parameters' settings for all runs. We used the same smoothing parameters for all experiments.

3.2.3 Document-to-document similarity

An obvious way to implement find-similar for documents is to treat the document as a very long query. A problem with this approach is that each document will often be about several topics of which only one is the user's search topic. A document may well be about "organized crime" but it may also be about the prosecution of criminals. Not all stories about criminal prosecution are about organized crime. Rather than finding documents that are similar to all the topics mentioned in a story, we think a user will want to find documents that are similar with respect to the current search topic.

We examined two types of similarity for find-similar: *regular* and *query-biased*. Regular similarity treats the document as a query to find other similar documents. Query-biased similarity aims to find similar documents given the context of the user's search and avoid extraneous topics. For both regular and query-biased similarity, we construct a unigram model of the find-similar document that is then used as a query to find similar documents (see equation 3.3). Regular similarity uses the maximum likelihood estimated (MLE) model of the document as the query. For query-biased similarity, we create a MLE model of the document text that consists of all words within a certain distance W of all query terms in the document. For our experiments, we set W to 5. Thus the 5 preceding words, the query term, and the 5 words following a query term are used. Should a document not contain any query terms, the whole document is used. For both types of similarity, we truncate the document model to the 50 most probable terms.

Our notion of query-biased similarity is more akin to query-biased summaries (Tombros and Sanderson, 1998; Sanderson, 1998) than to query-biased clustering (Eguchi, 1999; Iwayama, 2000) or query sensitive similarity (Tombros, 2002). The nature of query-biased summaries is to extract the sentences or text surrounding query terms in a document and use this extracted text as a summary of the document. In contrast to query-biased summaries, both Eguchi (1999) and Iwayama (2000) increase the weight of query terms in the documents before clustering. Tombros' query sensitive similarity modifies the cosine similarity measure to place more weight on the query terms (Tombros, 2002).

3.2.4 Hypothetical user interfaces

We ran all of our experiments in a batch style without user involvement. Assumptions about the interface affect the batch evaluation of retrieval features. In particular, we only consider browsing patterns that could be reasonably executed by

a user with our hypothetical user interface. We next describe our hypothetical user interfaces for find-similar and iterative relevance feedback.

The find-similar interface we envision is similar to the web-based PubMed search system.[1] Our hypothetical interface has "back button" support like a web browser. If a user has performed find-similar on a document, the user can decide to stop examining the documents presented as similar to that document and hit the back button. The back button returns the user to the previous list at the position in the list where they had applied find-similar.

Results are presented in rank order with query-biased summaries for both the initial query and the find-similar lists. Sanderson (1998) demonstrated that users are able to judge the relevance of documents from simple query-biased summaries with 75% of the accuracy of full documents. Thus, we assume users will examine most documents by reading the already visible summaries. When a user applies find-similar to a document, they will be presented with a new page listing the similar documents. The find-similar lists will contain some documents that the user has already examined on previous pages. The user will have to reexamine documents unless there is a visual marker to designate already examined documents.

In our evaluation, we compared two conceptual variations of our imagined find-similar interface. In one variation non-relevant documents are reexamined and in the other they are not. Both variations prevent the reexamination of relevant documents.

The hypothetical iterative relevance feedback interface displays the top N documents of the ranked results. The user judges each of the displayed documents and then submits the feedback to retrieve the next N documents. In our experiments, we set the iteration size N to 10. The previously displayed documents are not shown again for the current topic. This process repeats until 1000 documents have been

[1] `http://www.pubmed.gov`

examined. This interface does not provide for use of a back button like find-similar. The system only allows forward iteration.

3.2.5 Find-similar browsing patterns

To evaluate find-similar in an automatic fashion without a user study requires some assumed user behavior. We chose to examine two related and plausible browsing patterns.

Klöckner et al. (2004) used an eye tracker to observe how people processed search results. They used a Google results list containing 25 results for a query. The subjects' task was to find the relevant documents in the list. Subjects could click on a result to see the result's web page. Of the subjects, 65% followed a depth-first strategy. These users examined the documents in order, from highest to lowest rank, and did not look ahead. Another 15% used a breadth-first strategy by looking at the entire results list before opening a single web page. The remaining 20% used a partial breadth-first strategy by sometimes looking ahead at a few results before opening a web page. In a second experiment, Klöckner et al. restricted the number of pages the users could open and rewarded the users for the total number of relevant pages found. This experiment aimed to create an situation that would encourage breadth-first search behavior. Nevertheless, 52% of the subject still followed a depth-first strategy, 11% used an extreme breadth-first strategy, and 37% used a mixed strategy. Aula et al. (2005) found similar behavior in an eye-tracking study of search results evaluation.

Given the user behavior observed by Klöckner et al., we used two browsing patterns to evaluate find-similar. The *greedy* pattern represents the depth-first behavior, and the *breadth-like* pattern aims to capture the breadth-first search behaviors. Neither pattern is a true depth-first or breadth-first search pattern. A true depth-first pattern does not reflect that a user is likely to stop examining a results list if no relevant documents are found. A true breadth-first pattern is not feasible for a user

to implement. While inspired by the user behavior observed by Klöckner et al., these patterns are at best crude models of user behavior. Users could execute these patterns, but we have little knowledge of how users actually search with find-similar. Instead, these patterns give us insight into the potential of find-similar and the degree to which find-similar's performance can be affected by different browsing patterns. Both patterns use the baseline as the initial retrieval.

The greedy browser examines documents in the order that they appear in a results list. As section 3.2.4 explained, the browser will only examine a relevant document once. When a relevant document is examined, the greedy browser performs a find-similar operation on this document. The greedy browser ceases to examine documents in a results list after examining 5 contiguous non-relevant documents. When the browser stops examining a list, the browser hits the "back button" and returns to the previous list and continues examining documents in that list. If the browser is examining the initially retrieved list of documents, the only stopping criterion is that the browser stops when 1000 documents have been examined.

The breadth-like browser also examines documents in the order that they appear in a results list. What differs from the greedy pattern is that the breadth-like browser only begins to browse via find-similar when the results list's quality becomes too poor. As the breadth-like browser examines relevant documents, it places these documents in a first-in first-out queue local to the current list. When the precision at N, where N is the rank of the current document, drops below 0.5 or when 5 contiguous non-relevant documents are encountered, the browser applies find-similar to the first relevant document in the queue. When the browser returns to the current list, it applies find-similar to the next document in the queue until the queue is empty. The browser never uses find-similar on a relevant document more than once. Thus documents in the queue will be ignored if the browser has already performed find-similar on them. There is not any notion that the breadth-like browser knows which

relevant documents are the best for find-similar. The breadth-like browser merely delays exploration until the current list seems to have gone "cold." The browser stops examining a results list in the same manner and with the same criterion, i.e. 5 contiguous non-relevant documents, as the greedy browser.

Early experiments with a greedy browsing pattern influenced our design of the breadth-like browser. We saw that the greedy browser could degrade the performance of an already good retrieval. Thus, the breadth-like browser uses list quality as its criterion for delaying use of find-similar. While the breadth-like browsing pattern could be seen as a "corrected" greedy pattern, we feel that it does capture the goal of a breadth-first user, that is, to look ahead before acting.

3.2.6 Queries, documents, and retrieval tools

The topics used for the experiments consisted of TREC topics 301-450, which are the ad-hoc topics for TREC 6, 7, and 8. TREC topics consist of a short title, a sentence length description, and a paragraph sized narrative. The titles best approximate a short keyword query, and we used them as our queries.

We used TREC volumes 4 and 5 minus the Congressional Record for our collection. This 1.85 GB, heterogeneous collection contains 528,155 documents from the Financial Times Limited, the Federal Register, the Foreign Broadcast Information Service, and the Los Angeles Times.

We used the Lemur toolkit (Lemur, 2003) for all of our experiments including its Indri subsystem (Strohman et al., 2005). In particular, we generated the results for the find-similar runs using a Lemur index of the collection with stop words removed at index time. For the baseline and iterative relevance feedback runs we used an Indri index with stop word removal at query time. We stemmed all words with the Krovetz stemmer (Krovetz, 1993). We used an in-house stopword list of 418 noise words (see Appendix A).

3.2.7 Evaluation methodology

We constructed our runs' results lists for evaluation in the same manner as described in Section 3.1.2. The results lists that we evaluated represent the order in which the simulated user examines the documents. For the baseline retrieval, the documents are examined in rank order. For find-similar, the browsing patterns of section 3.2.5 determine the order in which documents are examined. For iterative relevance feedback, documents are examined in the same manner they are judged — one iteration of 10 documents at a time.

All relevance judgments are made using the "true" relevance judgments per NIST. Depending on the interface support for keeping track of examined documents (Section 3.2.4), find-similar may produce a results list in which non-relevant documents are repeatedly examined. We treat a reexamined non-relevant document the same as any other non-relevant document found at that position in the results. All of the retrieval techniques we studied do not reexamine relevant documents.

3.2.7.1 On the need or lack thereof for user errors

A possible complaint about our use of the NIST relevance judgments is that by doing so, our simulation will not account for user errors. For example, should not users sometimes apply find-similar mistakenly to non-relevant documents and head down dead ends?

For two reasons, we do not believe that there is a need for a user error model for either find-similar or for relevance feedback.

Firstly, the NIST relevance judgments can be thought of as the recorded answers to asking the user to judge the relevance of a document. When it comes time to make a decision about relevance in our simulation, we simply go to the recorded answers of the user as collected by NIST. The probability of a user making an error has already been captured by NIST's collection of relevance judgments. While we

take the NIST judgments to be true in our evaluations, it is entirely possible for the NIST assessors to make errors given the time pressures and volume of documents involved in the assessment process. NIST is providing human judgments, and humans make mistakes. Some documents judged relevant are actually non-relevant and some documents judged non-relevant are actually relevant. Adding another layer of noise into the simulation does not tell us much. We already have a noisy simulation by using the NIST judgments.

Secondly, evaluation of relevance feedback systems have not traditionally incorporated a notion of users making errors in relevance judgments. Since find-similar is being used as a feedback-like search tool, we see no reason to evaluate it differently than has historically been done for relevance feedback.

3.2.7.2 Metrics and statistical testing

We report metrics using both the arithmetic mean and the geometric mean. The TREC Robust track has established the geometric mean as a useful tool for analyzing performance (Voorhees, 2005). As opposed to the usual arithmetic mean, the geometric mean emphasizes the lower performing topics. The arithmetic mean can hide large changes in performance on poorly performing topics with small changes in the better performing topics. As with the 2005 TREC Robust track (Voorhees and Dang, 2005), for computing the geometric mean, we set values less than 0.00001 to 0.00001 to avoid zeros. As such, the geometric mean is computed:

$$\left(\prod_{i=1}^{n} max(x_i, 0.00001) \right)^{1/n}$$

We used trec_eval to compute per topic metrics (Buckley, 2006). Following Smucker et al. (2007), we measured statistical significance with a two-sided, paired, randomization test with $100,000$ samples. Unless otherwise stated, significance is at the $p < 0.05$ level.

		Reexamine non-relevant				Do not reexamine non-relevant				Iter.
		Greedy		Breadth-like		Greedy		Breadth-like		Rel.
	Baseline	Regular	Biased	Regular	Biased	Regular	Biased	Regular	Biased	FB.
All 150 topics										
AM Avg. Prec.	0.262	0.175*	0.226*	0.260	0.269	0.224*	0.281	0.303*	0.323*	0.322*
Pct. Change		-33%	-14%	-1%	3%	-14%	7%	16%	23%	23%
GM Avg. Prec.	0.130	0.122	0.151*	0.157*	0.169*	0.160*	0.193*	0.197*	0.216*	0.220*
Pct. Change		-6%	16%	21%	30%	23%	49%	52%	66%	69%
Baseline's 50 poorest performing topics										
AM Avg. Prec.	0.036	0.079*	0.091*	0.083*	0.101*	0.108*	0.119*	0.114*	0.134*	0.129*
Pct. Change		119%	151%	130%	179%	197%	228%	215%	270%	255%
Baseline's 50 middle performing topics										
AM Avg. Prec.	0.202	0.160*	0.190	0.190	0.196	0.218	0.261*	0.251*	0.275*	0.256*
Pct. Change		-21%	-6%	-6%	-3%	8%	29%	24%	36%	27%
Baseline's 50 best performing topics										
AM Avg. Prec.	0.548	0.285*	0.396*	0.505*	0.509*	0.346*	0.461*	0.544	0.560	0.580*
Pct. Change		-48%	-28%	-8%	-7%	-37%	-16%	-1%	2%	6%

Table 3.3. Arithmetic mean (AM) and geometric mean (GM) average precision for all 150 topics and the arithmetic mean average precision for the 150 topics grouped into three disjoint sets based on the baseline's average precision for that topic. Results with a $\star$ are different from the baseline at a statistically significant level ($p < 0.05$) as measured by a two-sided, paired, randomization test with 100,000 samples.

		Reexamine non-relevant				Do not reexamine non-relevant				Iter.
		Greedy		Breadth-like		Greedy		Breadth-like		Rel.
	Baseline	Regular	Biased	Regular	Biased	Regular	Biased	Regular	Biased	FB.
Precision at 10 documents										
Arith. Mean	0.428	0.335*	0.397*	0.439	0.442	0.345*	0.409	0.446	0.457*	0.428
Pct. Change		-22%	-7%	3%	3%	-19%	-4%	4%	7%	0%
Geo. Mean	0.093	0.081*	0.092	0.098	0.099	0.083*	0.095	0.100*	0.104*	0.093
Pct. Change		-13%	-1%	5%	6%	-11%	2%	7%	11%	0%
Precision at 20 documents										
Arith. Mean	0.374	0.254*	0.330*	0.372	0.387	0.282*	0.358	0.395*	0.415*	0.411*
Pct. Change		-32%	-12%	-1%	3%	-25%	-4%	6%	11%	10%
Geo. Mean	0.120	0.095*	0.116	0.121	0.130	0.104*	0.128	0.132*	0.143*	0.137*
Pct. Change		-21%	-3%	0%	8%	-13%	6%	9%	19%	14%
Precision at 100 documents										
Arith. Mean	0.225	0.163*	0.206	0.219	0.236	0.204*	0.246	0.250*	0.274*	0.277*
Pct. Change		-28%	-8%	-3%	5%	-10%	9%	11%	22%	23%
Geo. Mean	0.122	0.106*	0.128	0.125	0.137*	0.129	0.152*	0.145*	0.163*	0.162*
Pct. Change		-13%	5%	3%	12%	6%	25%	19%	34%	33%

Table 3.4. Arithmetic mean and geometric mean of the precision at 10, 20 and 100 documents. Results with a $\star$ are different from the baseline at a statistically significant level ($p < 0.05$) as measured by a two-sided, paired, randomization test with 100,000 samples.

		Reexamine non-relevant				Do not reexamine non-relevant				Iter.
		Greedy		Breadth-like		Greedy		Breadth-like		Rel.
	Baseline	Regular	Biased	Regular	Biased	Regular	Biased	Regular	Biased	FB.
Arith. Mean	0.687	0.741*	0.750*	0.747*	0.746*	0.806*	0.809*	0.808*	0.811*	0.823*
Pct. Change		8%	9%	9%	9%	17%	18%	18%	18%	20%
Geo. Mean	0.603	0.688*	0.703*	0.695*	0.700*	0.763*	0.765*	0.764*	0.767*	0.779*
Pct. Change		14%	17%	15%	16%	26%	27%	27%	27%	29%

Table 3.5. Arithmetic mean and geometric mean of the recall at 1000 documents. Results with a $\star$ are different from the baseline at a statistically significant level ($p < 0.05$) as measured by a two-sided, paired, randomization test with 100,000 samples.

3.3 Results

Table 3.3 shows the arithmetic mean, non-interpolated, average precision (AMAP) and the geometric mean (GMAP) across the 150 topics of TREC 6, 7, and 8, for the baseline, find-similar, and iterative relevance feedback runs. The find-similar runs vary based on whether or not non-relevant documents were reexamined (section 3.2.4), whether a greedy or breadth-like browsing pattern was used (section 3.2.5), and whether the similarity was regular or query-biased (section 3.2.3).

In general, find-similar and iterative relevance feedback are better able to improve on a poor initial retrieval than on a good initial retrieval. To highlight this behavior, Table 3.3 also reports results for the 150 topics divided into three sets of 50 topics. The topics are ordered by their performance on the baseline and then divided into three sets (like quartiles except into thirds instead of quarters). These sets are roughly equivalent to poor, fair, and good retrieval performance with baseline AMAPs of 0.036, 0.202, and 0.548 respectively. With the topics divided up in this manner, the geometric mean adds little insight and we report only the arithmetic mean of each topic set.

The average precision results are based on the TREC standard of 1000 results. To understand the performance when a user examines fewer documents, Table 3.4 shows the precision at 20 and 100 documents. Feedback techniques can increase recall as well as precision. Table 3.5 shows the recall at 1000 documents.

3.4 Discussion

The best find-similar run avoids reexamining non-relevant documents, follows a breadth-like browsing pattern, and uses query-biased similarity. Table 3.3 shows that this run matches the performance of our implementation of iterative relevance feedback and achieves a 23% improvement in the arithmetic mean average precision (AMAP) and a 66% improvement in the geometric mean average precision (GMAP)

over the baseline. Iterative relevance feedback achieves a 69% improvement in GMAP, but this is not a statistically significant difference compared to the best find-similar run.

The use of a high quality baseline retrieval is required to avoid overstating the performance gains possible with a retrieval technique. We used the state-of-the-art method developed by Metzler et al. (2005) for our baseline (see section 3.2.2). This method had the best title run as measured by mean average precision and had the second best geometric mean average precision for both title and description runs submitted to TREC's 2005 Robust track (Voorhees and Dang, 2005). We achieved larger relative performance improvements during initial experiments with a weaker baseline (query likelihood without any query expansion).

We also tested iterative relevance feedback with an iteration size of 1, which is Aalbersberg's incremental feedback (Aalbersberg, 1992). An iteration size of 1 performed as well as an iteration size of 10, with the larger iteration size yielding a negligibly larger AMAP (0.322 vs. 0.321). This result runs counter to Iwayama's (2000) negative results for incremental feedback.

We performed an analysis of variance for the find-similar $2 \times 2 \times 2$ factorial designed runs (see Table 3.1) with average precision as the dependent variable. We looked at the three independent variables and their interaction given the blocking by topics. All three variables resulted in statistically significant outcomes ($p < 0.001$). The browsing pattern had the most impact on find-similar, and the user interface's support for the avoidance of reexamination of documents had more impact than the document-to-document similarity measure. Of the 4 possible interactions between variables, only the interaction of similarity and browsing pattern was statistically significant ($p < 0.001$). Figure 3.3 shows an interaction plot of similarity and browsing pattern. The query-biased similarity aided the greedy browsing pattern more than breadth-like browsing pattern.

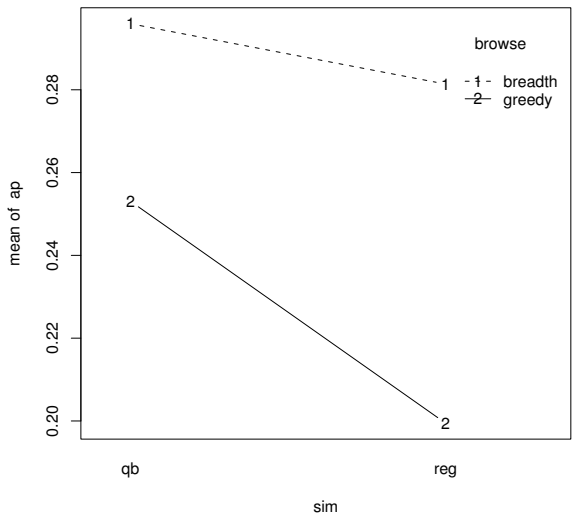


Figure 3.3. Interaction plot of browsing pattern and document-to-document similarity. The breadth-like and the greedy are the two browsing patterns. The x-axis shows the two similarity types: query-biased (qb) and regular (reg). The y-axis shows the mean average precision (ap). The query-biased similarity aids the greedy browsing pattern more than the breadth-like browsing pattern.

All the find-similar runs that avoid reexamination of non-relevant documents perform better than the corresponding runs that do reexamine non-relevant documents. An interface that supports find-similar may need to provide a mechanism to help the user avoid reexamination of non-relevant documents. Web interfaces that provide links to documents already come close to providing this functionality by changing the color of visited links. If a user has to keep track of judgments, it would seem that find-similar and traditional multiple item relevance feedback should be able to co-exist in the same retrieval system.

While both the greedy and breadth-like browsing patterns show significant improvements in GMAP over the baseline, following a breadth-like browsing pattern is superior to the greedy browsing pattern. Table 3.3 shows that the greedy browsing pattern in particular has difficulty with the better performing topics. As section 3.2.5 noted, the work by Klöckner et al. (2004) motivated the two browsing patterns we used, but the performance of the greedy pattern influenced our design of the breadth-

like browser. A user that follows a greedy browsing pattern will be harmed by the find-similar feature on better performing topics. The breadth-like browsing pattern avoids using find-similar while the retrieval quality of a list is high. We leave for future work the question of whether find-similar can be used to improve an already high quality retrieval.

We see the breadth-like browser as a more reasonable browsing pattern for a user to follow than the greedy browsing pattern. The greedy pattern shows us that a user could hurt already good results, but when the results are already good, why would a user utilize find-similar? Given this argument, the breadth-like browsing pattern's results should be given more weight than the greedy browser's results.

Query-biased similarity shows consistently better performance than regular similarity. The query-biased similarity helps the greedy browsing pattern perform over 20% better than with regular similarity as measured by AMAP and GMAP on all 150 topics. Query-biased similarity also helps the breadth-like browser but to a lesser degree.

Given a search topic, a perfect document-to-document similarity method for find-similar makes the topic's relevant documents most similar to each other. In Chapter 5 we characterize this notion of relevant documents being more similar to each other by measuring the distance from all relevant documents to all other relevant documents. We show that query-biased similarity creates a tighter grouping of relevant documents than does regular similarity.

The find-similar and feedback runs show a much greater improvement in GMAP than in AMAP. Table 3.3 highlights this difference and shows that the majority of the improvement comes from improving the poorer performing topics. For the poorest performing topics, the baseline has an AMAP of 0.036, and on average, 1 document in 28 is relevant. On these same topics, find-similar raises this ratio to 1 in 7 with an AMAP of 0.134. Besides having a large relative performance improvement for

poorly performing topics, find-similar can provide performance gains that should be noticeable by the end user.

Being able to improve precision early in a ranked list may influence user adoption of a retrieval tool such as find-similar. Table 3.4 shows that find-similar can achieve improvements over the baseline in precision at 10, 20, and 100 documents. The best find-similar run also obtained a statistically significant 7% increase in P@10 (arithmetic mean) over the baseline. Iterative relevance feedback with an iteration size of 10 documents shows no improvement in P@10 because our run forces the "user" to judge the first 10 documents before submitting feedback.

For find-similar's best run, its P@100 arithmetic mean improvement of 22% is comparable to its AMAP improvement of 23%. For this same run, the P@100 geometric mean improvement of 34% is nearly half that of the 66% improvement in GMAP. A fair amount of the GMAP performance may come from improving very poorly performing topics with feedback on low ranking relevant documents. For some poor performing topics, if users are unwilling to dig deep into the ranked results, they may be unable to use feedback to help their search.

Table 3.5 shows that all of the find-similar runs increase recall at 1000 documents and the best performance is comparable to iterative relevance feedback. Retrieval techniques that cluster or reorder the top N results cannot increase the recall at N (Iwayama, 2000; Leuski, 2000). Interestingly, the different similarity and browsing types do not significantly impact recall at 1000 documents.

Given these results, implementors of search systems with find-similar should look to ways to encourage good user behavior. For example, an interface that helps the user keep track of examined documents is very important. Interfaces should also help the user delay exploration with find-similar, which is the important characteristic of the breadth-like browsing pattern compared to the greedy browsing pattern. Finally,

where possible, a query-biased similarity should be used to make relevant documents easier to reach.

3.5 Conclusion

We found that find-similar, as a feedback-like search tool, has the potential to improve document retrieval. The best performance improvement attained by find-similar matched that of an implementation of relevance feedback. Find-similar achieved a 23% improvement in the arithmetic mean average precision and a 66% improvement in the geometric mean average precision. The geometric mean emphasizes the poorer performing topics.

We found differences in performance for find-similar along the dimensions of document-to-document similarity, reexamination of documents, and the browsing pattern. First, we discovered that a query-biased similarity performs significantly better than using a document as a query for find-similar. Secondly, interfaces supporting find-similar as a search tool will likely need to help the user avoid reexamining already examined documents. Finally, a user's browsing pattern can substantially affect the performance of find-similar. Between two simulated browsing patterns, we found that a breadth-first like pattern works better than a greedy, depth-first like pattern. Both patterns show significant improvement in the geometric mean average precision over a strong baseline retrieval.

We continue our investigation of find-similar in Chapter 4 by utilizing the simulation developed in this chapter to investigate the effect of a wide range of initial conditions on find-similar as well as to look at find-similar's applicability to another domain: biomedical search.

CHAPTER 4

CASE STUDY OF PUBMED AND THE EFFECT OF VARYING INITIAL CONDITIONS ON FIND-SIMILAR

In Chapter 3 we used simulation to evaluate the potential of find-similar compared to a state-of-the-art, baseline retrieval system as well as to relevance feedback. In those experiments, the baseline formed the initial conditions for find-similar. In this chapter, we examine the effect of many different initial conditions on find-similar's performance. We conduct these experiments in the context of a biomedical search engine, PubMed. The work in this chapter offers another, expanded analysis of the work in Lin and Smucker (2008).

4.1 Introduction

PubMed, the U.S. National Library of Medicine's search engine, plays an important role in providing access to the biomedical research papers. In a 24 hour sample of PubMed's logs, Herskovic et al. (2007) found that 627,455 different users issued 2,996,301 queries. While an important resource, PubMed is difficult to use. The underlying search uses Boolean retrieval, which can be very difficult for users to utilize well (Turtle, 1994). Evidence of PubMed's difficulty comes from its own query logs. Long chains of query reformulations have been found, and approximately a fifth of all PubMed queries result in zero results.[1]

PubMed is more than its Boolean retrieval engine. As we mentioned in Chapter 1, PubMed provides find-similar functionality to users. PubMed calls find-similar

[1]Lin and Smucker (2008) cite unpublished results of Jimmy Lin and W. John Wilbur.

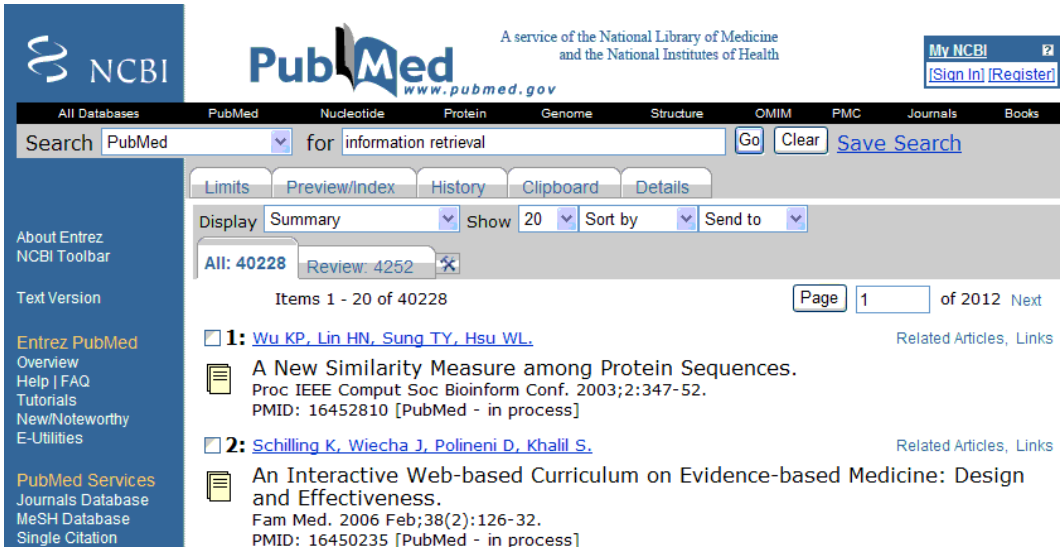


Figure 4.1. PubMed interface. The "related articles" text is a hyperlink that provides a list of similar documents for the given result, i.e. related articles is find-similar.

"related articles." Figure 4.1 shows the PubMed interface. Lin et al. (2007) have reported that nearly a fifth of all search sessions involve clicks on documents suggested by find-similar. Our hypothesis is that find-similar compensates for the poor retrievals users obtain from the underlying PubMed search system. In this chapter, we aim to determine the extent to which find-similar can compensate for poor retrievals and in particular, how find-similar can help make the overall PubMed search experience better.

In Chapter 3, we saw some evidence that find-similar's biggest gains come from helping the poorer performing topics. But we only saw that this was the case for one retrieval algorithm – the state-of-the-art baseline. In this chapter, we investigate how find-similar performs given a wide range of initial conditions, i.e. initial retrievals or alternate baseline systems.

Find-similar relies on the user being able to find some relevant documents as starting points. Assuming the user can find one or more relevant documents, it is quite likely that some relevant documents are better starting points than others. This

chapter investigates this issue while at the same time studying how find-similar can aid PubMed. We are able to do both by utilizing a large collection of actual retrievals by many different search systems over a subset of the same data that PubMed uses.

Another way to express the problem addressed by this chapter is to talk about the documents and find-similar's similarity measure in terms of graph theory. In Chapter 5, we will define networks where the nodes are documents and the similarity measure determines the links between documents. On this network, documents are nearer to other documents that the similarity measure considers to be similar. When a user issues a query, the retrieval method effectively creates for the user a temporary document/node and creates links to the other documents. Depending on how a retrieval algorithm connects a user's query to the other documents, find-similar may or may not perform well. Some relevant documents may be well connected to other relevant documents while some may represent "similarity dead ends" for find-similar.

4.2 Materials and Methods

In this section we describe the set of initial retrievals, document collection, topics, and find-similar simulation methods used in our experiment.

4.2.1 Initial Retrievals

We used the 62 runs submitted to the TREC 2005 Genomics track as our initial conditions. 32 groups submitted 58 runs to the pool and one group submitted an additional 4 runs not included in the judging pool (Hersh et al., 2005; Huang et al., 2005). These runs can be taken to be representative of the various results that users could have produced for the 49 topics. As we show in Section 4.3, these runs vary widely in their quality.

In our study of PubMed and find-similar, we are concerned with find-similar's ability to help improve poor retrieval performance. There are many reasons why the search results of a user can be poor. Three of these reasons are:

1. The underlying retrieval system may have a low quality ranking algorithm.

2. The user can have a complex information need that is difficult for any ranking algorithm.

3. Novice users are likely to have difficulty forming good queries.

With the 62 runs and 49 topics, we have potential examples of all three of the above reasons for poor retrieval quality: retrieval method, complex information needs, and query formulation. Each submitting group has some underlying retrieval algorithm and some algorithms are better than others. It is well known that most of the variation in retrieval quality comes from the topics with some topics being harder than others. The TREC Genomics track gives the participating groups topics, but does not specify keyword queries or other query formulation requirements. Each group must furthermore convert the topic into some explicit query usable by their retrieval system and this task is representative of the task users face in constructing queries.

While these runs can be representative examples of varying levels of retrieval quality, we do not know the cause of an individual retrieval's success or failure. What we can do is study the average performance of the runs and the topics. Treating each run as a different retrieval system, the average performance of a run relative to the other runs allows us to examine find-similar's ability to improve retrieval quality given the performance of a retrieval system. Similarly, we take the average performance of a topic to be an indication of the difficulty or complexity of the topic's information need. In other words, we define difficult topics to be the topics for which many systems produce poor retrievals. We do not have a means of looking at the quality of various queries.

Information describing standard [methods or protocols] for doing some sort of experiment or procedure.
methods or protocols: purification of rat IgM

Information describing the role(s) of a [gene] involved in a [disease].
gene: PRNP
disease: Mad Cow Disease

Information describing the role of a [gene] in a specific [biological process].
gene: casein kinase II
biological process: ribosome assembly

Information describing interactions between two or more [genes] in the [function of an organ] or in a [disease].
genes: Ret and GDNF
function of an organ: kidney development

Information describing one or more [mutations] of a given [gene] and its [biological impact or role].
gene with mutation: hypocretin receptor 2
biological impact: narcolepsy

Figure 4.2. The five templates used in the TREC 2005 genomics track with sample topics.

4.2.2 Document Collection

We used the 2004 TREC Genomics document collection (aka MEDLINE04) for our experiments along with the 2005 TREC Genomics topics. The document collection is a 10 year subset of MEDLINE (1994–2003) containing 4,591,008 records and was about one third the size of PubMed's collection when created. (PubMed indexes all of MEDLINE as well as some additional materials.) Most records contain a title, abstract, and associated metadata. We indexed the title and abstract and excluded other metadata. More details about the collection can be found in the 2004 track overview (Hersh et al., 2004).

4.2.3 Topics

The 49 TREC 2005 Genomics topics were built around 5 templates. Figure 4.2 shows examples of the topics. For our experiments, we utilized the topics' relevance

judgments to evaluate the performance of find-similar. The creators of the dataset solicited information needs from 25 biologists. Five people with varying biology experience provided the relevance judgments. More details concerning the topics can be found in the 2005 Genomics overview (Hersh et al., 2005). We did not use the topics to influence retrievals or document-to-document similarity.

4.2.4 Find-Similar Simulation

For find-similar, we used the breadth-like browsing pattern (Section 3.2.5). A variable of the breadth-like browsing pattern is the number of contiguous non-relevant documents that the browser will examine before beginning its exploration of similar relevant documents and/or hitting the "back button" of the hypothetical web browser. In Chapter 3, we set this variable to 5 non-relevant documents and do the same here, but we also briefly look at setting this variable to 2 non-relevant documents. There is little difference in the overall results between the choice of 5 or 2 non-relevant documents, and we report additional results for the browsing pattern with this variable set to 2 non-relevant documents in another work (Lin and Smucker, 2008).

For document-to-document similarity, we used regular similarity (Section 3.2.3) to mimic PubMed's similarity, which is not query-biased (Lin and Wilbur, 2007). As in Chapter 3, we stemmed with the Krovetz stemmer and removed stopwords based on our list of 418 stopwords. We set the Dirichlet prior smoothing parameter to 1500. We used a document's title and abstract concatenated together as our representation of a document.

We also used PubMed's similarity by obtaining the set of "related articles" for the known relevant document via PubMed's API. Our comparison to PubMed's similarity is more for the purposes of verifying our results rather than for direct comparison of similarity measures.

To obtain the related articles we used PubMed's eLink[2] interface and specified the `pubmed` database. We restricted results to have completion dates (CDAT) between 1994/1/1 and 2003/12/31, which is what the track did for the collection (Hersh et al., 2004). All other options are the default options for the interface call. We post-filtered any results that did not have a PubMed identifier (PMID) within the TREC 2005 Genomics collection. We used the PMIDs that are part of the first "LinkSet." There also seemed to be a second linkset returned, but we could not determine its purpose and did not include the PMIDs it provides. Because of the lack of documentation for PubMed's API, it is possible that we did not extract the complete set of related articles. The first linkset appeared to be the correct related articles. Each related articles list is approximately 140 results long. We found that there is a relevant PMID, 11706583, in the TREC 2005 Genomics collection that PubMed says does not exist. Thus, it has no related articles.

For another experiment, we programmatically issued queries against the PubMed API, which in this case is called the eSearch interface. In this case, we requested 2000 results, and then post-filtered and capped the results at 1000 results.

4.3 Results and Discussion

Figure 4.3 shows the distribution of precision at rank 20 (P20) for the 3038 initial retrievals (62 runs $\times$ 49 topics) and the topmost chart of Figure 4.6 shows the same for the average precision (AP). In both measures we have a wide range of performance and a large number of poor performing runs.

Table 4.1 shows overall results for find-similar when applied to the initial retrievals. These results are in line with those for TREC 6,7,8 in Chapter 3 where we found a 16%

[2] `http://www.ncbi.nlm.nih.gov/entrez/query/static/eutils_help.html`

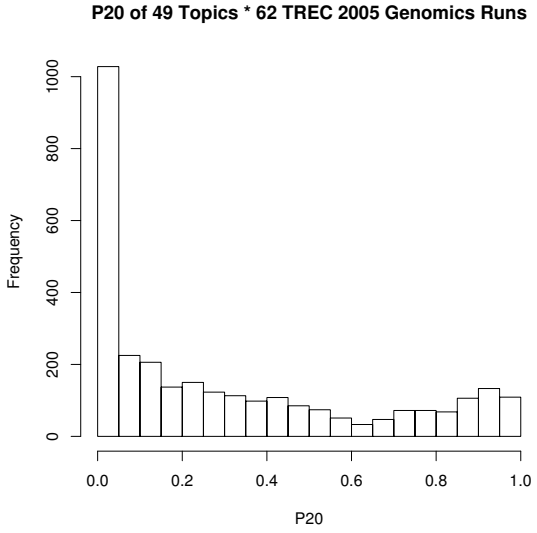


Figure 4.3. Distribution of precision at rank 20 (P20) for the 3038 initial retrievals submitted to the TREC 2005 Genomics track.

	Initial	With Find-Similar (Breadth-like browser, Max 5 Non-rel)			
	Results	Regular Sim.	Improvement	PubMed Sim.	Improvement
AP	0.200	0.231	15.3%	0.224	11.7%
P20	0.318	0.336	5.5%	0.327	2.7%

Table 4.1. Average precision (AP) and precision at rank 20 (P20). Overall averaged results for the 62 runs submitted to the TREC 2005 Genomics track both as submitted and with find-similar. Find-similar was simulated with no reexamination of documents, a maximum of 5 contiguous non-relevant documents, and a breadth-like browsing behavior. Shown are results for both regular similarity and PubMed's similarity. All find-similar gains over the initial results are statistically significant gains ($p < 0.05$) as measured by a paired, two-sided t-test with pairing done at the run level.

gain in mean average precision and a 6% gain in P20 for a breadth-like browser that avoided the reexamination of documents and used regular similarity (see Table 3.3).

Figure 4.4 shows the average P20 with find-similar vs. P20 without find-similar. Shown are the results for both the regular similarity implemented with Lemur and the PubMed similarity downloaded from PubMed. Of note is the gain for the lower performing P20 values. When the initial retrieval is poor, then find-similar offers the most value. It is possible for find-similar to hurt results if it is used when the results are already good. Note that for very high values of P20, the breadth-like browser will not apply find-similar and thus those results will not be degraded. We think that even for the P20 values of 0.4 and higher, where find-similar degraded performance, that in actual use, users are unlikely to use find-similar with such good results.

The top chart of Figure 4.5 shows a closeup of the same results as Figure 4.4. Here we see the substantial gains that find-similar offers when the initial results are poor. For initial P20 results of 0.05 to 0.15, find-similar produces over a 50% relative gain. Note that when P20 is zero, find-similar cannot improve the results since it relies on the user finding at least one relevant document. When P20 is zero, some other interaction mechanism would be needed to help the user improve the results.

The PubMed similarity performs worse than regular similarity, but the difference is relatively small. In particular, when we set the maximum number of contiguous non-relevant documents parameter to 2 instead of 5, there is effectively no difference between the two similarity measures. This raises the important point that when using simulation to evaluate systems, a good practice would be to report results for a range of parameter values unless those parameter values are supported by other research.

Figure 4.6 shows the results for the average precision (AP). In the top chart, we see the distribution of AP across the runs and topics. Like P20, the majority of initial retrievals result in a low AP. The middle chart shows a scatter plot of the AP both with and within find-similar. Points above the line are improvements while points

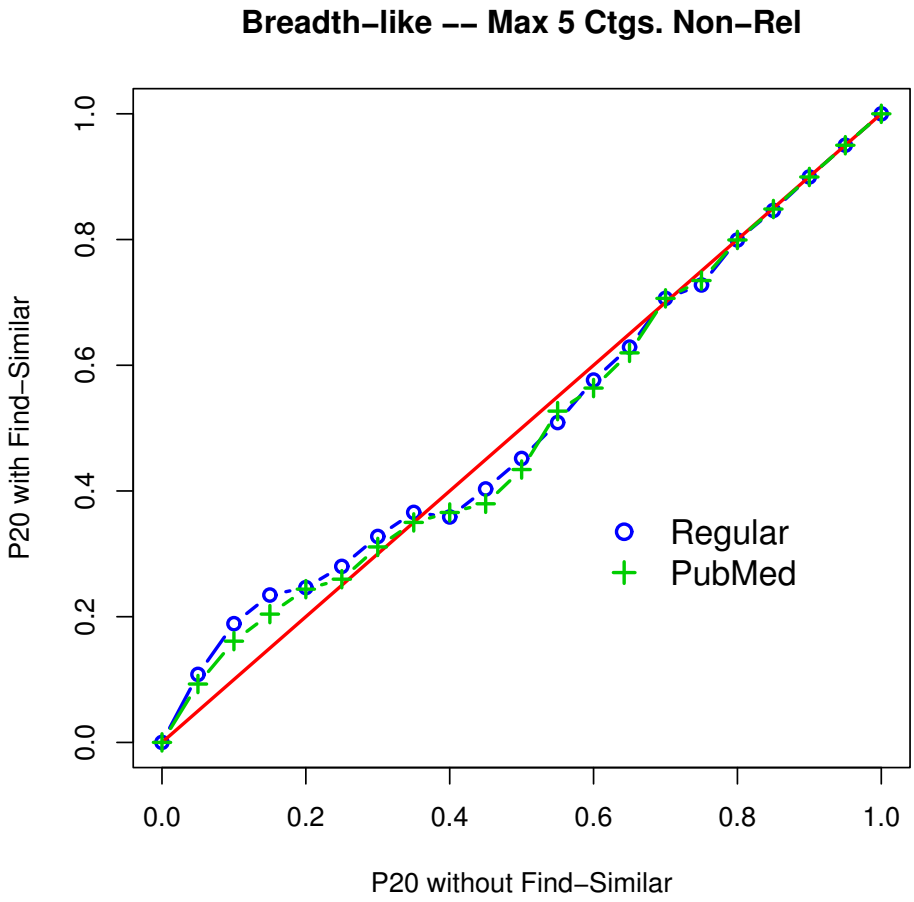


Figure 4.4. The precision at rank 20 (P20) with and without find-similar. The 3038 data points with find-similar are averaged for each of the 21 possible values of P20 without find-similar. Shown are the results for regular similarity and for the PubMed similarity. Find-similar utilized a breadth-like browsing pattern, a maximum of 5 contiguous non-relevant documents, and avoided the reexamination of documents.

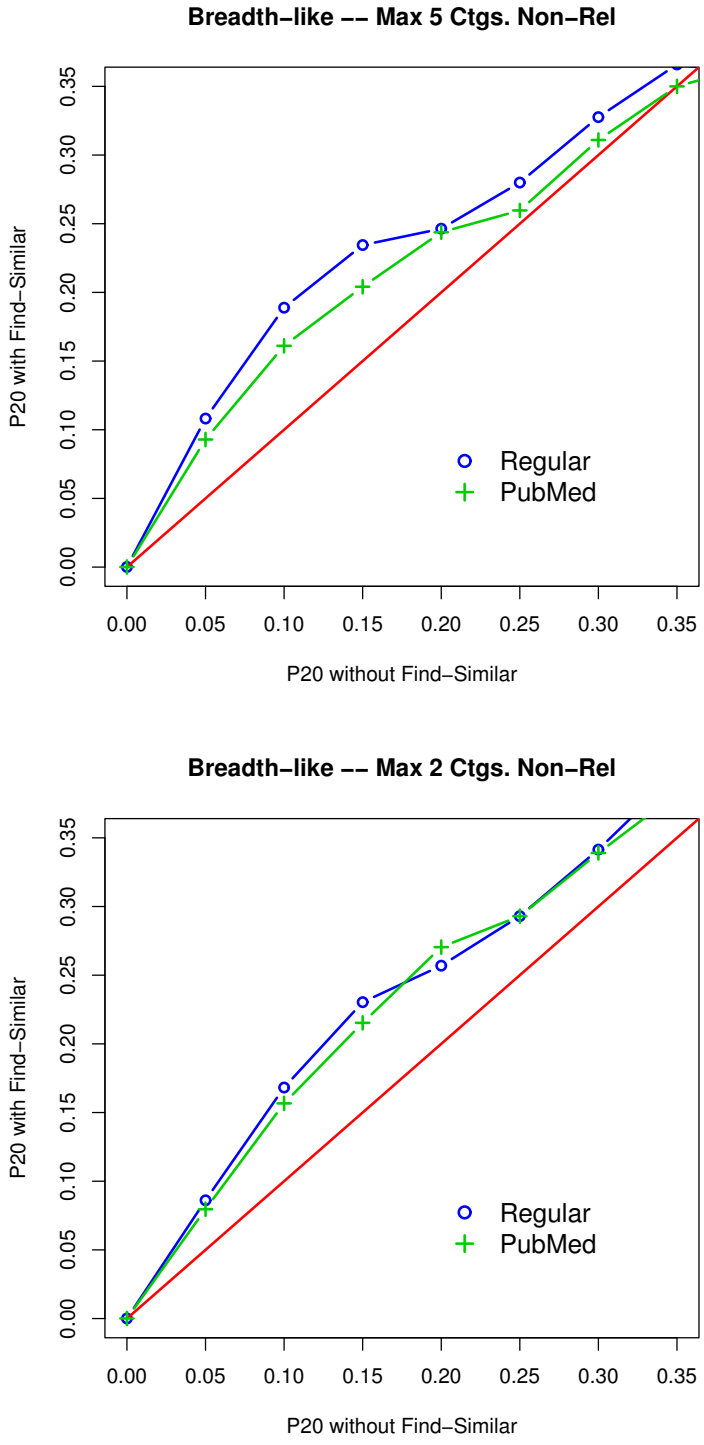


Figure 4.5. This figure shows a closeup view of Figure 4.4 on the top. On the bottom, this figure shows the same except the maximum number of contiguous non-relevant documents parameter has been set to 2 documents.

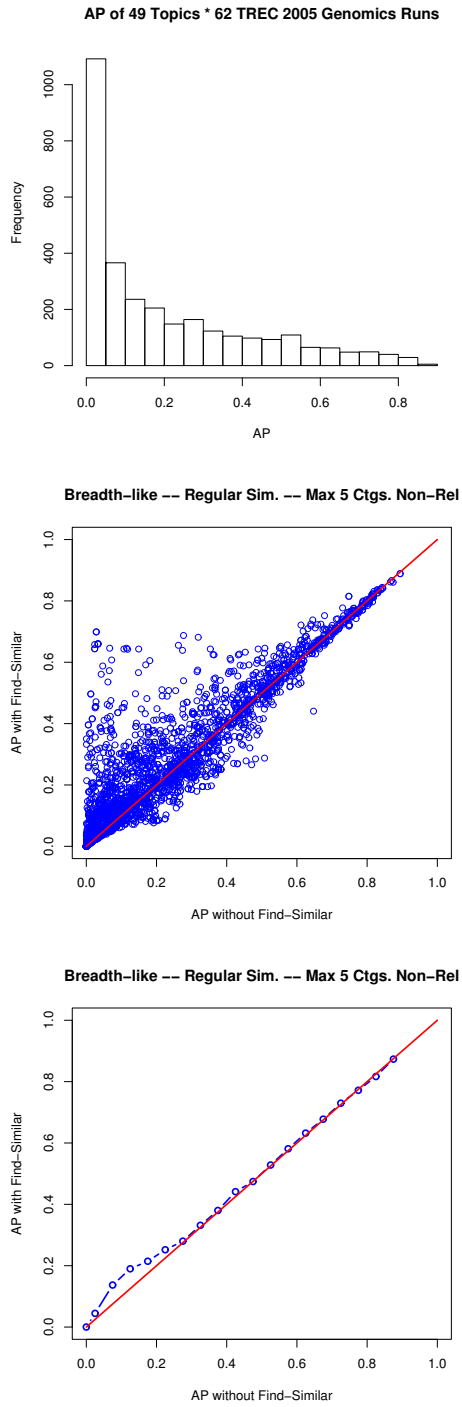


Figure 4.6. The top chart shows the distribution of the average precision (AP) without find-similar. The middle chart shows a scatter plot of the AP with and without find-similar, and points above the line are improvements. In the bottom figure we show the average of AP with the data of the middle chart placed into bins 0.05 wide. We have artificially added a point at 0 for the purposes of drawing a representative line between the bins.

below the line represent degradations caused by find-similar. The bottom chart shows the AP results averaged into bins. On average, we see that find-similar provides gains when the initial retrievals are poor in quality.

Taken together, the results for P20 and AP show that find-similar as an interaction mechanism works to compensate for poor retrievals. This is significant given that no matter what retrieval algorithm is used, there will always be cases where the user has difficulty obtaining high quality results. Find-similar acts to make the overall retrieval system higher performing and more robust. By more robust we mean the system can handle a wider range of topics and queries and still offer the user a route to better results. IR systems can utilize many such interaction mechanisms to provide higher quality results to the user without having to change the underlying retrieval algorithm.

Figure 4.7 shows P20 and AP with and without find-similar with the data grouped and averaged by run. For the TREC 2005 Genomics track, there were 62 submitted runs. Each run represents a retrieval method and means to convert the topic into a query. Here we see that find-similar aids the poorer performing systems and is unable to help the best performing ones.

Figure 4.8 shows P20 and AP with and without find-similar with the data grouped by topic. For the TREC 2005 Genomics track, there were 49 topics. Here we see that find-similar is unable to help the worst performing topics and has the most success with the mid to good performing topics (topics with an AP of 0.2 or higher). Taken with Figure 4.7, it appears that find-similar helps most when a retrieval system performs poorly. When the topic is inherently difficult, find-similar appears to be of little help.

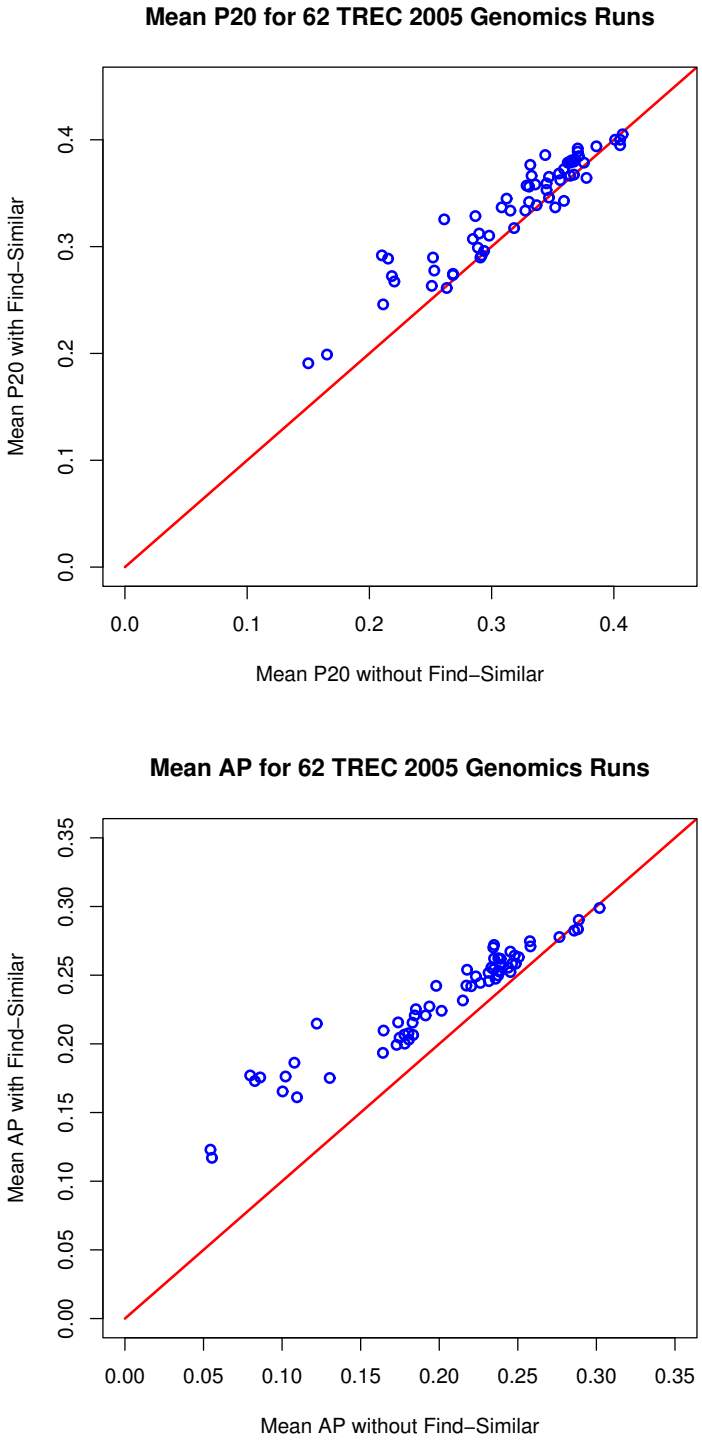


Figure 4.7. Shown on the top is the precision at rank 20 (P20) and on the bottom the average precision (AP) both with and without find-similar. Each point is a submitted run's average score for the 49 topics of the TREC 2005 Genomics track.

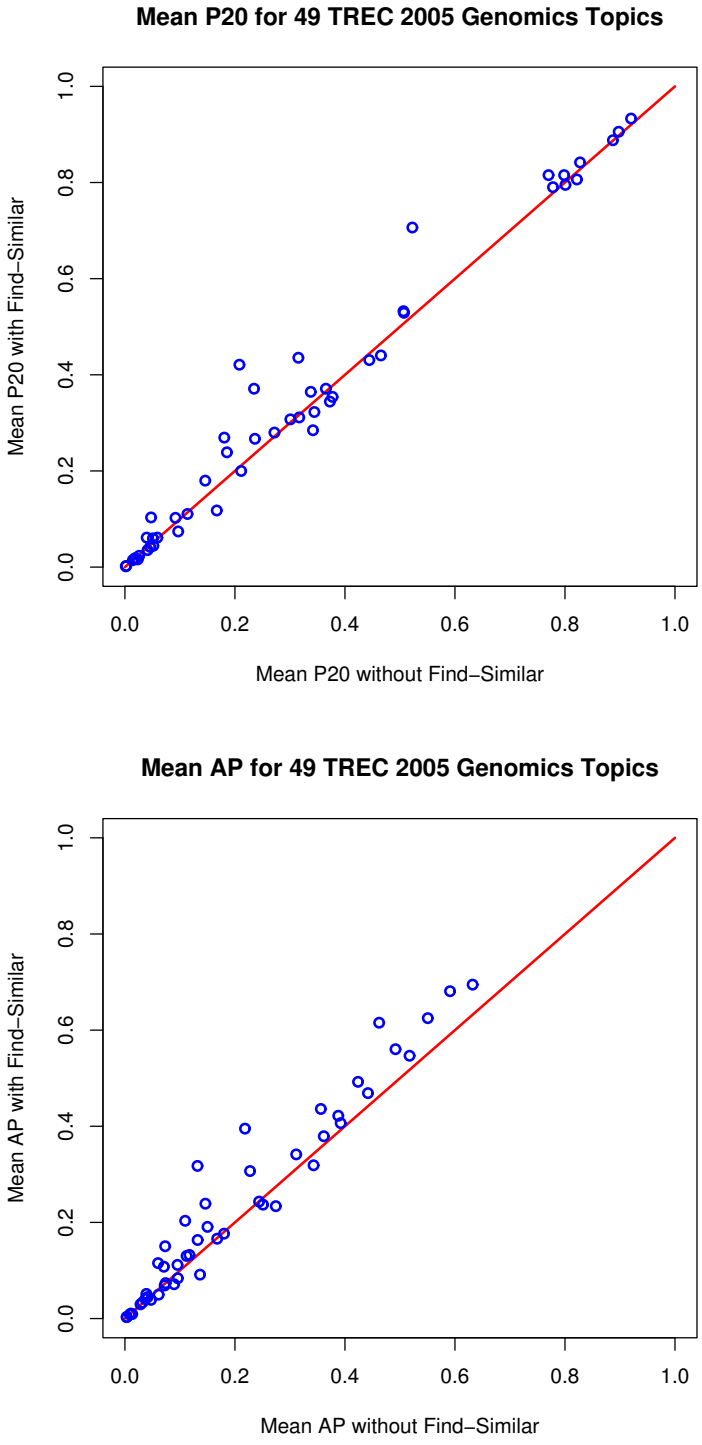


Figure 4.8. Shown on the top is the precision at rank 20 (P20) and on the bottom the average precision (AP) both with and without find-similar. Each point is a topic's average score for 62 submitted runs to the TREC 2005 Genomics track.

4.3.1 Helping PubMed Users

The above results show how find-similar can improve poor initial retrievals, but does not specifically address the improvement possible for PubMed, which has a different retrieval algorithm than the submitted systems to the TREC 2005 Genomics track. To get a sense of how much improvement is possible for PubMed, we took the 49 topics and for each one of them we crafted a sensible but simple Boolean query. We attempted to modify the original topic description as little as possible. We refined the queries until at least one relevant document was returned in the top 1000 results. For many topics, several iterations of query refinement were required and in many cases we needed to examine known relevant documents to determine how to craft a working query.

The default Boolean AND of PubMed in many cases leads to high precision, low recall results, but some topics did get a very good average precision. The number of results returned by PubMed was highly variable with 264 results on average and the median number being 58 results.

Across the 49 topics, our queries to PubMed resulted in a mean average precision (MAP) of 0.096 and a precision at rank 20 (P20) of 0.183. When we simulated find-similar usage for these results, the MAP jumped to 0.185 and P20 increased to 0.262. The nearly 100% increase in MAP performance likely comes from the increased recall of the find-similar searching. Without find-similar, only 1244 relevant documents are retrieved across the 49 topics while with find-similar 2979 relevant documents are found. To simulate find-similar, we used the breadth-like browser, a maximum of 5 contiguous non-relevant documents, no reexamination of documents, and the PubMed similarity.

While our queries could be considered to be those crafted by an expert searcher since the searcher is an IR researcher, the searcher had no training in the PubMed language and has little knowledge of the biomedical topics. While we cannot draw

any generalizations from this experiment, the results are indicative and in line with the results of the other experiments in this chapter. Find-similar appears to have the ability to compensate for poor PubMed retrievals and help the user find relevant documents.

4.4 Conclusion

In this chapter we saw how find-similar acts to compensate for poor retrievals. Find-similar appears to offer PubMed users a route to improved retrievals. Find-similar helped IR systems that performed poorly on the TREC 2005 Genomics track while it was unable to aid the best performing systems. On a topic level, find-similar is able to help the less difficult topics while the most difficult topics are beyond find-similar's reach.

CHAPTER 5

MEASURING THE NAVIGABILITY OF DOCUMENT NETWORKS

Find-similar provides the search user a means to travel from one document to another. In effect, find-similar links documents into a network, and just as a traveler in the physical world needs a good road system with direct routes, the search user needs find-similar to produce links that minimize the travel time to relevant documents.

A find-similar tool embodies some document-to-document similarity method. We would like to be able to test many variations of document-to-document similarity in well defined, a low cost manner that is largely independent of the user interface. While our simulated users in Chapter 3 are very affordable, the results are tied to a given user interface.

In this chapter, we utilize a combination of simple metrics to measure the navigability of document networks. These measures provide for low cost evaluation of the document networks formed by similarity measures and other link creation methods.

5.1 Introduction

Furnas (1997) has developed a theory of *effective view navigation* that is related to our goal of efficient navigation from relevant document to relevant document. Furnas details his theory in terms of two types of graphs: a logical graph and a view graph.[1] The logical graph represents how objects, such as documents, are truly connected to

[1]We will use the terms network and graph interchangeably. In each case, we are referring to directed graphs, which consist of nodes and directed edges. Each directed edge connects a source node to a target node.

each other. Furnas gives the web with its hyperlinks as an example of a logical graph. The view graph adds directed links to each node in the logical graph and represents the ways a user who is viewing the current node can immediately get to other nodes in the view. With find-similar, we are looking at ways to augment the logical graph and create a view graph that makes it easier for a user to find relevant documents.

To achieve effective view navigation, a system needs to be both *efficiently view traversable* (EVT) and *view navigable* (VN).

To be efficiently view traversable, Furnas requires two things. The first, EVT1, is that the views should be small, in other words, the out-degree of each node should be low when considering the view graph. The second, EVT2, is that the distance from each node to each other node on the view graph be short compared to the size of the overall structure.

Furnas' view navigability concerns itself with the "signage" aspects of a system. Links in the network need to provide good "residue" of the objects reachable via the link. Furnas' residue is similar to Pirolli's information scent (Pirolli, 2007). In other words, the user needs the link labeled in a manner that provides a form of lookahead. At the same time, the label must be small. Simply providing a listing of everything reachable via the link would provide good residue but would result in too large of a label.

We see Furnas' use of out-degree as an approximation of the user's cost to use the link. As such, while the links in Furnas' graphs are unweighted, we will weight each link in the network proportional to the time it takes a user to discover, evaluate, and travel a link.

One of the two measures of document navigability that we will define is based on the shortest paths between relevant documents. With regard to EVT2 (shortest paths), the question for information retrieval is not how easy is it to get from one document to another, but how easy is it to get from a *relevant* document to *other*

relevant documents. The searcher cares about the time to find relevant documents and not the time to travel between arbitrary documents. With a weighted document network, shortest paths now represent the optimal path for a user to follow between two documents.

A network with paths shorter than another network may actually be less navigable. For example, a randomly constructed network of low degree can have short paths between most nodes in the network. No user would be expected to navigate well in a random network.

Our other measure of network navigability will aim to capture the quality of the similarity measure given the neighborhood it creates for a node. Hierarchical navigation networks such as the Yahoo! or DMOZ directories of web sites are examples of the difficulty of providing good node residue to achieve Furnas' view navigability for large document collections. The links at the top of these hierarchies are broad descriptions of the content available and offer little help in selecting the correct links. While we agree with the need for good link labels, with respect to the network structure, the network should be locally navigable. We are interested in document networks linked primarily at a local level — document to document. A good similarity measure produces links from relevant documents to other relevant documents. A random network would do poorly on this measure of navigability.

We will use our two measures in combination to evaluate the navigability of a document network. When comparing two similarity methods, the better method should produce a network that is more navigable given both measures. We next discuss the two measures in detail.

5.1.1 Measures

Given a user's information need or search topic, a perfect similarity method for find-similar makes the topic's relevant documents most similar to each other. This is

a restatement of the cluster hypothesis (Jardine and van Rijsbergen, 1971). If a user finds a relevant document, and we have a "cluster hypothesis made true" similarity method, all a user needs to do is to request similar documents and the user will retrieve all of the relevant documents.

To measure the cluster hypothesis, Jardine and van Rijsbergen (1971) plotted the distributions of relevant pairs (R-R) and relevant and non-relevant pairs (R-NR) to visually determine the extent to which the cluster hypothesis was true. This same procedure was examined in more detail by van Rijsbergen and Sparck Jones (1973). Griffiths et al. (1997) replaced the visual inspection of the distributions with a measure of separation of the two distributions called the *overlap coefficient*.

Voorhees (1985) pointed out that the relative frequency of very similar R-NR pairs is reduced by the large number of R-NR pairs in comparison to the number of R-R pairs. As an alternative, Voorhees proposed the *nearest neighbor* test, which counted the number of relevant documents found in the N nearest neighbors of a relevant document. Voorhees set $N = 5$. Voorhees' test is equivalent to examining the precision at 5 for the ranked lists produced by using relevant documents as queries. In place of precision at 5, any other retrieval metric such as average precision could be used in a similar manner. Using average precision would result in the computation of a mean average precision (MAP) for each given topic where each relevant document for that topic acts as a query. Voorhees' methodology has an added benefit that it is a measure that is more closely mapped to user notions of distance and separation.

Diaz (2008, chap. 3) provides an overview and examination of cluster hypothesis research. In Section 3.3 of Diaz's work, he argues for Voorhees' measure because it can handle cases when the relevant documents form multiple clusters as opposed to the method of Jardine and van Rijsbergen (1971).

We use Voorhees' methodology to measure the local quality of the document network. For each relevant document, we measure the precision at rank N, where

$N = 5, 10, 20$ and also the average precision (AP) given the ranking of the document's neighbors formed by taking the weighted links as each neighbor's retrieval score.

One issue with P5, P10, or P20 compared to AP is that when the number of documents is less than the rank cutoff N, for precision at rank N, then the local metric can report a low precision when in fact all documents are easily reached from all other documents. For example, if there are 6 relevant documents, the maximum P10 is 0.6. As we will show, this limitation of P5, P10, and P20 does not appear to be a serious one on average.

The introduction of query-biased similarity makes it much easier to obtain a similarity method that performs well using Voorhees' methodology but which fails to cluster relevant documents well. For example, assume we have a query that has many relevant documents and a P5 of 1. If we query-bias the similarity until the query dominates over the given document, then the rankings for every relevant document will be nearly identical and also have a P5 of 1. Using Voorhees' methodology, we would declare the clustering performance to be excellent when in fact it could be very poor. Our imagined query may be high in precision but low in recall. Thus, all the relevant documents will be close to a few relevant documents but far away from the majority of relevant documents.

This was not a concern when Voorhees proposed her nearest neighbors measure. In her study, the documents were the queries. As such, each document was a point in the vector space and a measure like P5 does give a sense of how close the other relevant documents are to a given document. When we mix the query with the document, at some point we are only measuring how close the relevant documents are to a single point in space, that is, how close the documents are to the query.

Another potential problem with the above mentioned measures of the cluster hypothesis is that they fail to accommodate the triangle inequalities that make the cluster hypothesis so appealing. We want to reward a similarity measure for making it

easy to get from relevant document A to relevant document C by going first from A to relevant document B and then from B to C even if the similarity measure considers A and C to be dissimilar. To capture this feature of similarity and the value of navigating from document to document, we turn to a measure of the distance between documents measured on the network.

5.1.1.1 Document Networks

In a document network, the nodes represent documents in the collection and the edges represent a user's ability to traverse from a given document to another document via some user interface.

We aim to weight the links between documents in a manner that approximates the user's cost to find that link. Given a document-to-document similarity measure embodied by an implementation of find-similar or other user interface feature, for each document in a collection, we can compute a ranking of all other documents in that collection. While at best a crude approximation of user cost, we follow traditional information retrieval metrics and set a link's weight equal to its rank.

While the primary goal of the link weight is to represent the user's cost to navigate the link, another benefit of weighting links in this manner is similar to the benefit reaped by Voorhees for going to document ranks and away from the raw similarity score. The distance from a relevant document to another relevant document in a ranked list captures the issue with non-relevant documents also being similar to the relevant document. Just because two relevant documents are very similar given a similarity measure does not preclude many non-relevant documents being more similar to the document.

Using document rank as our distance also provides us with another benefit. If we assume that shortest paths between relevant documents avoid passing through non-relevant documents, then we can delete the non-relevant documents from the

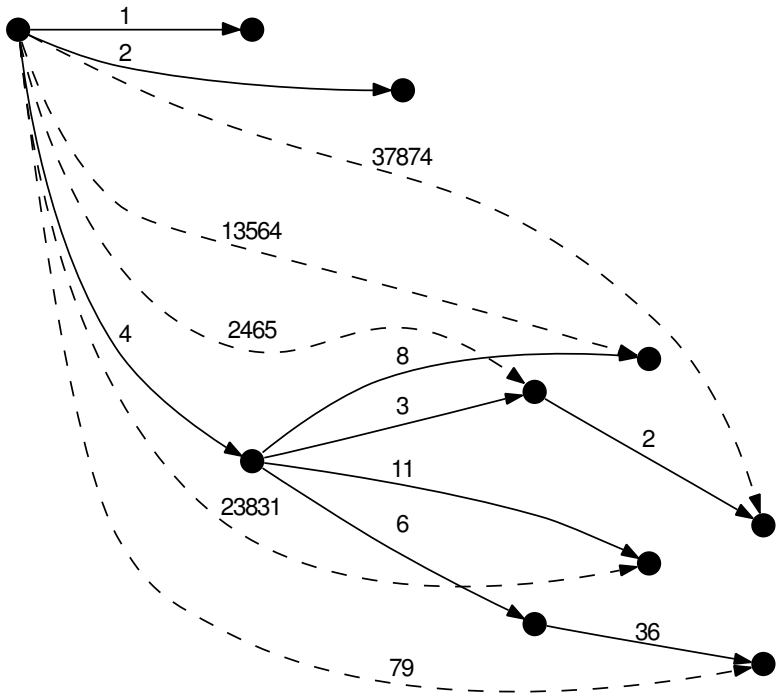


Figure 5.1. This figure is the same as Figure 1.5 except that rather than show the lists of ranked results, the initial query and the documents are shown as nodes. Edges are drawn between nodes and weighted by the rank of the document in the results list. The left uppermost node represents the initial query.

graph and obtain the same results for the shortest paths between all pairs of relevant documents. Deleting the non-relevant documents produces what we term a *relevant document network*. Find-similar is closely related to relevant document networks as shown in Figure 5.1.

Figures 5.2 and 5.3 show examples of relevant document networks. Figure 5.2 shows a representation of the relevant document network for TREC topic 335, "adoptive biological parents." The plotted distance between nodes is relative to how close they are given their link weights. Figure 5.3 shows an example of how query-biased similarity can result in a better clustering of relevant documents. Appendix B shows additional examples of relevant document networks.

We obtain a substantial computational savings by deleting the non-relevant documents to form a relevant document network. For the relevant document network, we

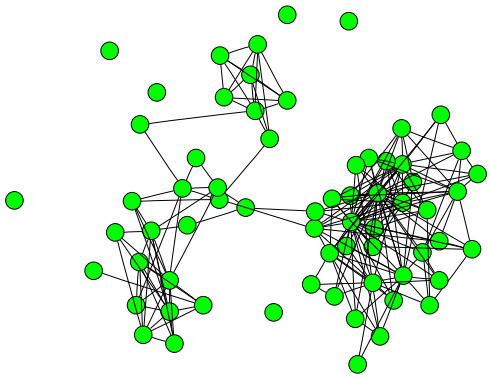
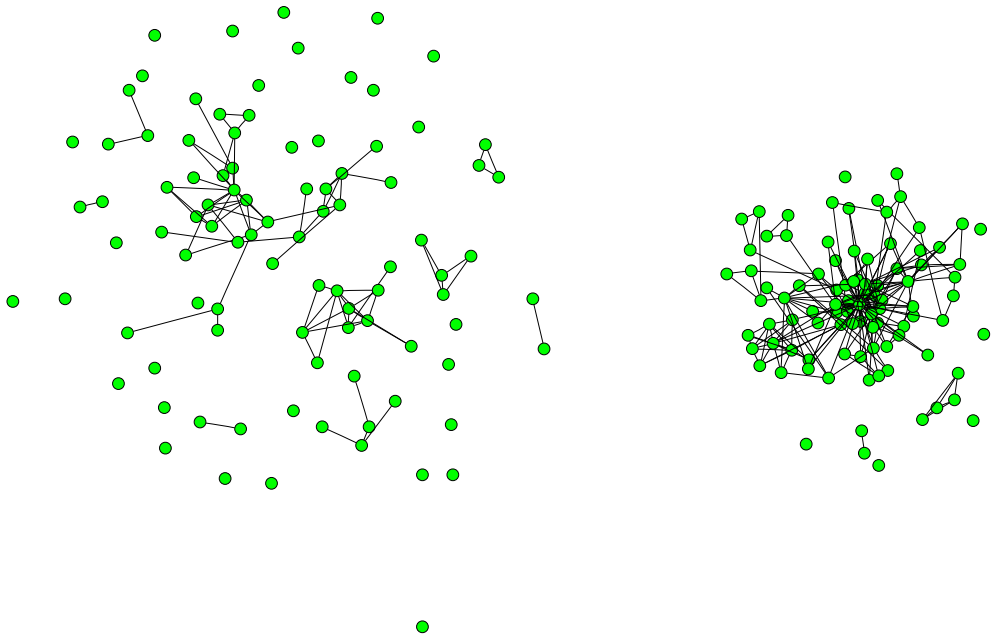


Figure 5.2. A simplified depiction of the relevant document network for TREC topic 335, "adoptive biological parents." Each node is a relevant document. Links are drawn between two documents if one the links between them has a link weight of 5 or less. The actual relevant document network is a weighted, directed graph. It appears that for this topic there are two or three clusters of relevant documents. Note that several documents appear unconnected to the larger clusters of documents. These outliers may be difficult to reach using document feedback techniques such as find-similar.

only need to calculate similarity information for the relevant documents rather than for all documents.

Because the cost of calculating the pairwise similarity for all documents is so expensive, we might be tempted to approximate the full document network by only considering the top 1000 or so documents found to be similar to a topic's query. The assumption here is that all the documents of concern will fall in the top ranked results. This approximation fails to examine relevant documents found at ranks greater than rank 1000. For difficult topics, this would mean ignoring most relevant documents. Also, limiting our analysis to the top 1000 documents prevents us from seeing the power of similarity information to make relevant documents found at ranks greater than 1000 easier to find. With the relevant document network we can examine all relevant documents as past measures of the cluster hypothesis have done.

If non-relevant documents were to be on the shortest paths to relevant documents, relevant documents should have non-relevant documents as common neighbors. The cluster hypothesis says that relevant documents share something in common to make



(a) (b)

Figure 5.3. Simplified depictions of the relevant document networks for TREC topic 337, "viral hepatitis." The network on the left (a) uses regular similarity while the network on the right (b) uses query-biased similarity, which better clusters relevant documents. The documents are closer in figure (b) because they are higher ranked in each other's ranked lists. As with Figure 5.2, links are drawn between two documents when one of the links between the pair has a weight of 5 or less. The actual relevant document network is a weighted, directed graph.

	Non-Relevant		
	10	20	100
Minimum	0.000	0.000	0.003
1st Quartile	0.018	0.024	0.039
Median	0.036	0.044	0.069
Mean	0.057	0.066	0.091
3rd Quartile	0.066	0.080	0.119
Maximum	0.593	0.717	0.543

Table 5.1. The average overlap coefficient among the top $N = 10, 20, 100$ ranked non-relevant documents in the nearest neighbors of relevant documents for TREC topics 301-450. For example, the mean fraction of non-relevant documents in common is 0.066 or 6.6% for the top 20 highest ranked non-relevant documents.

them more similar to each other. In contrast, there is a limitless set of reasons that a document is non-relevant.

As a test of the extent to which non-relevant documents are common neighbors of relevant documents, we took the TREC topics 301-450 and we measured the overlap of the first N non-relevant documents occurring in the ranked lists produced by using a relevant document as a query. The document collection for topics 301-450 is composed of newswire and government documents.

Our measure of overlap was the overlap coefficient:

$$\frac{|A \cap B|}{\min(|A|, |B|)}$$

where A is the set of N highest ranked non-relevant documents for relevant document A and similarly for document B. For each topic we computed the average overlap over all pairs of non-relevant documents and then computed summary statistics over all 150 topics. Table 5.1 shows that the amount of overlap is quite small with the mean overlap for $N = 20$ being 0.066 or 6.6% and three quarters of the topics have an overlap of 8% or less. Thus it appears that non-relevant documents play a role more akin to "noise" than as potentially useful stepping stones between relevant documents.

The assumption that a user will not navigate through non-relevant documents does not hold for document networks such as the web. On the web, links have a mixture of types. Some links go directly to other content rich pages while other links may go to a navigational page. Many navigation pages are not likely to be considered relevant pages in and of themselves. Imagine for example a web site that provides a find-similar link from each content page. The find-similar page is for navigational purposes and may link to a relevant page, but is not in itself a relevant page. By requiring paths to only go through relevant pages, for a similarity measure such as the web graph, we could cut off valid paths.

The relevant document network should only be used in situations where the document network is formed using a feedback-like technique such as find-similar. The relevant document network provides a reasonable upper bound on the shortest path where there is little sense in a user searching for relevant documents starting from a non-relevant document. While a non-relevant document may bridge two relevant documents, how would a user know how to decide between the good non-relevant documents and the bad ones? In a feedback situation, the user would be forced to "lie" to the system and judge a non-relevant document relevant.

5.1.1.2 Shortest Paths Measure: Normalized Mean Reciprocal Distance

Given a weighted document network, we can efficiently compute shortest paths using Dijkstra's shortest paths algorithm or the Floyd-Warshall all pairs shortest paths (APSP) algorithm (Cormen et al., 2001). We used the Boost Graph Library's implementation of the Floyd-Warshall APSP algorithm (Siek et al., 2001).

Distance on our weighted document networks represents the number of documents a user would need to examine by reading link labels such as document titles and summaries before reaching the other document. Other weighting schemes could

approximate the individual costs of discovering, evaluating, and traversing links more closely.

For our measure of global navigability, we use the *global efficiency* measure of Latora and Marchiori (2001). This metric computes on a per topic basis, for each relevant document the normalized, mean reciprocal distance (nMRD) of all other relevant documents. The normalized, mean reciprocal distance of relevant document R_i is calculated as:

$$nMRD(R_i) = \frac{1}{Z(|R|-1)} \sum_{R_j \in R, j \neq i} \frac{1}{S(R_i, R_j)} \tag{5.1}$$

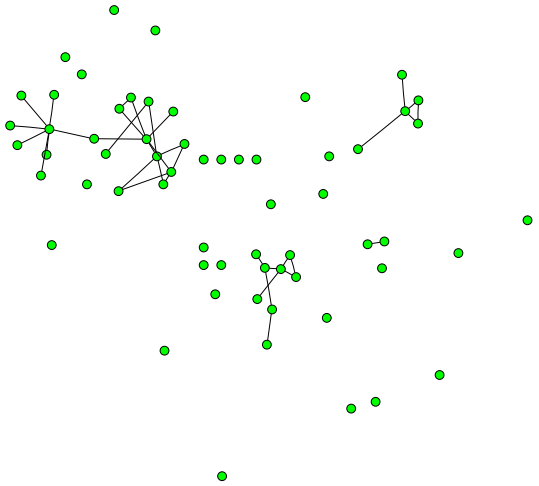
where R is the topic's set of relevant documents, $|R|$ is the number of relevant documents, $S(R_i, R_j)$ is the shortest path distance from R_i to R_j, and Z is the normalization factor. For each topic, we average the nMRD over all the known relevant documents, and finally we average over all topics to produce a final metric. This metric varies from 0 to 1 with 1 being the most efficient or navigable network possible. Figure 5.4 shows examples of relevant document networks with varying nMRD values.

Figure 5.5 and Figure 5.6 show how to compute the optimal MRD for a given number of relevant documents. This method works for our definition of document networks, but would need to be modified for other networks.

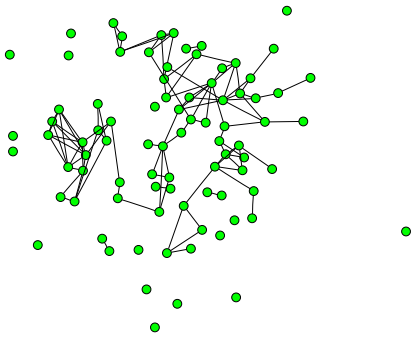
5.1.1.3 Previous Shortest Paths Measures

Our proposed shortest paths metric is based on our previous experience using similar techniques to measure the cluster hypothesis. In our first work (Smucker and Allan, 2006), we briefly introduced relevant document networks and measured the average all pairs shortest paths distance for TREC topics 301-450.

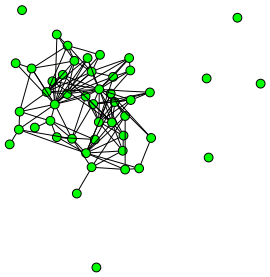
The document collection for topics 301-450 consists of TREC volumes 4 and 5 minus the Congressional Record. This is a 1.85 GB, heterogeneous collection that



nMRD = 0.06
(Topic 323: Literary/Journalistic Plagiarism, 61 relevant documents)



nMRD = 0.19
(Topic 387: radioactive waste, 85 relevant documents)



nMRD = 0.45
(Topic 351: Falkland petroleum exploration, 48 relevant documents)

Figure 5.4. Examples of relevant document networks with varying nMRD values. The document-to-document similarity is "regular" with no weight given to the query/topic.

```
optimalMRD = 0
nodesRemaining = numRelDocs - 1
nodesThisLevel = 1
level = 1
while nodesRemaining > 0 do
  if level > 1 then
    nodesThisLevel = nodesThisLevel * 2
  end if
  if nodesThisLevel > nodesRemaining then
    nodeThisLevel = nodesRemaining
  end if
  optimalMRD = optimalMRD + (nodesThisLevel * (1 / level))
  nodesRemaining = nodesRemaining - nodesThisLevel
  level = level + 1
end while
optimalMRD = optimalMRD / (numRelDocs - 1)
```

Figure 5.5. This figure accompanies Figure 5.6 and shows how to compute the optimal mean reciprocal distance (MRD, see Equation 5.1) for a given number of relevant documents, numRelDocs.

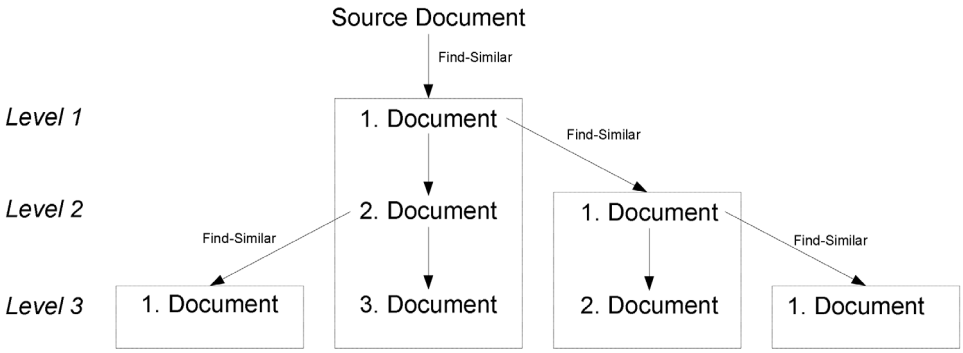


Figure 5.6. This figure accompanies Figure 5.5 and shows graphically the optimal paths from a source document to 7 other relevant documents (8 relevant documents in total including the source document). Each link has a weight of 1. One document can be reached in a distance of 1, two documents can be reached at a distance of 2, and four documents at a distance of 3. Each box represents a list of documents similar to a given document.

contains 528,155 documents from the Financial Times Limited, the Federal Register, the Foreign Broadcast Information Service, and the Los Angeles Times.

For each relevant document, we computed a maximum likelihood estimated model and truncated this model to the 50 most probable terms. We then used this model as a query to rank the documents in the collection and construct a relevant document network for each topic. The mean average all pairs shortest paths (APSP) distance was 3507 for regular similarity and 381 for query-biased similarity. Here we implemented query-biased similarity using the proximity method discussed in section 3.2.3. We reported the median distance which was 70.8 for regular similarity and 33.0 for query-biased similarity. This demonstrated that query-biased similarity can better cluster relevant documents.

We reported the median because the distribution of distances was highly skewed. In hindsight, reporting only the median distance was a mistake. The median will report that a similarity measure clusters well when it actually is only making a subset of documents close to all documents and not producing a good clustering of documents. The mean distance captures the overall degree of clustering better by giving equal weight to each relevant document. This is why for our global metric, nMRD, we will use the average and not the median.

Another mistake we made was to use the distance between relevant documents directly. At some point, the distance to some documents is so large that they can be considered to be "out of reach." What difference does it make if the distance to a document is ten thousand or ten million? Using the distances directly results in a cluster metric whereby a similarity measure that moves relevant documents closer but still keeps them out of reach unfairly gets rewarded. In addition, using the distance directly prevents good comparison across collections for the same reason. Documents that are out of reach in a small collection or a large collection are still too far away and the cluster metric should not be sensitive to the difference. This is why we have

proposed using the mean reciprocal distance (inverse distance). Even so, we may want to place a maximum distance past which a document is considered to be at infinity.

In a previous work, (Smucker and Allan, 2007a), we proposed a variant of Latora and Marchiori's measure that was not normalized. While using an unnormalized metric on the same data set will produce comparable results for different similarity measures, without normalization one cannot compare across topics or different data sets.

5.2 Experiments

Building on the work of Chapter 3, we look at 3 types of document-to-document similarity. The first is *regular* similarity where we treat the entire document as a query. The second is the *query-biased* similarity as presented in Section 3.2.3. The third is a query-biased similarity where a model of the query is mixed with a model of the entire document. We next discuss this third form of similarity in more detail.

A possible reason for the better performance of the query-biased similarity in Chapter 3 is that the query terms are given a significant fraction of the probability mass in the computed query model. Recall that for each occurrence of a query term in a document, we place a window over that term. The text that falls within the various windows makes up our final query. The windows included 5 words before and after the query terms. Assuming non-overlapping windows, the query terms found in the document would be given a combined mass of $1/11 = 0.09$, and likely more than 10% of the mass with some windows overlapping (words in overlapping windows are only counted once).

An alternative way to implement query-biased similarity is to mix the query model with the document model:

$$P(w|M_B) = \lambda P(w|Q) + (1 - \lambda)P(w|D) \tag{5.2}$$

where $P(w|M_B)$ is the probability of the word w in the biased model and $P(w|Q)$ and $P(w|D)$ are the maximum likelihood estimated models of the query and document respectively that are linearly combined with λ determining how much influence the query should have versus the document.

While different than Tombros' query sensitive similarity measure (QSSM) (Tombros, 2002), which was a measure for vector space retrieval, the above formulation for language modeling retrieval captures the nature of QSSM. This similarity measure is also similar in nature to the measures used in query-biased clustering (Eguchi, 1999; Iwayama, 2000). Both Eguchi (1999) and Iwayama (2000) increase the weight of query terms in the documents before clustering.

By comparing the "window" version of query-biased similarity (Section 3.2.3) to the "mix" version (Equation 5.2), we can see if the context captured by the windows holds an advantage over simply mixing the query with the document.

When λ in Equation 5.2 is 0, this "mix" version of query-biased similarity becomes the "regular" similarity of Section 3.2.3. In addition, we can replace the MLE model of the document in Equation 5.2 with any model of the document, $P(w|M_D)$:

$$P(w|M_B) = \lambda P(w|Q) + (1 - \lambda)P(w|M_D) \tag{5.3}$$

Besides testing the "window" version of query-biased similarity as it was presented in Section 3.2.3, we will also take the query-biased model of the document that it produces and mix this model with the MLE model of the query.

We investigated λ with values of 0, 0.1, 0.25, 0.5, 0.75, and 0.9.

We used the same documents and 150 topics from TREC 6, 7, and 8 as in Chapter 3 (see Section 3.2.6).

For each of the types of similarity, we measured the global measure nMRD as well as precision at rank 5 (P5), P10, P20, and average precision (AP) as local measures of clustering.

5.3 Results and Discussion

Table 5.2 shows the results for our experiments. The *context* represents the number of words included before and after the presence of a query term. The query-biased similarity of Chapter 3 has a context size of 5. The λ is the λ of Equation 5.3. Thus, regular similarity is represented by a context of "wholeDoc" and a λ of 0.

Figure 5.7 shows the data of Table 5.2 with the global measure of navigability, nMRD, plotted against the four measures of local navigability. As can be seen, when averaged over the set of queries, there appears to be little difference between the local measures.

Most significantly, we can see from Figure 5.7 that while in general a higher score for one of the local measures of navigability implies a higher score for the global measure (nMRD), this is not always the case. There are numerous runs where the document-to-document similarity measure produced results with high local measures but with low nMRD. For example, Voorhees' P5 measure has a value of 0.53 for the runs with a context size of 5 words and the λ values of 0.1 and 0.5, but when $\lambda = 0.1$, nMRD = 0.32 and when $\lambda = 0.5$, nMRD drops to 0.22.

The local measures of navigability are unable to detect the global navigability of a network. This lower performance for the high values of λ is likely the result of a lack of diversity in the similarity lists across documents. In other words, giving the query too much weight produces a ranking of similar documents that is more or less the same for all documents.

The original regular similarity from Chapter 3 (context of wholeDoc and $\lambda = 0$) has a nMRD of 0.20. The query-biased similarity of Chapter 3 (context of 5 and

Context	λ	P5	P10	P20	AP	nMRD
1	0.00	0.42	0.37	0.31	0.19	0.26
2	0.00	0.44	0.38	0.32	0.19	0.29
5	0.00	0.44	0.38	0.31	0.18	0.29
10	0.00	0.42	0.36	0.29	0.16	0.28
15	0.00	0.40	0.34	0.28	0.15	0.27
wholeDoc	0.00	0.32	0.27	0.22	0.11	0.20
1	0.10	0.47	0.41	0.34	0.21	0.27
2	0.10	0.50	0.44	0.37	0.23	0.30
5	0.10	0.53	0.46	0.39	0.24	0.32
10	0.10	0.53	0.47	0.39	0.25	0.32
15	0.10	0.53	0.47	0.39	0.25	0.32
wholeDoc	0.10	0.51	0.45	0.38	0.24	0.30
1	0.25	0.50	0.44	0.37	0.23	0.25
2	0.25	0.54	0.47	0.40	0.25	0.28
5	0.25	0.57	0.50	0.42	0.27	0.29
10	0.25	0.57	0.51	0.43	0.27	0.29
15	0.25	0.58	0.51	0.43	0.27	0.29
wholeDoc	0.25	0.56	0.50	0.42	0.27	0.27
1	0.50	0.50	0.45	0.38	0.24	0.21
2	0.50	0.52	0.46	0.39	0.24	0.22
5	0.50	0.53	0.47	0.40	0.25	0.22
10	0.50	0.52	0.47	0.40	0.25	0.21
15	0.50	0.52	0.47	0.40	0.25	0.21
wholeDoc	0.50	0.51	0.46	0.39	0.24	0.20
1	0.75	0.48	0.43	0.37	0.23	0.18
2	0.75	0.48	0.43	0.37	0.23	0.18
5	0.75	0.47	0.43	0.37	0.23	0.17
10	0.75	0.47	0.43	0.37	0.22	0.17
15	0.75	0.47	0.43	0.37	0.22	0.17
wholeDoc	0.75	0.46	0.42	0.36	0.22	0.17
1	0.90	0.45	0.41	0.36	0.22	0.16
2	0.90	0.45	0.41	0.36	0.22	0.16
5	0.90	0.45	0.41	0.35	0.22	0.16
10	0.90	0.45	0.41	0.35	0.21	0.16
15	0.90	0.45	0.41	0.35	0.21	0.15
wholeDoc	0.90	0.45	0.41	0.35	0.21	0.15

Table 5.2. Global and local measures of navigability. Each row is a different similarity measure. Local measures are precision at rank 5 (P5), P10, P20, and average precision (AP). The normalized mean reciprocal distance (nMRD) is a global measure. *Context* refers to the size of the "window" in Section 5.2. When the context is *wholeDoc*, the entire document is used. The λ is the same as in Equation 5.3. In Chapter 3, regular similarity used the whole document and query-biased similarity used a context of 5 and both set $\lambda = 0$.

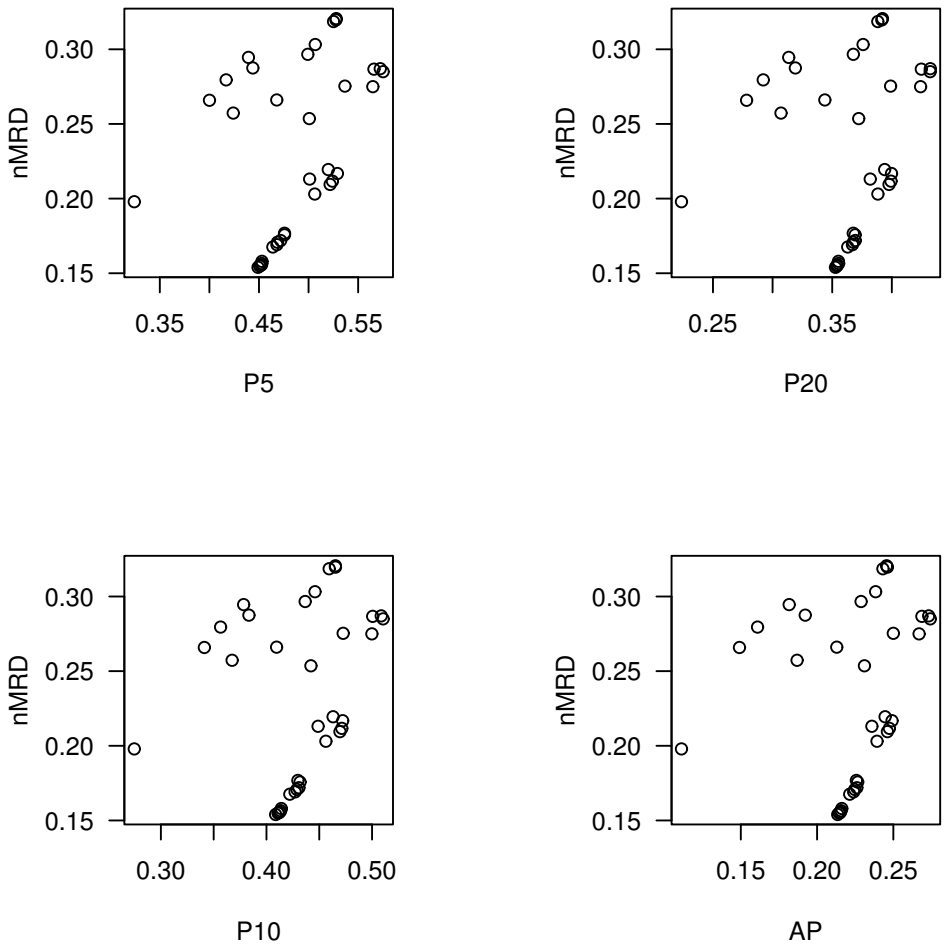


Figure 5.7. Global navigability versus local navigability. The normalized mean reciprocal distance (nMRD) is plotted vs. the precision at rank 5 (P5), P20, P10, and mean average precision (MAP). Each point is the average of 150 topics. There is one point for each of the similarity measures in Table 5.2.

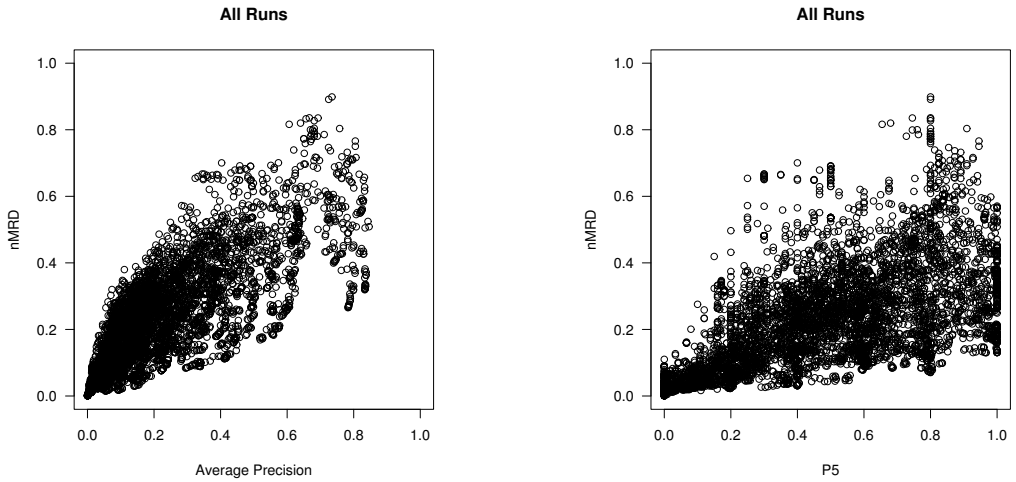


Figure 5.8. The normalized mean reciprocal distance (nMRD) vs. average precision (AP) on the left and precision at rank 5 (P5) on the right. Each point in the charts is a topic for each of the similarity measures of Table 5.2.

$\lambda = 0$) has a nMRD of 0.29. Here we see that the query-biased similarity has an absolute gain of 9% in navigability and a 45% relative gain. The better navigability of query-biased similarity mirrors its better find-similar performance.

Setting $\lambda = 0.1$ produced the best nMRD scores for all context sizes. Using a reduced context of 5, 10, or 15 words produced slightly better results than using the whole document (nMRD of 0.32 versus 0.30). It appears that there is some value to the window form of query-biased similarity although the majority of the benefit seems to come from giving the original query enough, but not too much weight.

While Figure 5.7 shows that AP and P5 are very similar when averaged over many topics, we can see in Figure 5.8 that P5 as a metric can be maxed out by topics. We would recommend the use of AP over precision at rank N measures since AP performs similarly to P5 and does not appear to have the same risk of being maxed out by topics.

Figure 5.9 shows the similarity methods of Chapter 3, regular similarity and query-biased similarity, compared by plotting nMRD vs. P5 for all 150 topics. There is

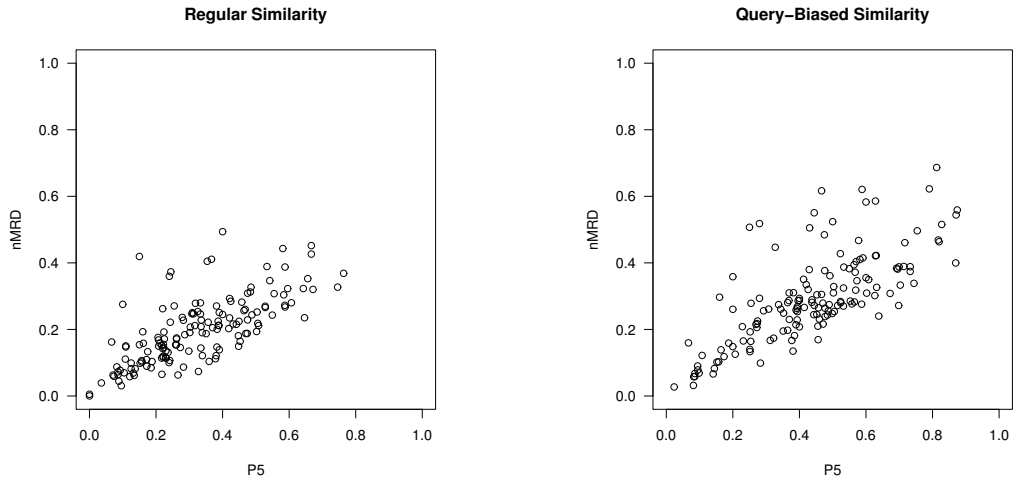


Figure 5.9. Regular similarity (left) navigability compared to query-biased similarity (right) by plotting the normalized mean reciprocal distance (nMRD) vs. precision at rank 5 (P5) for each similarity measure. Each point in the charts is a topic. Query-biased similarity uses a context size of 5 words before and after query terms while regular similarity uses the entire document as a query. Neither similarity is mixed with the original query.

clearly a general trend with increasing local navigability resulting in greater global navigability. Query-biased similarity's largest benefit appears to be the significant increase in global navigability for a handful of topics. These topics have increases in global navigability that go far beyond any increase in local navigability.

Figure 5.10 shows how navigability is affected as the λ in Equation 5.3 is increased from 0 to 0.9 when the whole document is used as the context. While some topics maintain high levels of nMRD, most topics' global navigability (nMRD) suffers as the query weight becomes too high. As evidenced by the move to the fixed values of P5 (0, 0.2, ..., 1.0), the documents all begin to look like the query. When the similarity lists for each document are essentially the same, it is clear that traversing the similarity space from relevant document will not work well. This also shows why both a global and local measure of navigability are needed to measure the cluster hypothesis. If

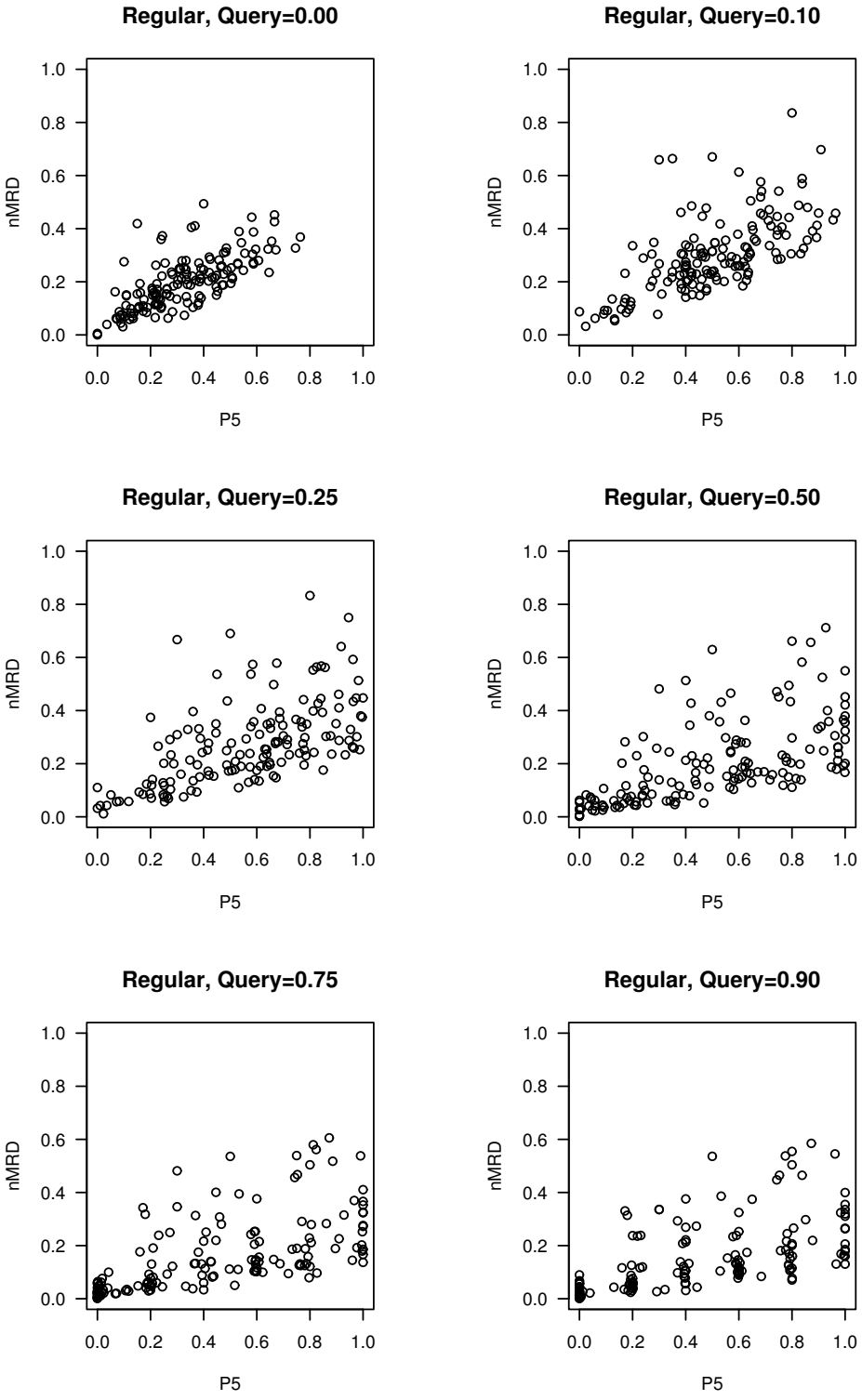


Figure 5.10. This figure shows how navigability changes as the query weight is increased for similarity where the whole document is used as the context.

either measure is used without the other, it is possible to develop similarity measures that appear to cluster relevant documents well, but do not cluster well in actuality.

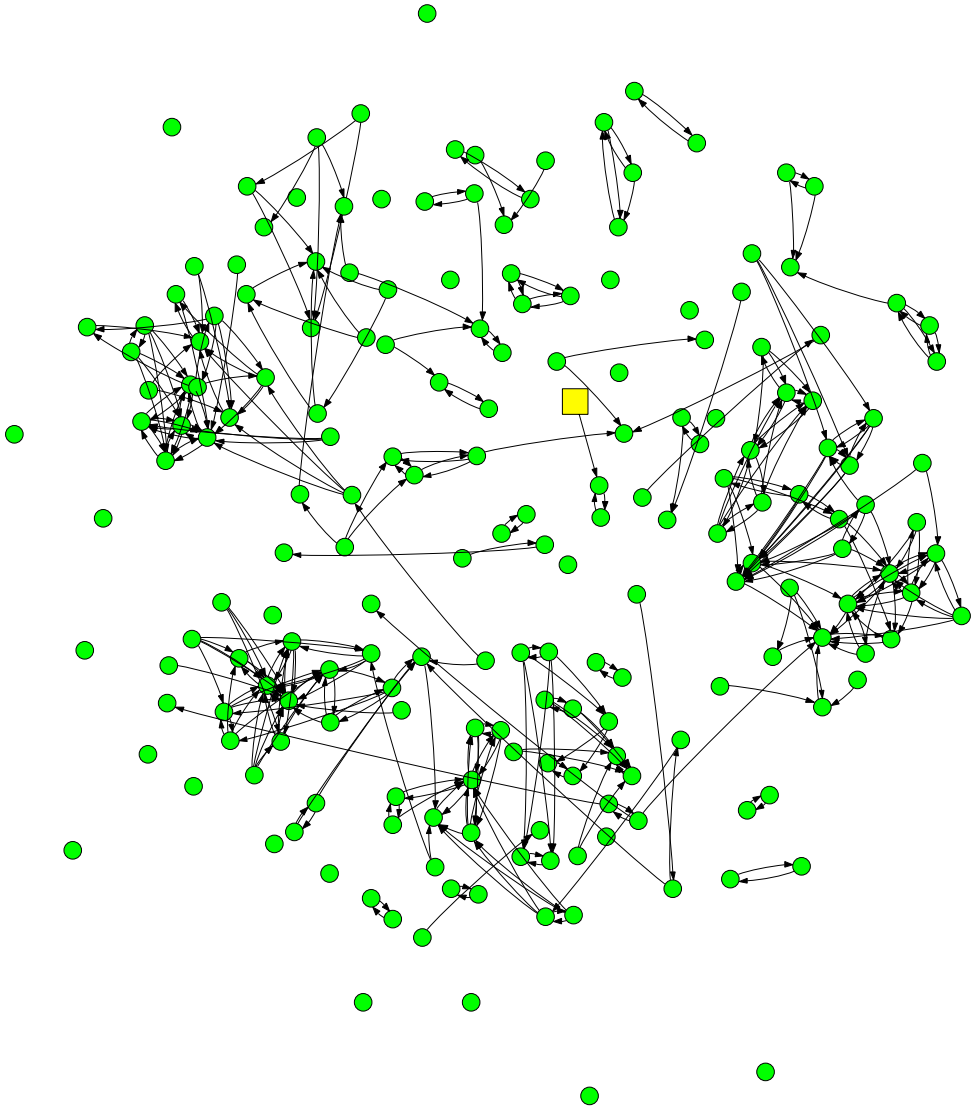
Figures 5.11, 5.12, and 5.13 show an example of how too much query-biasing of the similarity can hurt the global navigability of relevant document networks. Figure 5.13 shows that with a high λ of 0.5, the majority of documents find the same one or two documents to be the documents most similar to themselves. Too much query-biasing produces a similarity with all documents having very similar lists of similar documents. From each document it becomes easy to navigate to a few relevant documents, but once at these documents, there is no easy way to navigate to other relevant documents. While a local measure of similarity can stay high under these circumstances, the ability to traverse from document to document is hurt. In cases where the original query performs very poorly, too much query-biasing can hurt both local and global navigability.

5.4 Conclusion

In this chapter we presented a method of measuring the navigability of a document network using a global and local measure. The nodes in the network represent the documents in the collection and the directed links represent the ability of a user to traverse from a source document to a target document. The weight of a link is set proportional to the user's cost to find, evaluate, and traverse the link. One measure captures a local and the other a global quality of the network. The local quality of a network can be measured as follows. For each relevant document, we rank a document's neighbors by their link weights and measuring the average precision of this ranking. The measure of local quality is the mean average precision for the relevant documents. The global measure captures the cost to follow the shortest path, navigating from a relevant document to another relevant document. For each relevant document, we measure the normalized mean reciprocal distance (nMRD)

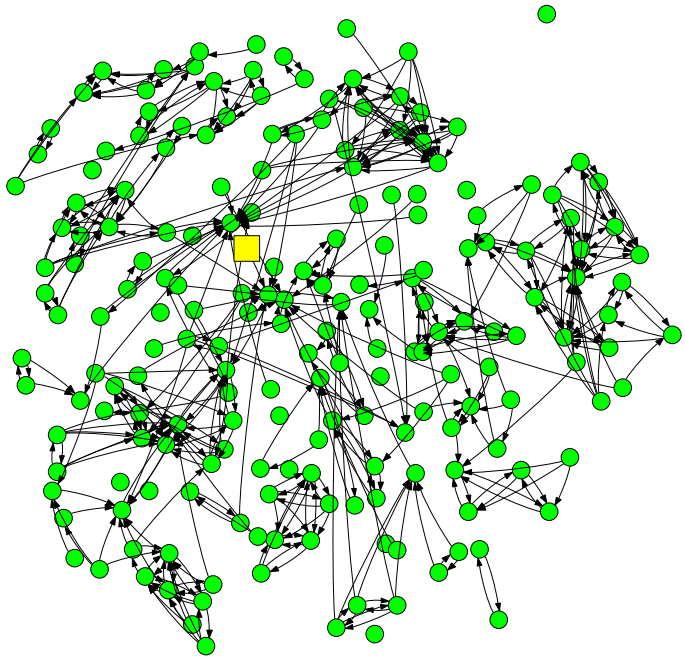
to all other relevant documents. The overall measure is the average of these mean reciprocal distances. Together, these two measures give us a good understanding of the navigability of a document network and allow us to design similarity methods that construct more navigable networks.

We examined the document-to-document similarity methods of Chapter 3 and showed that query-biased similarity is better able to cluster relevant documents. Our result that query-biased similarity better clusters relevant documents echoes the results of Tombros' work on query sensitive similarity (Tombros and van Rijsbergen, 2001; Tombros, 2002). We also showed that a local metric of the cluster hypothesis, such as precision at rank 5 (P5), is not sufficient. A similarity measure can score well on a local metric but perform poorly on the global measure. To measure the navigability of document networks both a global and local measure should be used.



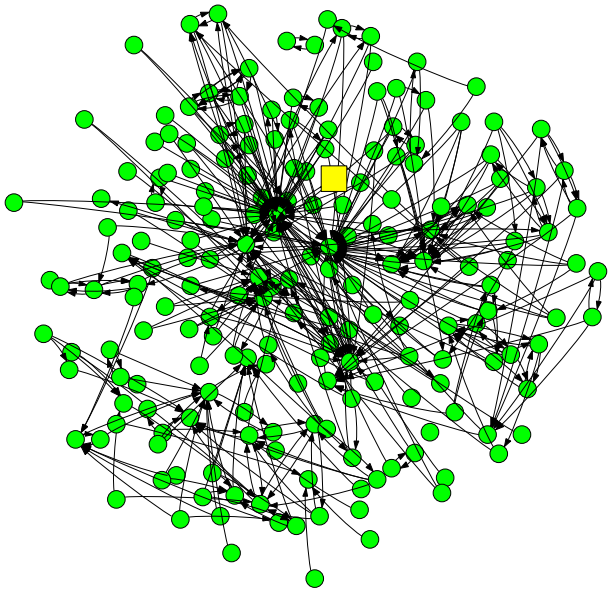
Regular similarity with $\lambda = 0$, nMRD = 0.12, P5 = 0.38, AP = 0.05

Figure 5.11. This figure in combination with Figure 5.12 and Figure 5.13 shows how too much query-biasing of the similarity can hurt the global navigability of Topic 304: "Endangered Species (Mammals)". The square, yellow node represents a query likelihood retrieval using the topic's title as a query. Nodes represent relevant documents. A link is drawn from a node to another node if the target node is within the top 5 most similar documents of the source node. This figure represents regular similarity with no query-biasing.

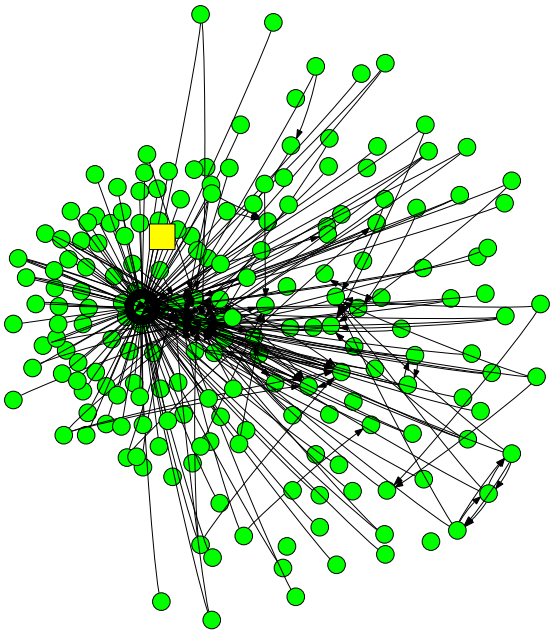


Regular similarity with $\lambda = 0.1$, nMRD = 0.17, P5 = 0.48, AP = 0.12

Figure 5.12. This figure in combination with Figure 5.11 and Figure 5.13 shows how too much query-biasing of the similarity can hurt the global navigability of Topic 304. Here we see that a little query-biasing ($\lambda = 0.1$) can greatly increase both local and global measures of navigability. The precision at rank 5 (P5) and average precision (AP) are local measures of navigability. The normalized mean reciprocal distance (nMRD) is a global measure of navigability.



Regular similarity with $\lambda = 0.25$, nMRD = 0.15, P5 = 0.44, AP = 0.14



Regular similarity with $\lambda = 0.5$, nMRD = 0.09, P5 = 0.27, AP = 0.12

Figure 5.13. This figure in combination with Figure 5.11 and Figure 5.12 shows how too much query-biasing of the similarity can hurt the global navigability of Topic 304. Once $\lambda = 0.5$, the majority of documents place the same one or two documents at the top of their lists of similar documents. It has become easy to navigate to these documents that are very similar to the query, but these documents do not provide good paths to the other relevant documents. While AP with $\lambda = 0.5$ stays the same as with $\lambda = 0.1$, the global navigability as measured by nMRD has decreased from 0.17 to 0.09 as λ increased from 0.1 to 0.5.

CHAPTER 6

USING FIND-SIMILAR ON THE WEB TO CREATE SHORTCUTS TO RELEVANT WEB PAGES

Browsing by similarity is a search tactic familiar to most people but one that the web unevenly supports. Successful navigation from a relevant web page to other relevant pages depends on the page linking to other relevant pages. In this chapter, we examine the navigability of the web as well as the navigability of the web when augmented with similarity links provided by find-similar. We find that the addition of 10 similarity links to a web page will significantly increase the navigability of the web.

6.1 Introduction

After a long and tiring search, a user finally finds a relevant web page. While the page is relevant, it does not fully satisfy the user's need. How should the user proceed? If the page provides links to other pages, the user can follow those links. Alternatively, the user could follow links automatically produced by a tool that examines the page's content and provides links to similar pages.

A user attempting to navigate from a relevant document to other relevant documents on the web is faced with at least two challenges. First, the web page's author may create hyperlinks with goals other than pointing the reader to related material. Second, the web is dynamic and many web pages fail to ever link to newer pages.

As applied to the web, find-similar aims to create a more navigable network by adding additional links to the existing document network that consists of web pages

Figure 6.1. Mockup of a possible find-similar web browser add-on. Find-similar provides a list of similar web pages in the left pane given the web page currently viewed in the right page. The results shown are a subset of the results produced by Yahoo! for the query "Center for Intelligent Information Retrieval."

and hyperlinks. The find-similar links are specifically aimed at finding related material and will also stay up to date as the web grows since they would be most likely be dynamically produced by a search engine that itself stays up to date with the web.

There are many ways that find-similar could be made available to a user via a web browser. One possible interface provides find-similar capabilities in a side pane as shown in Figure 6.1. In this mockup, the user triggers find-similar using a single button and gets a simple listing of similar web pages. This "single button" interface could also be created as a toolbar that produces a list of similar web pages either in the left browser pane, the current web viewing pane, or in a new browser tab.

We are concerned with the navigability of the web both on its own and with the addition of a find-similar like tool. We first establish the degree to which the web is navigable, which as we discussed in Chapter 5, is the same as talking about the

degree to which the well-known cluster hypothesis is true on the web. The web itself represents a type of document-to-document similarity measure where the similarity measure is the distance to navigate from one document to another using hyperlinks.

After investigating the extent to which the cluster hypothesis is true on the web graph, we then attempt to improve the navigability of the web with the automatic addition of links to similar web documents.

We will show that:

- Relevant documents are either within distance 5 of another relevant page or are as likely to be reached at greater distances as non-relevant documents.
- The automatic addition of content similarity hyperlinks can significantly increase the number of relevant documents reachable from a given relevant document.
- The addition of 10 similarity links produces an absolute gain in global navigability of 13.8% while at the same time increasing the local navigability of the web.

6.2 Methods and Materials

In place of working with the actual web, we used the wt10g TREC web collection. This collection consists of 1,692,096 web pages that were carefully selected from a larger collection (VLC2 or wt100g) of 18,571,671 documents to "ensure a high proportion of inter-server links" (Hawking and Craswell, 2005). Soboroff (2002) has shown the wt10g collection to have structural characteristics similar to the web. No images or other multimedia content is included in the collection. We constructed the web graph using the wt10g out_links file. We stemmed the collection using the Krovetz stemmer and used an in-house list of 418 stop words (see Appendix A).

To compute the document-to-document content similarity, we used the regular similarity as defined in Section 3.2.3. This method creates a maximum likelihood estimated model of each document. This method truncates each model to consist of only the document's 50 most probable terms. Using this model, the method measures the similarity of the other documents using KL-divergence and uses Dirichlet prior smoothing with its parameter set to 1500. We used the Lemur toolkit for our experiments.

We used the TREC 2001 web ad-hoc topics numbered 501-550. Each topic defines a set of relevant documents in the wt10g collection. We do not use the topics' titles or descriptions in any way.

6.3 Experiments

We conduct two experiments. In the first experiment, we measure the distribution of the shortest path distance from each relevant document to all other relevant documents on the web graph and the web graph augmented with similarity links. In the second experiment, we utilize the navigability metrics of Chapter 5 to characterize the navigability of the networks.

6.3.1 Distribution of Relevant Documents on the Web Graph

In this experiment, we compared the web graph with two augmented versions of the graph. For each topic, we augmented the graph by adding 10 out-links to each relevant document. In the first case, we added links to the 10 most content similar documents. This case corresponds to our envisioned browser plug-in that provides a list of the 10 web pages most similar to the current page. In the second case, we added 10 random links. This case allows us to make sure that the mere addition of links does not make relevant documents closer to each other on the web graph.

For each topic, we measured the shortest path distance from each relevant document to all other documents. Traversing a link or one "hop" is a distance of 1. This experiment gives us a simple understanding of the distribution of relevant documents with a distance metric common to other studies (Albert et al., 1999; Broder et al., 2000; Menczer, 2004).

We did not add any out-links to the non-relevant documents. We think that users would only utilize a feature providing these similarity links when they are looking for pages similar to a current relevant web page. In effect, we say the links do not exist on non-relevant pages because we do not believe that users will utilize the feature on non-relevant pages. A result of not adding links to non-relevant documents is that our measured distance from relevant document to relevant document is an upper bound on the shortest path length. Augmenting non-relevant documents with additional links could only have shortened the path lengths. On the other hand, if we had augmented all documents with additional out-links, the non-relevant documents would be closer than we report.

We computed overall averages by first averaging the measurements for a topic's relevant documents and then averaging all the topics.

6.3.1.1 Results and Discussion

Figure 6.2 shows the distribution of relevant and non-relevant documents as a function of their web graph distance from relevant documents. The "Web Only" distribution of relevant documents shows a clear bimodal shape peaking at distances of 4 and 15. The distribution of non-relevant documents is unimodal peaking around 17. The relevant documents reached at distances greater than 6 are most likely reached simply as a result of the interconnectedness of the web and are no easier to reach than non-relevant documents. The peak of relevant documents at distances less

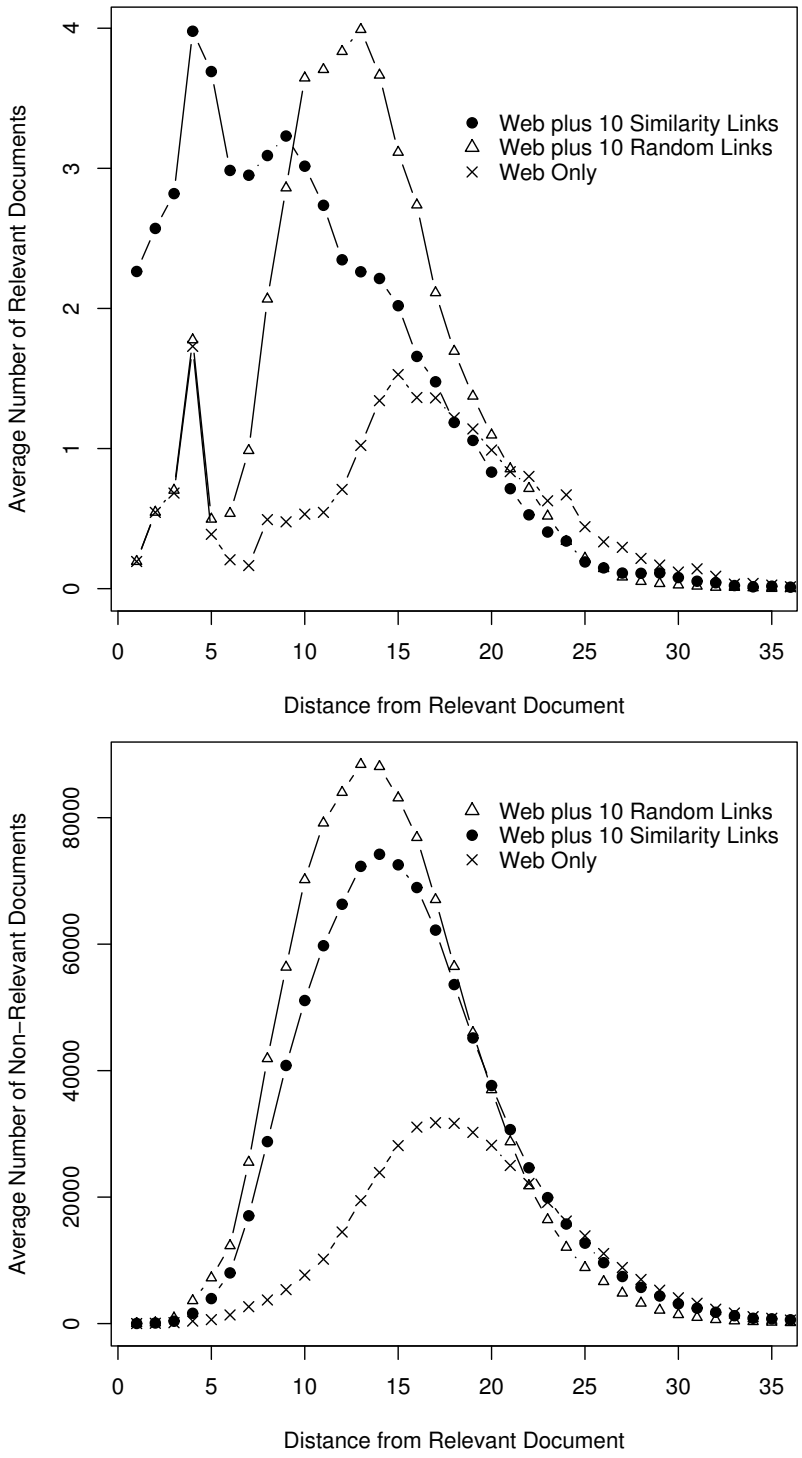


Figure 6.2. The distance of relevant and non-relevant documents from relevant documents.

than 6 demonstrates that the cluster hypothesis is true on the web graph at least for some relevant documents.

The addition of 10 content similar links brings a significant number of relevant documents closer to the average relevant document. A similar peaking at a distance of 4 shows that the relevant documents reached using the similarity links are likely independent of the ones being found using the existing links. With existing web links, on average 0.19 relevant documents are within distance 1 from a relevant document. With the addition of 10 content similarity links, an average of 2.26 relevant documents are distance 1 from a relevant document. The performance boost continues for the larger distances. At distance 4, the web-only distribution has an average 1.73 relevant documents and the web plus similarity links has 3.98.

The addition of 10 random links results in no significant gains at distance 5 or less. The additional links do bring non-relevant documents closer but do not help at short distances. Thus, the gains obtained by similarity linking appear to result from relevant documents being more similar to each other than simply from increasing the graph connectivity.

Our results are in line with Menczer's prediction that one would be able to infer relevance of a page at a maximum of around 4 to 5 hops (Menczer, 2004). Vassilvitskii and Brill (2006) used distance on the web graph to perform a reranking of search results given that relevant documents link to other relevant documents. For each top ranked search result, they performed a limited breadth-first search and found that searching to a distance of 4 resulted in the best performance. Our results explain their finding by showing that relevant documents are found within a distance of 5 or are as likely to be found as non-relevant documents.

The topmost chart in Figure 6.3 shows the growth of the number of non-relevant documents as a function of the distance from the average relevant document. This chart shows the number of documents the user would have to examine via a blind,

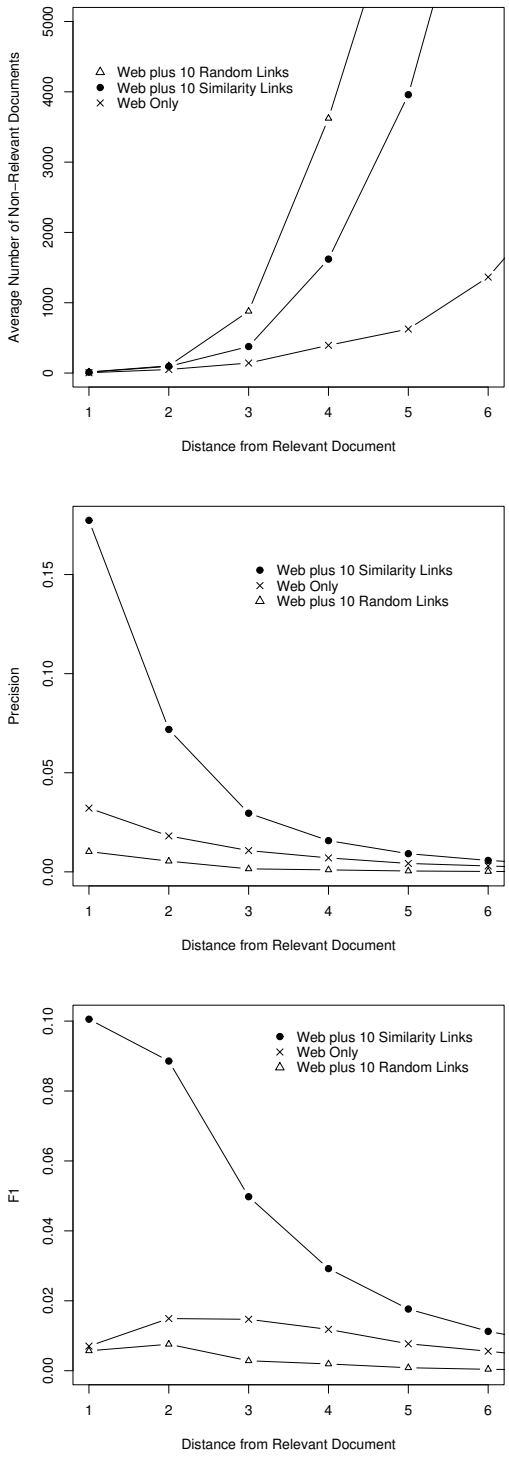


Figure 6.3. These three charts present another look at the value of adding similarity links to web pages. All three charts show a measure as a function of distance from the average relevant document. The top chart is a closeup of the bottom chart of Figure 6.2 showing the rapid growth of non-relevant documents. The middle chart shows the precision of documents and the bottom chart shows the F measure with $\beta = 1$, which is known as F1.

breadth-first search that ignores relevant documents. While we believe users will navigate from relevant document to relevant document and will thus be able avoid examining this many documents, another way to view the utility of the similarity links is to look at the fraction of pages that are relevant at different distances from a relevant document. We define this fraction to be the precision at a distance d. This precision is the number of relevant documents within distance d divided by the total number of documents within distance d:

$$P_d = \frac{|R_{i \leq d}|}{|R_{i \leq d}| + |N_{i \leq d}|} \tag{6.1}$$

where $|R_{i \leq d}|$ is defined to be (and likewise for $|N_{i \leq d}|$):

$$|R_{i \leq d}| = \sum_{i=1}^{d} |R_i| \tag{6.2}$$

with $|R_i|$ being the number of relevant documents at distance i. We utilize the shortest path from relevant document to relevant document to determine the distance at which a relevant document is found.

The middle chart of Figure 6.3 shows the overall P_d for the 50 topics. The overall P_d is the mean of the 50 topics' P_d. Each topic's P_d is the mean of the topic's relevant documents' P_d. Here we see the significant gain that adding similarity links has to the precision.

We are also interested in the recall of relevant documents. Recall is the fraction of known relevant documents found. The F measure gives us a single measure that captures both precision and recall. The F measure is:

$$F = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

where P is precision and R is recall.

The bottom chart of Figure 6.3 shows the F measure with $\beta = 1$, which is also known as F1. F1 places an equal weight on precision and recall. In this chart we again see the significant gains attainable from adding 10 similarity links to web pages.

While this experiment shows that the cluster hypothesis is true to a limited extent on the web and that find-similar can enhance the cluster hypothesis, it does not give us any measure of how much we improved the clustering of relevant documents. Our next experiment gives us measures of navigability.

6.3.2 Navigability of the Web with and without Find-Similar

In this experiment, we applied the local and global measures of navigability from Chapter 5 to four document networks: the three networks of the first experiment plus a truncated relevant document network.

In Chapter 5, we gave links a weight equal to the rank of its target document in the list of documents similar to the link's source document. For the web graph, we take the links on a web page and treat them as a ranking of the other web pages. We give all links a weight equal to the number of links on a page plus 1 divided by 2, i.e. the average ranking. For example, each link on a page with 9 links will get a weight of 5. We do this because it is not easy to determine the within page ranking of the links given our dataset.

The relevant document network is formed with the same content similarity, regular similarity, as described in the previous section (and in Section 3.2.3). Rather than create a complete network, we only included links that had a weight of 10 or less, where the weight is the rank of the document in the list of documents similar to the given document. This allows us to measure the navigability of the 10 similarity links minus the web links. Recall that the relevant document network contains only relevant documents and thus links to non-relevant documents are not included, but

	Web	Web + 10 Rand. Links	10 Sim. Links	Web + 10 Sim. Links
nMRD	0.021	0.028	0.134	0.166
AP	0.002	0.001	0.051	0.046
P5	0.027	0.011	0.269	0.225

Table 6.1. Average global and local navigability for 4 document networks. The four networks are the web alone, the web plus 10 random links added to relevant documents, the relevant document network constructed with 10 similarity links, and the web plus 10 similarity links added to the relevant documents. The normalized mean reciprocal distance (nMRD) is a global measure while the average precision (AP) and the precision at rank 5 (P5) are local measures. The measures are the average of the per-topic measures of the 50 topics of the TREC 2001 web track.

the remaining links' weights reflect the existence of the non-relevant documents in the similarity ranking.

The combination of the web and random network is the same as in the first experiment. Here we add ten random outgoing links to each relevant document. We give the 10 random links weights of 1, 2, ..., and 10, respectively.

6.3.2.1 Results and Discussion

Table 6.1 shows the results. This table shows the mean of the global and local measures across the 50 topics for each measure. The normalized mean reciprocal distance (nMRD) is the global measure while the average precision (AP) and precision at rank 5 (P5) are local measures. Here AP should be preferred to P5 because the web pages links were weighted in a way that could fail to allow the P5 measure to properly measure their contribution or lack thereof to the local navigability.

The web alone does not appear to provide good navigability either locally or globally. Treating nMRD as a measure of efficiency, the web alone is only 2.1% efficient while the 10 link similarity network is 13.4% efficient.

Combining the web and the similarity links produces the highest nMRD of 0.166. The web plus random links has a nMRD of 0.028. Compared to the web plus random links, adding similarity links to the web increases the web's global navigability by

0.138, or a 13.8% absolute gain in efficiency. The local navigability of the web also increases with similarity links, but the local navigability of the web plus similarity links is slightly lower than the similarity links alone.

The web plus the random links provides little boost in navigability. As such, the similarity links help navigability by making relevant documents closer to each other rather than simply by adding more paths to the web. This matches the findings of the first experiment where we also found that adding 10 content similarity links to web pages brings relevant documents closer to each other.

Figure 6.4 shows the per topic nMRD and AP for the 50 topics for the web graph, the similarity network, and the combination of the web and the similarity links. From these charts we can see that the web is boosting the global navigability of the topics that have lower local navigability.

6.4 Conclusion

We conducted two experiments that investigated the ability of find-similar to increase the navigability of the web. With our first experiment, we found a bimodal distribution for the distance of relevant documents to each other on the web graph. Relevant documents are within a distance of 5 of each other or as likely to be reached as non-relevant documents. The automatic addition of 10 content similarity links brings significantly more relevant documents close to each other. In our second experiment, we utilized the measures of navigability from Chapter 5 to measure the navigability of the web alone and with the addition of similarity links. We found that the addition of 10 similarity links produced an absolute gain in global navigability of 13.8% while at the same time increasing the local navigability of the web. Augmenting the web with similarity links should aid the searcher attempting to navigate from a relevant document to other relevant documents.

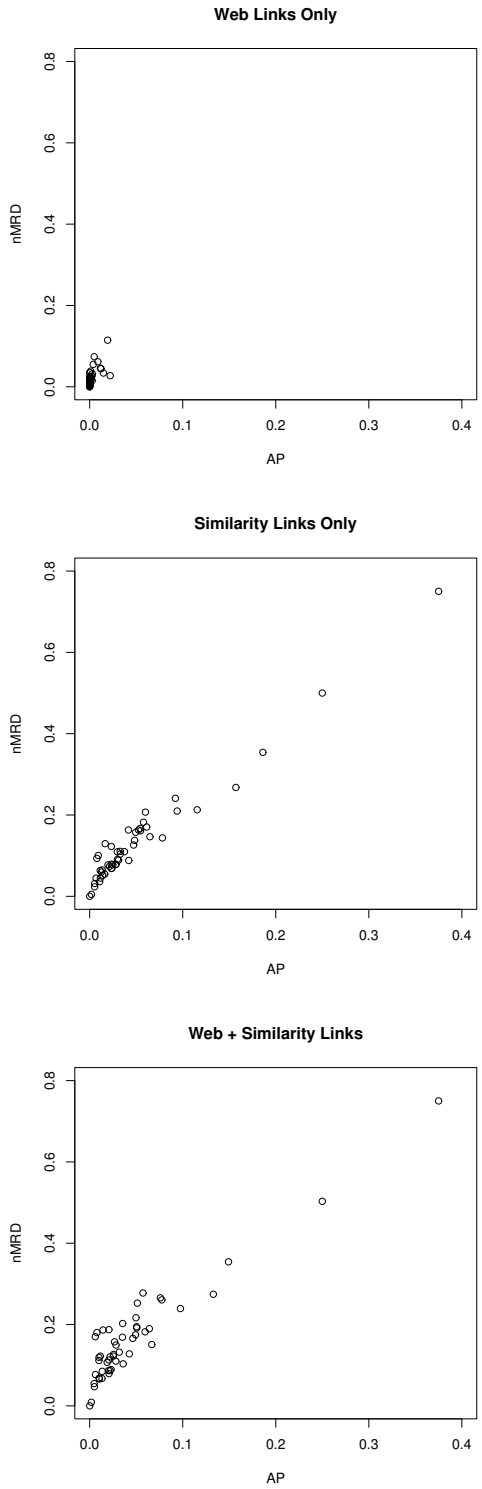


Figure 6.4. Global (nMRD) and local navigability (AP) for the 50 topics of the TREC 2001 web track. The normalized mean reciprocal distance (nMRD) and the average precision (AP) are shown for the web, similarity, and web plus similarity networks.

CHAPTER 7

CONCLUSION AND FUTURE WORK

It has long been known that user feedback or additional user input can substantially boost search quality. The problem faced by information retrieval researchers has been to create interaction mechanisms that are both powerful and that users will adopt. In this dissertation, we have focused on the study and improvement of an already adopted interaction mechanism: find-similar.

Find-similar is an example of the simple interaction mechanisms that we believe are the route to continued improvements in retrieval quality. These mechanisms can be integrated with the existing search interface paradigm of a user entering a text query and receiving a ranked list of results.

The first part of the dissertation focused on measuring the potential of find-similar using simulation. By simulating user behavior given hypothetical user interfaces, we discovered that find-similar has the potential to powerfully improve retrieval quality for users. In the second part of the dissertation, we used network analysis to study find-similar. We developed measures of navigability that allowed us to study find-similar and document-to-document similarity measures in a well-defined manner that is independent of user interfaces and simulated behavior. In combination, the simulation and network analysis experiments showed how a simple search tool like find-similar can help users better find relevant documents.

In summary, our key contributions were:

1. We showed that find-similar has significant potential to improve retrieval performance. Using simulation, we found that find-similar can match multiple item

relevance feedback. While perceived as an easier to use version of relevance feedback, find-similar can be as powerful as a traditionally styled multiple item relevance feedback.

2. Find-similar is not without its issues. We showed that if users apply find-similar to already good results, they are likely to degrade the retrieval performance.

3. We also showed that an interface should help the user avoid reexamining documents to maximize retrieval quality with find-similar.

4. We showed that find-similar can compensate for poor retrieval quality. In effect, find-similar can be added to existing systems and make them more robust to variations in users and queries.

5. Poorer quality retrieval systems are helped more by find-similar, and find-similar boosts performance more on easier search topics than on difficult topics.

6. We found that a query-biased document-to-document similarity outperforms a similarity measure that ignores the query.

7. We demonstrated that both a local and global measure of the cluster hypothesis should be used. Our global measure, borrowed from the field of network analysis, is new to the field of information retrieval.

8. Our measures of the cluster hypothesis allow us to quantify the value of similarity measures like query-biased similarity without ad-hoc assumptions of user behavior found in our simulation studies.

9. We found the cluster hypothesis to be true to a limited extent on the graph of the World Wide Web. Using both local and global measures, we showed that adding 10 content similarity links to web pages should make the web significantly more navigable.

7.1 Future Work

In this section, we conclude the dissertation by discussing two avenues of future work. First, we outline work to extend find-similar to problems requiring the novelty of information. Second, we discuss the further study of different types of similarity.

7.1.1 Novelty

One aspect of find-similar that we have not examined in this dissertation is the need for some users to find a set of unique answers or information nuggets. For example, many different news agencies will report on the same event. In many cases, there will be several documents that say the same information but simply say it in a different way. Under our present evaluation methodology, which is standard for the IR tasks we examined, such repetition of information is ignored. In other words, the documents or information found by the user for these tasks is not required to be novel compared to the information already found by the user as part of their search.

The TREC complex, interactive question answering (ciQA) track's evaluation does require information to be novel. As part of the TREC 2007 ciQA track, we have begun studying how best to create an interactive IR system for users searching for answers to complex questions (Smucker et al., 2008). A good information retrieval system should inherently be a good question answering system. Our work to date has been to establish a baseline for human performance with a standard interactive IR system that only allows the user to reformulate the query. Shown in Figure 7.1 is the interface we created for our ciQA experiments. Future work will involve adding additional interaction mechanisms to the interface to boost performance.

Extending find-similar to ciQA would involve two significant changes. The first would be to have find-similar work at the answer level. Users would be able to ask for answers similar to a given answer. Answers are typically on the order of one sentence long compared to the many sentence long documents used in this dissertation. We

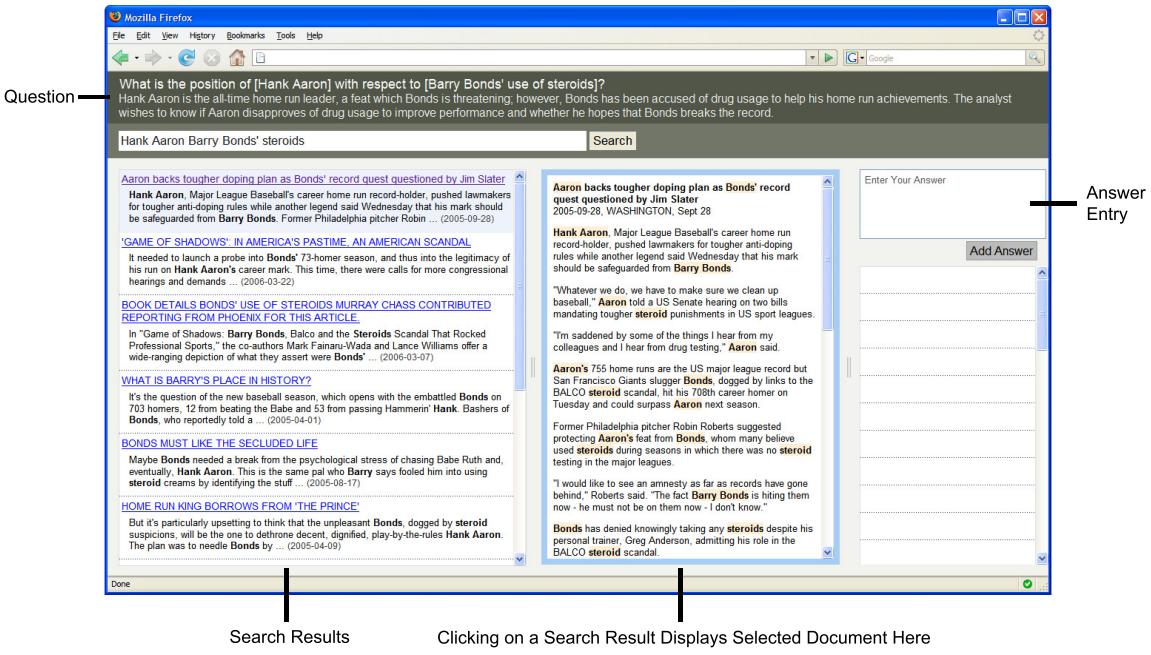


Figure 7.1. A screenshot of the interactive, IR system we used for the TREC ciQA experiments. At the top of the interface, we presented the question and a search textbox. The area below the question and search box consisted of three vertically oriented panes. The left pane showed search results. Clicking on a result showed the respective document in the middle pane. The right hand pane provided a textbox allowing the user to enter and save an answer to the question. A list of the user's saved answers appeared below the answer entry box.

would likely utilize existing techniques for measuring the similarity of sentences (Murdock and Croft, 2005; Metzler et al., 2005; Murdock, 2006; Balasubramanian et al., 2007).

The second extension would be to address the novelty requirements of ciQA, which penalize systems for repeating information nuggets in their list of answers. We would first aim to test a version of find-similar that performs sentence retrieval and removes sentences too similar to the given answer. We could apply the techniques of Chapter 5 with the modification that only the novel answers would be used and not all answers in the calculation of the navigability metrics. If the filtering is successful, the navigability of the "answer networks" should be improved.

7.1.2 Multiple Types of Similarity

As we have seen in this dissertation, different types of similarity have better navigability than other types of similarity. We have measured navigability for one goal: finding relevant documents. Users may have multiple goals or needs, and for some goals, different similarity measures will produce more navigable networks. We envision extending find-similar to allow for multiple types of similarity within a single button or link. As mentioned in Chapter 1, CiteSeer provides different types of similarity for each document in its repository. Our study would focus on the user interface and usability issues of presenting multiple types of similarity to the user. For example, while there may be better types of similarity to use to find a piece of information, certain paths can make more sense to users (Teevan et al., 2004). We would be particularly interested in which types of similarity make the most sense for users. In this dissertation we have taken the stance that the primary notion of similarity that matters is the one that makes document networks cluster relevant documents better. Even though we have a local measure of navigability to capture users' need to have an *information scent* (Pirolli, 2007), some similarity measures may offer a better scent for users in the field.

APPENDIX A

STOPWORDS

a, about, above, according, across, after, afterwards, again, against, albeit, all, almost, alone, along, already, also, although, always, am, among, amongst, an, and, another, any, anybody, anyhow, anyone, anything, anyway, anywhere, apart, are, around, as, at, av

be, became, because, become, becomes, becoming, been, before, beforehand, behind, being, below, beside, besides, between, beyond, both, but, by

can, cannot , canst, certain, cf, choose, contrariwise, cos, could, cu

day, do, does, doesn't, doing, dost, doth, double, down, dual, during

each, either, else, elsewhere, enough, et, etc, even, ever, every, everybody, everyone, everything, everywhere, except, excepted, excepting, exception, exclude, excluding, exclusive

far, farther, farthest, few, ff, first, for, formerly, forth, forward, from, front, further, furthermore, furthest

get, go

had, halves, hardly, has, hast, hath, have, he, hence, henceforth, her, here, hereabouts, hereafter, hereby, herein, hereto, hereupon, hers, herself, him, himself, hindmost, his, hither, hitherto, how, however, howsoever

i, ie, if, in, inasmuch, inc, include, included, including, indeed, indoors, inside, insomuch, instead, into, inward, inwards, is, it, its, itself

just

kind, kg, km

last, latter, latterly, less, lest, let, like, little, ltd

many, may, maybe, me, meantime, meanwhile, might, moreover, most, mostly, more, mr, mrs, ms, much, must, my, myself

namely, need, neither, never, nevertheless, next, no, nobody, none, nonetheless, noone, nope, nor, not, nothing, notwithstanding, now, nowadays, nowhere

of, off, often, ok, on, once, one, only, onto, or, other, others, otherwise, ought, our, ours, ourselves, out, outside, over, own

per, perhaps, plenty, provide

quite

rather, really, round

said, sake, same, sang, save, saw, see, seeing, seem, seemed, seeming, seems, seen, seldom, selves, sent, several, shalt, she, should, shown, sideways, since, slept, slew, slung, slunk, smote, so, some, somebody, somehow, someone, something, sometime, sometimes, somewhat, somewhere, spake, spat, spoke, spoken, sprang, sprung, stave, staves, still, such, supposing

than, that, the, thee, their, them, themselves, then, thence, thenceforth, there, thereabout, thereabouts, thereafter, thereby, therefore, therein, thereof, thereon, thereto, thereupon, these, they, this, those, thou, though, thrice, through, throughout, thru, thus, thy, thyself, till, to, together, too, toward, towards

ugh, unable, under, underneath, unless, unlike, until, up, upon, upward, upwards, us, use, used, using

very, via, vs

want, was, we, week, well, were, what, whatever, whatsoever, when, whence, whenever, whensoever, where, whereabouts, whereafter, whereas, whereat, whereby, wherefore, wherefrom, wherein, whereinto, whereof, whereon, wheresoever, whereto, whereunto, whereupon, wherever, wherewith, whether, whew, which, whichever, whichsoever, while, whilst, whither, who, whoa, whoever, whole, whom, whomever, whomso-

ever, whose, whosoever, why, will, wilt, with, within, without, worse, worst, would, wow

ye, yet, year, yippee, you, your, yours, yourself, yourselves

APPENDIX B

RELEVANT DOCUMENT NETWORKS

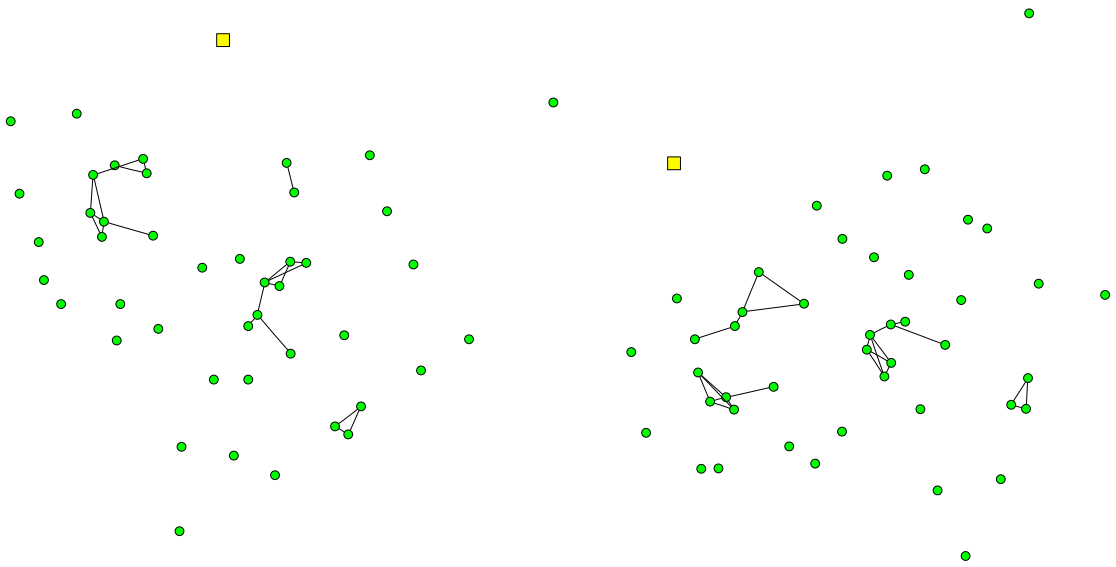
This appendix shows 30 examples of relevant document networks randomly selected from the 150 topics of TREC 6, 7, and 8. Included with each figure is the precision at rank 5 (P5), a local measure of navigability, and the normalized mean reciprocal distance (nMRD), which is a global measure of navigability. The square node in yellow represents the initial results as computed with a query likelihood retrieval. In each figure, the left network is formed by regular similarity and the right network is formed by query-biased similarity as described in Chapter 3. The topic number is given as well as the topic's title.

To create the drawings, we take the complete, directed relevant document network and convert it to an undirected network by giving the single link between documents a weight equal to the lowest of the two links between documents in the directed network. While all links are used for the layout, the only links drawn are those with a weight of 5 or less. In other words, a visible edge means that at least one of the documents in a pair of documents was in the top 5 similar documents for the other document.

Regular
nMRD = 0.16, P5 = 0.07

Query-Biased
nMRD = 0.16, P5 = 0.07

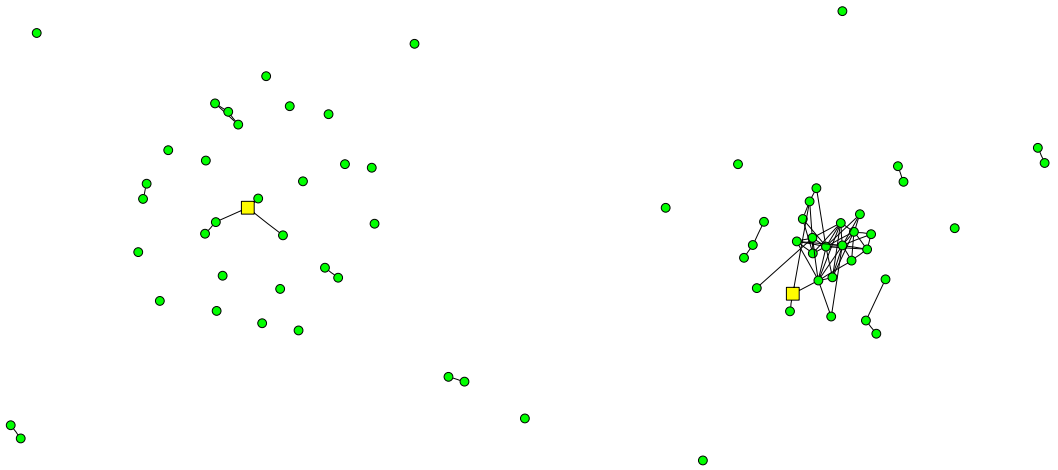
Figure B.1. Topic 309: Rap and Crime



Regular
nMRD = 0.07, P5 = 0.13

Query-Biased
nMRD = 0.07, P5 = 0.14

Figure B.2. Topic 314: Marine Vegetation



Regular
nMRD = 0.04, P5 = 0.09

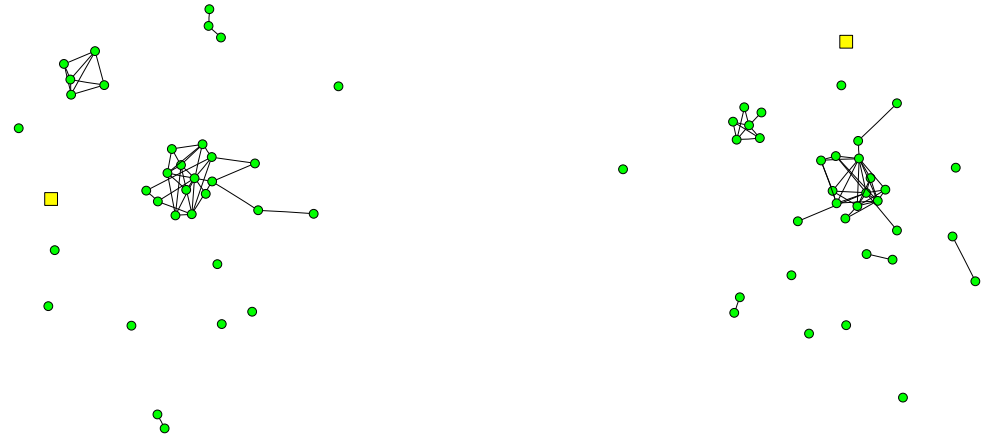
Query-Biased
nMRD = 0.29, P5 = 0.40

Figure B.3. Topic 316: Polygamy Polyandry Polygyny

Regular
nMRD = 0.49, P5 = 0.40

Query-Biased
nMRD = 0.62, P5 = 0.47

Figure B.4. Topic 320: Undersea Fiber Optic Cable



Regular
nMRD = 0.21, P5 = 0.34

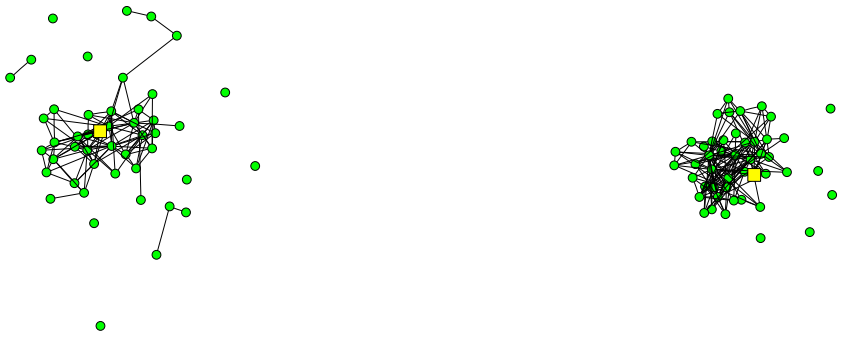
Query-Biased
nMRD = 0.20, P5 = 0.36

Figure B.5. Topic 322: International Art Crime

Regular
nMRD = 0.21, P5 = 0.32

Query-Biased
nMRD = 0.36, P5 = 0.49

Figure B.6. Topic 325: Cult Lifestyles



Regular
nMRD = 0.32, P5 = 0.60

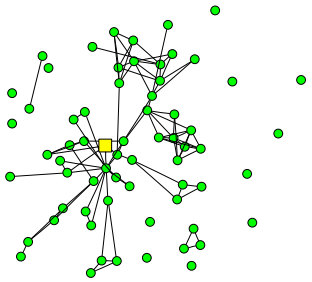
Query-Biased
nMRD = 0.47, P5 = 0.82

Figure B.7. Topic 326: Ferry Sinkings

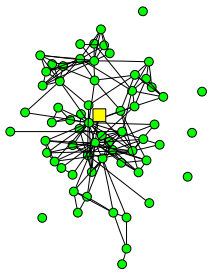
Regular
nMRD = 0.16, P5 = 0.21

Query-Biased
nMRD = 0.23, P5 = 0.27

Figure B.8. Topic 329: Mexican Air Pollution

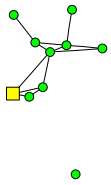


Regular
nMRD = 0.24, P5 = 0.40

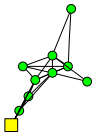


Query-Biased
nMRD = 0.42, P5 = 0.63

Figure B.9. Topic 333: Antibiotics Bacteria Disease

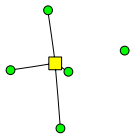


Regular
nMRD = 0.37, P5 = 0.24



Query-Biased
nMRD = 0.55, P5 = 0.44

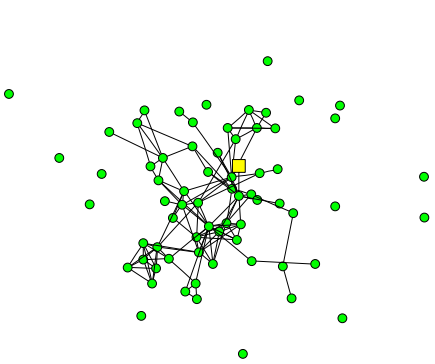
Figure B.10. Topic 334: Export Controls Cryptography



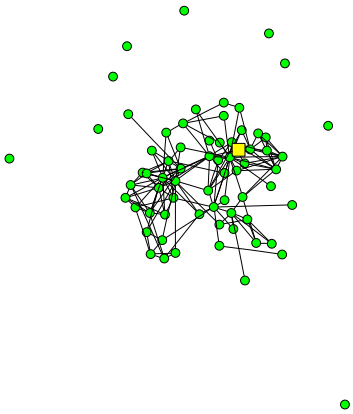
Regular
nMRD = 0.00, P5 = 0.00

Query-Biased
nMRD = 0.52, P5 = 0.28

Figure B.11. Topic 348: Agoraphobia

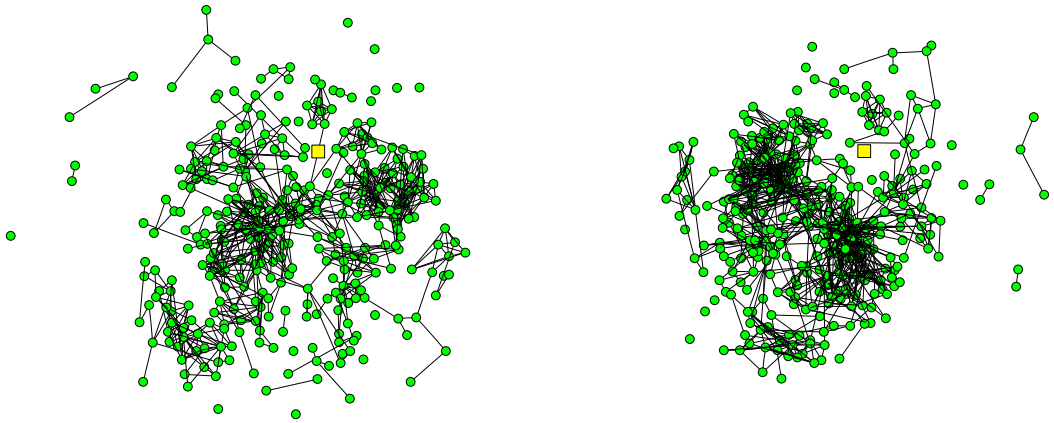


Regular
nMRD = 0.33, P5 = 0.49



Query-Biased
nMRD = 0.37, P5 = 0.57

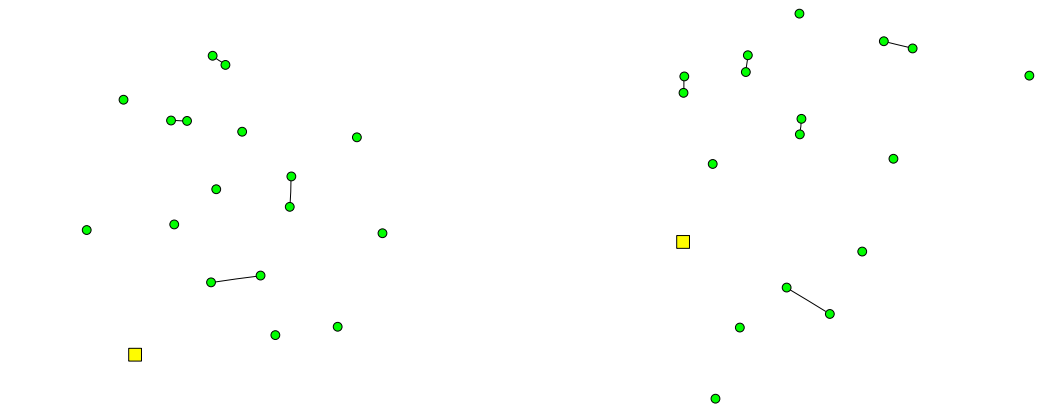
Figure B.12. Topic 350: Health and Computer Terminals



Regular
nMRD = 0.27, P5 = 0.59

Query-Biased
nMRD = 0.31, P5 = 0.67

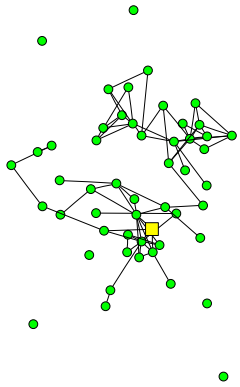
Figure B.13. Topic 370: food/drug laws



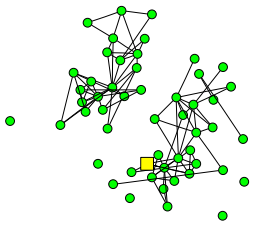
Regular
nMRD = 0.09, P5 = 0.08

Query-Biased
nMRD = 0.06, P5 = 0.08

Figure B.14. Topic 371: health insurance holistic

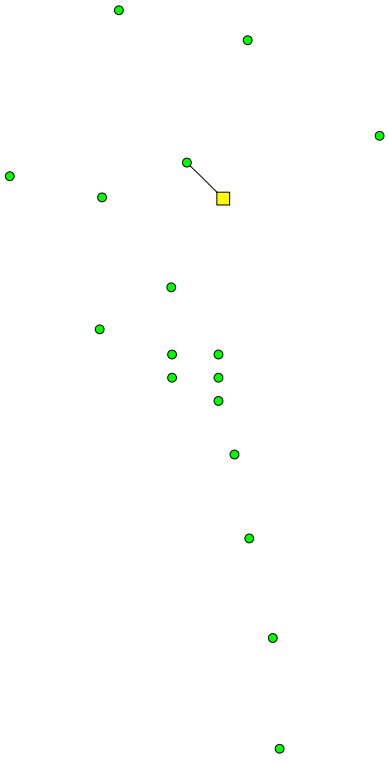


Regular
nMRD = 0.25, P5 = 0.33

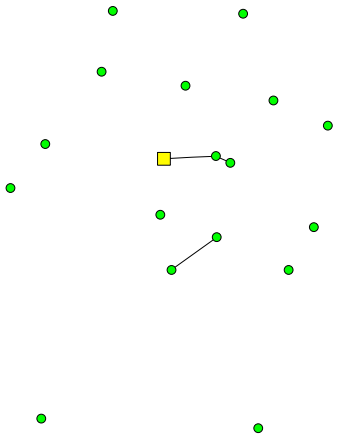


Query-Biased
nMRD = 0.29, P5 = 0.40

Figure B.15. Topic 384: space station moon



Regular
nMRD = 0.01, P5 = 0.00



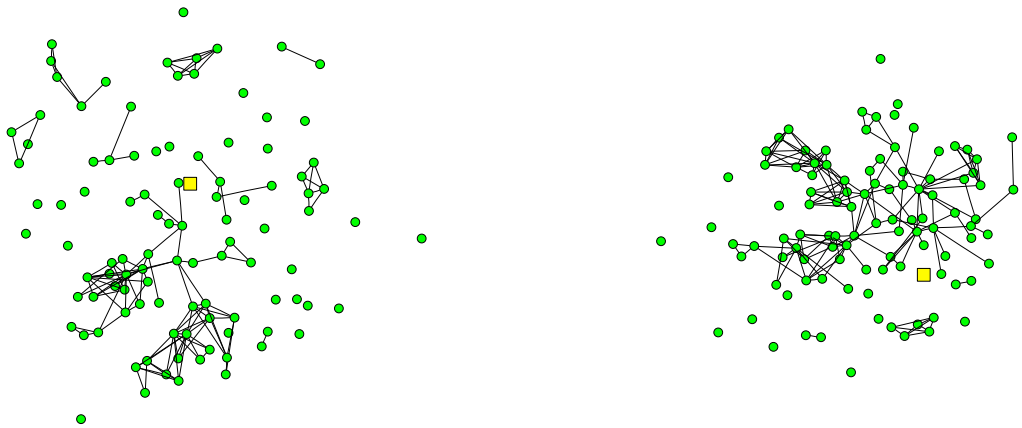
Query-Biased
nMRD = 0.03, P5 = 0.02

Figure B.16. Topic 394: home schooling

Regular
nMRD = 0.17, P5 = 0.21

Query-Biased
nMRD = 0.19, P5 = 0.25

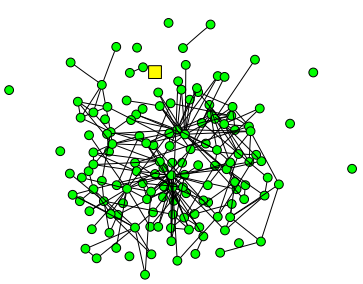
Figure B.17. Topic 397: automobile recalls



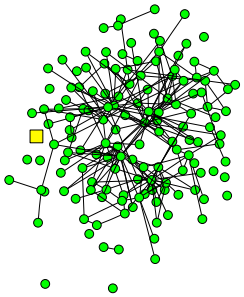
Regular
nMRD = 0.12, P5 = 0.34

Query-Biased
nMRD = 0.24, P5 = 0.44

Figure B.18. Topic 399: oceanographic vessels

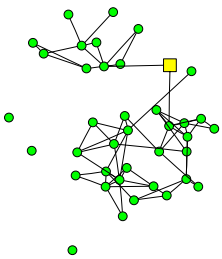


Regular
nMRD = 0.21, P5 = 0.39

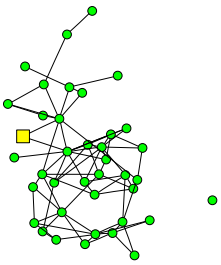


Query-Biased
nMRD = 0.17, P5 = 0.32

Figure B.19. Topic 404: Ireland, peace talks

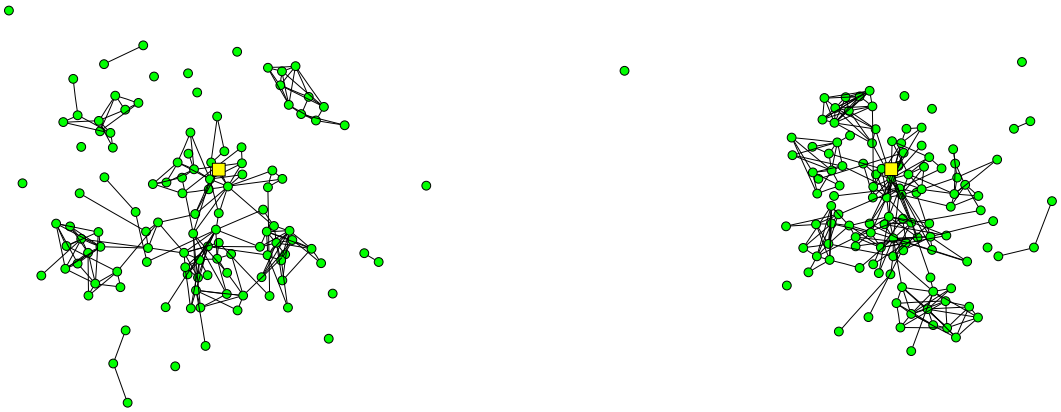


Regular
nMRD = 0.29, P5 = 0.42



Query-Biased
nMRD = 0.30, P5 = 0.45

Figure B.20. Topic 405: cosmic events



Regular
nMRD = 0.18, P5 = 0.45

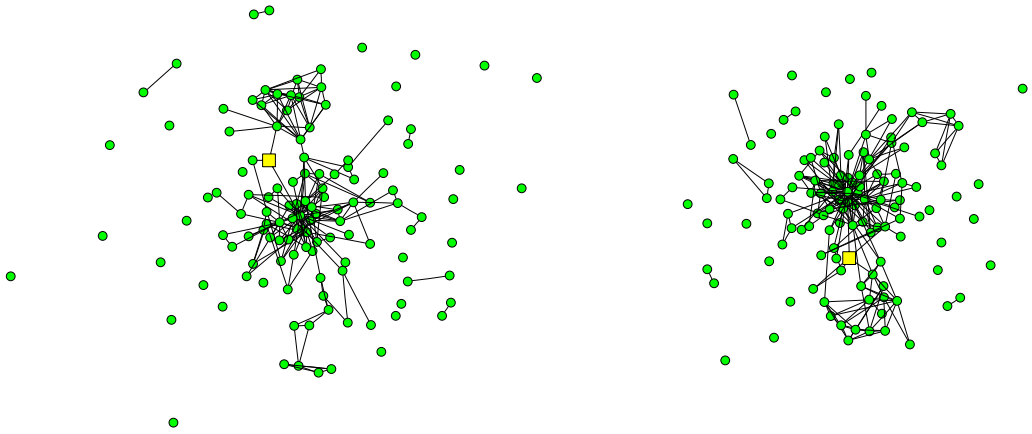
Query-Biased
nMRD = 0.27, P5 = 0.53

Figure B.21. Topic 408: tropical storms

Regular
nMRD = 0.28, P5 = 0.42

Query-Biased
nMRD = 0.52, P5 = 0.83

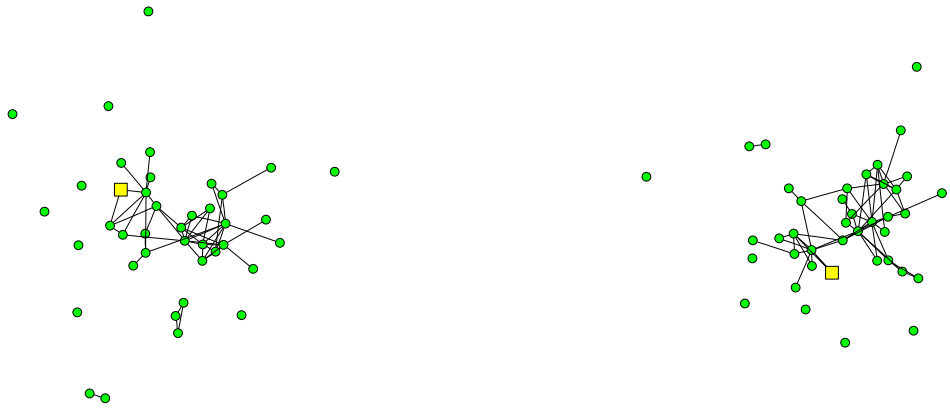
Figure B.22. Topic 410: Schengen agreement



Regular
nMRD = 0.20, P5 = 0.38

Query-Biased
nMRD = 0.27, P5 = 0.41

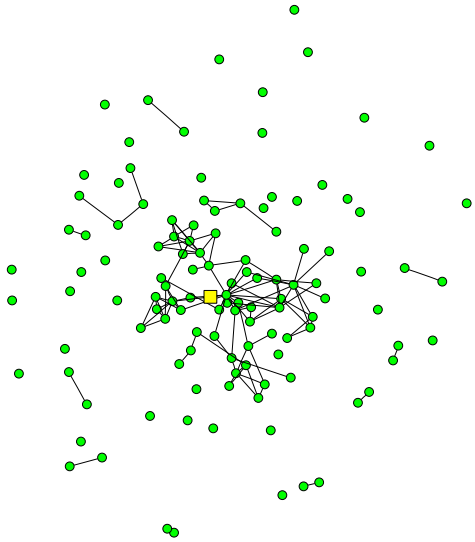
Figure B.23. Topic 412: airport security



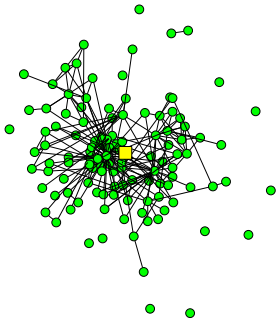
Regular
nMRD = 0.25, P5 = 0.33

Query-Biased
nMRD = 0.31, P5 = 0.37

Figure B.24. Topic 414: Cuba, sugar, exports

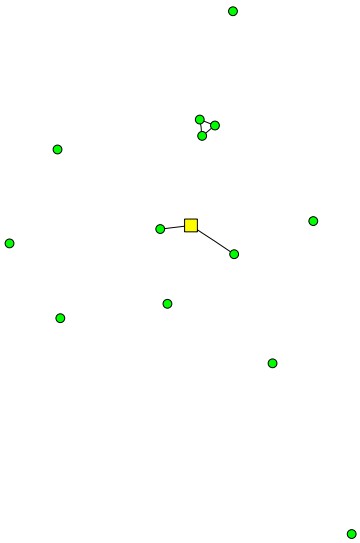


Regular
nMRD = 0.12, P5 = 0.23

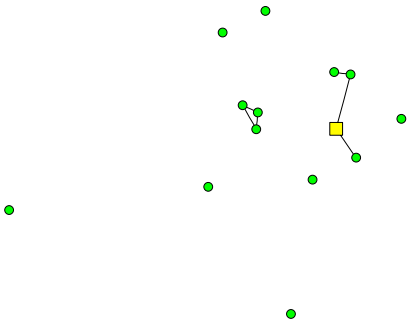


Query-Biased
nMRD = 0.23, P5 = 0.39

Figure B.25. Topic 428: declining birth rates

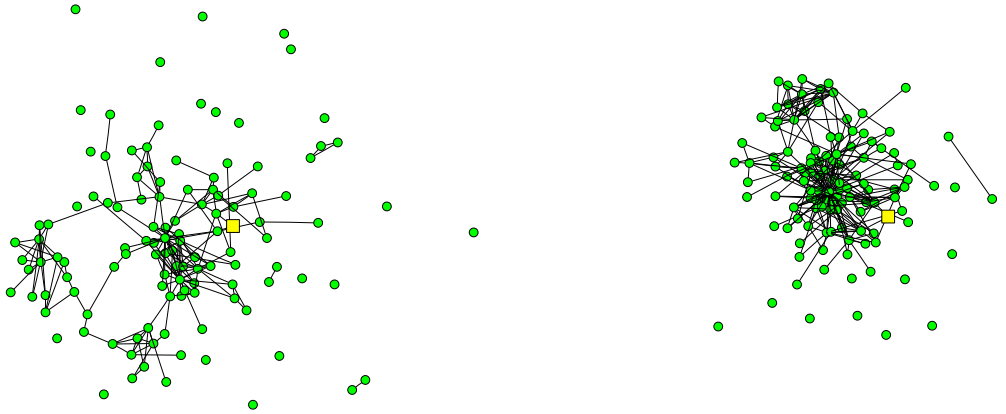


Regular
nMRD = 0.08, P5 = 0.09



Query-Biased
nMRD = 0.12, P5 = 0.11

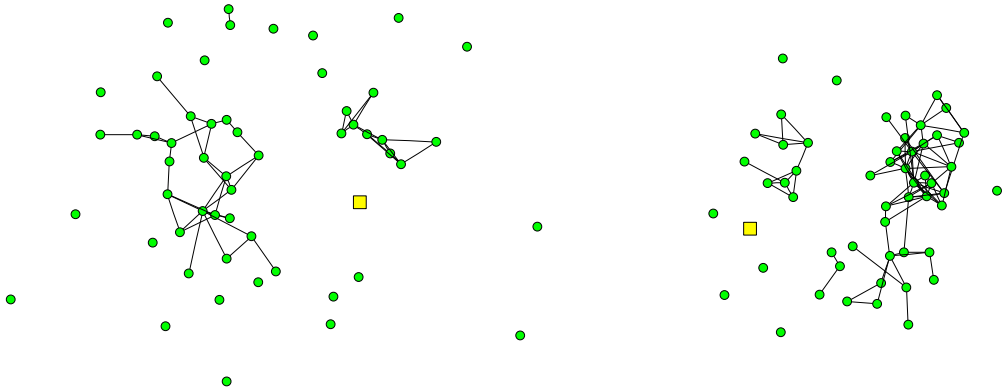
Figure B.26. Topic 433: Greek, philosophy, stoicism



Regular
nMRD = 0.19, P5 = 0.35

Query-Biased
nMRD = 0.26, P5 = 0.48

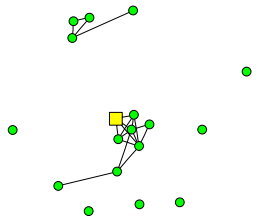
Figure B.27. Topic 435: curbing population growth



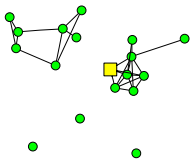
Regular
nMRD = 0.14, P5 = 0.23

Query-Biased
nMRD = 0.26, P5 = 0.39

Figure B.28. Topic 440: child labor

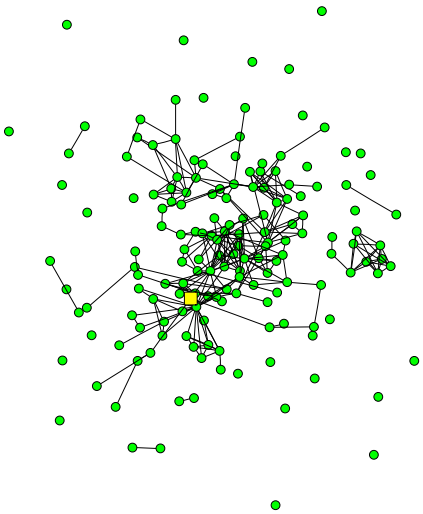


Regular
nMRD = 0.23, P5 = 0.28

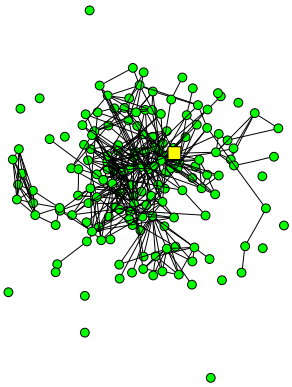


Query-Biased
nMRD = 0.28, P5 = 0.36

Figure B.29. Topic 441: Lyme disease



Regular
nMRD = 0.20, P5 = 0.42



Query-Biased
nMRD = 0.32, P5 = 0.57

Figure B.30. Topic 446: tourists, violence

BIBLIOGRAPHY

Aalbersberg, I. J. (1992). Incremental relevance feedback. In *SIGIR '92: Proceedings of the 15th annual international ACM SIGIR conference on research and development in information retrieval*, pp. 11–22. ACM Press.

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

Allan, J. (1997). Building hypertext using information retrieval. *Information Processing and Management 33*(2), 145–159.

Aula, A., P. Majaranta, and K.-J. Räihä (2005). Eye-tracking reveals the personal styles for search result evaluation. In *Human-Computer Interaction – INTERACT 2005*, Volume 3585 of *Lecture Notes in Computer Science (LNCS)*, pp. 1058–1061. IFIP International Federation for Information Processing: Springer.

Balasubramanian, N., J. Allan, and W. B. Croft (2007). A comparison of sentence retrieval techniques. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, pp. 813–814. ACM.

Barthélemya, M., A. Barratb, R. Pastor-Satorrasc, and A. Vespignanib (2005, February). Characterization and modeling of weighted networks. *Physica A: Statistical Mechanisc and its Applications 346*(1-2), 34–43.

Bates, M. J. (1979). Information search tactics. *Journal of the American Society for Information Science 30*, 205–214.

Bates, M. J. (1989). The design of browsing and berrypicking techniques for the online search interface. *Online Review 13*, 407–431. `http://www.gseis.ucla.edu/faculty/bates/berrypicking.html`.

Bates, M. J. (1990). Where should the person stop and the information search interface start? *Information Processing and Management 26*(5), 575–591.

Beaulieu, M. (1997, January). Experiments on interfaces to support query expansion. *Journal of Documentation 53*(1), 8–19.

Belkin, N. J., C. Cool, D. Kelly, S.-J. Lin, S. Y. Park, J. Perez-Carballo, and C. Sikora (2001). Iterative exploration, design and evaluation of support for query reformulation in interactive information retrieval. *Information Processing and Management 37*, 403–434.

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

Bhavnani, S. K., R. T. Jacob, J. Nardine, and F. A. Peck (2003). Exploring the distribution of online healthcare information. In *CHI '03: CHI '03 extended abstracts on Human factors in computing systems*, New York, NY, USA, pp. 816–817. ACM.

Birbeck, R. D., H. Joho, and J. M. Jose (2006). A sentence-based ostensive browsing and searching on the web. In *First International Workshop on Adaptive Information Retrieval (AIR)*, pp. 34–35.

Blustein, J., R. E. Webber, and J. Tague-Sutcliffe (1997). Methods for evaluating the quality of hypertext links. *Information Processing and Management 33*(2), 255–271.

Bodner, R. and M. Chignell (1999). Dynamic hypertext: querying and linking. *ACM Computing Surveys*, 15.

Bodner, R. C. and M. H. Chignell (1998). ClickIR: Text Retrieval using a Dynamic Hypertext Interface. In *Proceedings of the Seventh Text REtrieval Conference*.

Bodner, R. C., M. H. Chignell, N. Charoenkitkarn, G. Golovchinsky, and R. W. Kopak (2001). The impact of text browsing on text retrieval performance. *Information Processing and Management 37*(3), 507–520.

Bollacker, K. D., S. Lawrence, and C. L. Giles (1998). Citeseer: an autonous web agent for automatic retrieval and identification of interesting publications. In *AGENTS '98: Proceedings of the second international conference on Autonomous agents*, New York, NY, USA, pp. 116–123. ACM.

Botafogo, R. A., E. Rivlin, and B. Shneiderman (1992). Structural analysis of hypertexts: identifying hierarchies and useful metrics. *ACM Transactions on Information Systems 10*(2), 142–180.

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

Buckley, C. (2006). trec_eval. `http://trec.nist.gov/trec_eval/trec_eval.8.0.tar.gz`.

Campbell, I. (2000). *The ostensive model of developing information needs*. Ph. D. thesis, University of Glasgow.

Chen, H. and S. Dumais (2000). Bringing order to the web: automatically categorizing search results. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*, New York, NY, USA, pp. 145–152. ACM.

Cleverdon, C. (1967). The Cranfield tests on index language devices. In *Aslib Proceedings*, Volume 19, pp. 172–192. Reprinted in (Cleverdon, 1997).

Cleverdon, C. (1997). The Cranfield tests on index language devices. In *Readings in Information Retrieval*, pp. 47–59. Morgan Kaufman.

Cormen, T. H., C. Stein, C. Stein, R. L. Rivest, and C. E. Leiserson (2001). *Introduction to Algorithms*. McGraw-Hill Higher Education.

Costa, L., F. A. Rodrigues, G. Travieso, and P. R. V. Boas (2007, January). Characterization of complex networks: A survey of measurements. *Advances in Phyiscs 56*(1), 167–242.

Croft, W. B. (1995, Nov.). What do people want from information retrieval? *D-Lib Magazine*.

Croft, W. B., T. J. Lucia, J. Cringean, and P. Willett (1989). Retrieving documents by plausible inference: an experimental study. *Information Processing and Management 25*(6), 599–614.

Croft, W. B. and R. H. Thompson (1987). I3R: A new approach to the design of document retrieval systems. *Journal of the American Society for Information Science 38*(6), 389–404.

Cutrell, E., D. Robbins, S. Dumais, and R. Sarin (2006). Fast, flexible filtering with phlat. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, New York, NY, USA, pp. 261–270. ACM Press.

Dean, J. and M. R. Henzinger (1999a). Finding related pages in the world wide web. In *WWW '99: Proceedings of the eighth international conference on World Wide Web*, New York, NY, USA, pp. 1467–1479. Elsevier North-Holland, Inc.

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

Diaz, F. (2008, February). *Autocorrelation and Regularization of Query-Based Information Retrieval Scores*. Ph. D. thesis, University of Massachusetts Amherst.

Doyle, L. B. (1962). Indexing and abstracting by association. part 1. Technical Report SP-718/001/00, System Development Corporation, Santa Monica, CA. Reprinted in (Doyle, 1997).

Doyle, L. B. (1997). Indexing and abstracting by association. part 1. In *Readings in Information Retrieval*, pp. 25–38. Morgan Kaufman.

Dumais, S., E. Cutrell, R. Sarin, and E. Horvitz (2004). Implicit queries (IQ) for contextualized search. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval*, pp. 594. ACM Press.

Dunlop, M. D. (1997). Time, relevance and interaction modelling for information retrieval. In *SIGIR '97: Proceedings of the 20th annual international ACM SIGIR conference on research and development in information retrieval*. ACM Press.

Eguchi, K. (1999). Adaptive cluster-based browsing using incrementally expanded queries and its effects (poster abstract). In *SIGIR '99: Proceedings of the 22th annual international ACM SIGIR conference on research and development in information retrieval*, pp. 265–266. ACM Press.

Ellis, D. (1989). A behavioral approach to information retrieval system design. *J. Doc. 45*(3), 171–212.

Furnas, G. W. (1997). Effective view navigation. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 367–374. ACM Press.

Golovchinsky, G. (1997). What the query told the link: the integration of hypertext and information retrieval. In *HYPERTEXT '97: Proceedings of the eighth ACM conference on Hypertext*, New York, NY, USA, pp. 67–74. ACM.

Golovchinsky, G. and M. Chignell (1993). Queries-r-links: graphical markup for test navigation. In *CHI '93: Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems*, New York, NY, USA, pp. 454–460. ACM.

Griffiths, A., H. C. Luckhurst, and P. Willett (1997). Using interdocument similarity information in document retrieval systems. pp. 365–373.

Hancock-Beaulieu, M., M. Fieldhouse, and T. Do (1995). An evaluation of interactive query expansion in an online library catalogue with a graphical user interface. *Journal of Documentation 51*(3), 225–243.

Harper, D. J. and D. Kelly (2006). Contextual relevance feedback. In *IIiX: Proceedings of the 1st international conference on Information interaction in context*, New York, NY, USA, pp. 129–137. ACM Press.

Haveliwala, T. H., A. Gionis, D. Klein, and P. Indyk (2002). Evaluating strategies for similarity search on the web. In *WWW '02: Proceedings of the 11th international conference on World Wide Web*, New York, NY, USA, pp. 432–442. ACM Press.

Hawking, D. and N. Craswell (2005). The very large collection and web tracks. In E. M. Voorhees and D. K. Harman (Eds.), *TREC*, Chapter 9, pp. 199–231. MIT Press.

Hearst, M. A. and J. O. Pedersen (1996). Reexamining the cluster hypothesis: scatter/gather on retrieval results. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on research and development in information retrieval*, pp. 76–84. ACM Press.

Hersh, W., A. Cohen, J. Yang, R. T. Bhupatiraju, P. Roberts, and M. Hearst (2005). TREC 2005 genomics track overview. In *The Fourteenth Text REtrieval Conference (TREC 2005)*. Department of Commerce, National Institute of Standards and Technology.

Hersh, W. R., R. T. Bhupatiraju, L. Ross, P. Johnson, A. M. Cohen, and D. F. Kraemer (2004). TREC 2004 genomics track overview. In *The Thirteenth Text REtrieval Conference (TREC 2004)*. Department of Commerce, National Institute of Standards and Technology.

Herskovic, J. R., L. Y. Tanaka, W. Hersh, and E. V. Bernstam (2007). A Day in the Life of PubMed: Analysis of a Typical Day's Query Log. *Journal of the American Medical Informatics Association 14*(2), M2191.

Huang, X., M. Zhong, and L. Si (2005). York University at TREC 2005: Genomics track. In *The Fourteenth Text REtrieval Conference (TREC 2005)*. Department of Commerce, National Institute of Standards and Technology.

Huggett, M. and J. Lanir (2007). Static reformulation: a user study of static hypertext for query-based reformulation. In *JCDL '07: Proceedings of the 2007 conference on Digital libraries*, pp. 319–328. ACM Press.

Ivory, M. Y. and M. A. Hearst (2001). The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys 33*(4), 470–516.

Iwayama, M. (2000). Relevance feedback with a small number of relevance judgments: incremental relevance feedback vs. document clustering. In *SIGIR '00: Proceedings of the 23th annual international ACM SIGIR conference on research and development in information retrieval*, pp. 10–16. ACM Press.

Jardine, N. and C. J. van Rijsbergen (1971). The use of hierarchic clustering in information retrieval. *Information Storage and Retrieval 7*(5), 217–240.

Jeh, G. and J. Widom (2002). Simrank: a measure of structural-context similarity. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, USA, pp. 538–543. ACM.

Joho, H., R. D. Birbeck, and J. M. Jose (2007). An ostensive browsing and searching on the web. In *Proceedings of the 2nd International Workshop on Context-based Information Retrieval, CONTEXT'07, Copenhagen, Denmark: Roskilde University*, pp. 81–92.

Kalna, G. and D. J. Higham (2006, March). Clustering coefficients for weighted networks. In *Proceedings of AISB'06: Adaptation in Artificial and Biological Systems*. Also as University of Strathclyde Mathematics Research Report 03 (2006), March 2006.

Kelly, D. and J. Teevan (2003). Implicit feedback for inferring user preference: a bibliography. *SIGIR Forum 37*(2), 18–28.

Klöckner, K., N. Wirschum, and A. Jameson (2004). Depth- and breadth-first processing of search result lists. In *CHI '04: CHI '04 extended abstracts on Human factors in computing systems*, New York, NY, USA, pp. 1539–1539. ACM.

Koenemann, J. and N. J. Belkin (1996). A case for interaction: a study of interactive information retrieval behavior and effectiveness. In *CHI '96: Proceedings of the SIGCHI conference on Human factors in computing systems*, New York, NY, USA, pp. 205–212. ACM.

Krovetz, R. (1993). Viewing morphology as an inference process. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on research and development in information retrieval*, pp. 191–202. ACM Press.

Latora, V. and M. Marchiori (2001, Oct). Efficient behavior of small-world networks. *Physical Review Letters 87*(19), 198701.

Lavrenko, V. and W. B. Croft (2001). Relevance based language models. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval*, pp. 120–127. ACM Press.

Lawrie, D. J. (2003). *Language models for hierarchical summarization*. Ph. D. thesis.

Lemur (2003). Lemur Toolkit for Language Modeling and IR. `http://www.lemurproject.org/`.

Leuski, A. (2000). Relevance and reinforcement in interactive browsing. In *CIKM '00: Proceedings of the ninth international conference on Information and knowledge management*, New York, NY, USA, pp. 119–126. ACM.

Liben-Nowell, D. and J. Kleinberg (2003). The link prediction problem for social networks. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, New York, NY, USA, pp. 556–559. ACM.

Lieberman, H. (1995). Letizia: An agent that assists web browsing. In *IJCAI-95*, pp. 924–929.

Lin, J., M. DiCuccio, V. Grigoryan, and W. J. Wilbur (2007, July). Exploring the effectiveness of related article search in PubMed. Technical Report LAMP-TR-145/CS-TR-4877/UMIACS-TR-2007-36/HCIL-2007-10, College of Information Studies, University of Maryland, College Park.

Lin, J. and M. D. Smucker (2008). How do users find things with PubMed? Towards automatic utility evaluation with user simulations. In *SIGIR 2008, Singapore*. To appear.

Lin, J. and W. J. Wilbur (2007). PubMed related articles: A probabilistic topic-based model for content similarity. *BMC Bioinformatics 8*(423).

Lin, Z., M. R. Lyu, and I. King (2006). Pagesim: a novel link-based measure of web page aimilarity. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, New York, NY, USA, pp. 1019–1020. ACM.

Linden, G., B. Smith, and J. York (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing 7*(1), 76–80.

Mackinlay, J. D., R. Rao, and S. K. Card (1995). An organic user interface for searching citation links. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, New York, NY, USA, pp. 67–73. ACM Press/Addison-Wesley Publishing Co.

Meho, L. I. and H. R. Tibbo (2003). Modeling the information-seeking behavior of social scientists: Ellis's study revisited. *Journal of the American Society of Information Science and Technology 54*(6), 570–587.

Melucci, M. (1999). An evaluation of automatically constructed hypertexts for information retrieval. *Information Retrieval 1*(1-2), 91–114.

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

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

Metzler, D., Y. Bernstein, W. B. Croft, A. Moffat, and J. Zobel (2005). Similarity measures for tracking information flow. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, New York, NY, USA, pp. 517–524. ACM.

Metzler, D. and W. B. Croft (2004). Combining the language model and inference network approaches to retrieval. *Information Processing and Management 40*(5), 735–750.

Metzler, D. and W. B. Croft (2005). A markov random field model for term dependencies. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval*, pp. 472–479. ACM Press.

Metzler, D., F. Diaz, T. Strohman, and W. B. Croft (2005). UMass robust 2005 notebook: Using mixtures of relevance models for query expansion. In *TREC 2005 Notebook*.

Murdock, V. (2006). *Aspects of Sentence Retrieval*. Ph. D. thesis, University of Massachusetts Amherst.

Murdock, V. and W. B. Croft (2005). A translation model for sentence retrieval. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Morristown, NJ, USA, pp. 684–691. Association for Computational Linguistics.

Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Review 45*, 167–256.

Oddy, R. N. (1977, March). Information retrieval through man-machine dialogue. *Journal of Documentation 33*(1), 1–14.

Olston, C. and E. H. Chi (2003). Scenttrails: Integrating browsing and searching on the web. *ACM Transactions on Computer-Human Interaction 10*(3), 177–197.

Pirolli, P. (1997). Computational models of information scent-following in a very large browsable text collection. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, New York, NY, USA, pp. 3–10. ACM Press.

Pirolli, P. (2007). *Information Foraging Theory*. Oxford University Press.

Pirolli, P. and S. Card (1999). Information foraging. *Psychological Review 106*(4), 643–675.

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

Ponte, J. M. and W. B. Croft (1998). A language modeling approach to information retrieval. In *SIGIR '98: Proceedings of the 21th annual international ACM SIGIR conference on research and development in information retrieval*, pp. 275–281. ACM Press.

Rhodes, B. and T. Starner (1996, April). The remembrance agent: A continuously running automated information retrieval system. In *The Proceedings of The First International Conference on The Practical Application of Intelligent Agents and Multi Agent Technology (PAAM '96), London, UK*, pp. 487–495.

Rocchio, J. J. (1971). Relevance feedback in information retrieval. In G. Salton (Ed.), *The SMART Retrieval System*, pp. 313–323. Prentice Hall.

Rose, D. E. and D. Levinson (2004). Understanding user goals in web search. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, New York, NY, USA, pp. 13–19. ACM Press.

Ruthven, I. and M. Lalmas (2003). A survey on the use of relevance feedback for information access systems. *The Knowledge Engineering Review 18*(2), 99–145.

Sanderson, M. (1998). Accurate user directed summarization from existing tools. In *CIKM '98: Proceedings of the seventh international conference on Information and knowledge management*, New York, NY, USA, pp. 45–51. ACM.

Siek, J. G., L.-Q. Lee, and A. Lumsdaine (2001). *The Boost Graph Library*. Addison Wesley.

Şimşek, Özgür. and D. Jensen (2005). Decentralized search in networks using homophily and degree disparity. In *Proceedings of the Nineteenth Internation Joint Conference on Artificial Intelligence (IJCAI-05), Edinburgh, Scotland*, pp. 304–310.

Smucker, M. D. and J. Allan (2006). Find-similar: Similarity browsing as a search tool. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval*, pp. 461–468. ACM Press.

Smucker, M. D. and J. Allan (2007a). Measuring the navigability of document networks. In *SIGIR '07 Web Information-Seeking and Interaction Workshop*.

Smucker, M. D. and J. Allan (2007b). Using similarity links as shortcuts to relevant web pxages. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, pp. 863–864. ACM.

Smucker, M. D., J. Allan, and B. Carterette (2007). A comparison of statistical significance tests for information retrieval evaluation. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, New York, NY, USA, pp. 623–632. ACM.

Smucker, M. D., J. Allan, and B. Dachev (2008). Human question answering performance using an interactive information retrieval system. Center for Intelligent Information Retrieval Technical Report IR-655, University of Massachusetts.

Soboroff, I. (2002). Do TREC web collections look like the web? *SIGIR Forum 36*(2), 23–31.

Spink, A., B. J. Jansen, and H. C. Ozmultu (2000). Use of query reformulation and relevance feedback by excite users. *Internet Research: Electronic Networking Applications and Policy 10*(4), 317–328.

Spink, A., D. Wolfram, B. J. Jansen, and T. Saracevic (2001). Searching the web: The public and their queries. *Journal of the American Society of Information Science and Technology 52*(3), 226–234.

Strasberg, H. R., C. D. Manning, T. C. Rindfleisch, , and K. L. Melmon (2000). What's related? generalizing approaches to related articles in medicine. In *AMIA Symp.*, pp. 838–842.

Strohman, T., D. Metzler, H. Turtle, and W. B. Croft (2005). Indri: A language-model based search engine for complex queries (extended version). Technical Report IR-407, CIIR, CS Dept., U. of Mass. Amherst.

Takaki, T., A. Fujii, and T. Ishikawa (2004). Associative document retrieval by query subtopic analysis and its application to invalidity patent search. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, New York, NY, USA, pp. 399–405. ACM.

Teevan, J., C. Alvarado, M. S. Ackerman, and D. R. Karger (2004). The perfect search engine is not enough: a study of orienteering behavior in directed search. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, New York, NY, USA, pp. 415–422. ACM Press.

Terra, E. and R. Warren (2005). Poison pills: harmful relevant documents in feedback. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, New York, NY, USA, pp. 319–320. ACM.

Thelwall, M. and D. Wilkinson (2004). Finding similar academic web sites with links, bibliometric couplings and colinks. *Information Processing and Management 40*(3), 515–526.

Thompson, R. H. and W. B. Croft (1989). Support for browsing in an intelligent text retrieval system. *Int. J. Man-Machine Studies 30*, 639–668.

Tombros, A. (2002). *The effectiveness of query-based hierarchic clustering of documents for information retrieval*. Ph. D. thesis, University of Glasgow.

Tombros, A. and M. Sanderson (1998). Advantages of query biased summaries in information retrieval. In *SIGIR '98: Proceedings of the 21th annual international ACM SIGIR conference on research and development in information retrieval*, New York, NY, USA, pp. 2–10. ACM Press.

Tombros, A. and C. J. van Rijsbergen (2001). Query-sensitive similarity measures for the calculation of interdocument relationships. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, New York, NY, USA, pp. 17–24. ACM.

Travers, J. and S. Milgram (1969). An experimental study of the small world problem. *Sociometry 32*(4), 425–443.

Turtle, H. (1994). Natural language vs. boolean query evaluation: a comparison of retrieval performance. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, pp. 212–220. Springer-Verlag New York, Inc.

Turtle, H. and W. B. Croft (1991). Evaluation of an inference network-based retrieval model. *ACM Transactions on Information Systems 9*(3), 187–222.

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

Vassilvitskii, S. and E. Brill (2006). Using web-graph distance for relevance feedback in web search. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval*, pp. 147–153. ACM Press.

Voorhees, E. M. (1985). The cluster hypothesis revisited. In *SIGIR '85: Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 188–196. ACM Press.

Voorhees, E. M. (2005). The TREC robust retrieval track. *SIGIR Forum 39*(1), 11–20.

Voorhees, E. M. and H. T. Dang (2005). Draft: Overview of the TREC 2005 robust retrieval track. In *TREC 2005 Notebook*, pp. 105–112.

Voorhees, E. M. and D. K. Harman (Eds.) (2005). *TREC*. MIT Press.

Warren, R. H. and T. Liu (2004). A review of relevance feedback experiments at the 2003 reliable information access (ria) workshop. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval*, pp. 570–571. ACM Press.

White, R. W., J. M. Jose, C. J. van Rijsbergen, and I. Ruthven (2004). A simulated study of implicit feedback models. In *Proceedings of the 2004 European Conference on Information Retrieval*.

Wilbur, W. J. and L. Coffee (1994). The effectiveness of document neighboring in search enhancement. *Information Processing and Management 30*(2), 253–266.

Wilkinson, R. and A. F. Smeaton (1999). Automatic link generation. *ACM Computing Surveys*, 27.

Zellweger, P. T., B.-W. Chang, and J. D. Mackinlay (1998). Fluid links for informed and incremental link transitions. In *HYPERTEXT '98: Proceedings of the ninth ACM conference on Hypertext and hypermedia : links, objects, time and space—structure in hypermedia systems*, New York, NY, USA, pp. 50–57. ACM Press.

Zhai, C. and J. Lafferty (2001). A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval*, pp. 334–342. ACM Press.