A Framework for Automated Evaluation of Hypertext Search Interfaces

Rick Bodner and Mark Chignell
Interactive Media Lab
Dept. of Mechanical and Industrial Engineering
University of Toronto
Toronto, Ontario, Canada, M5S 1A4
Email: rbodner@acm.org, chignell@mie.utoronto.ca

Abstract

An evaluation framework and simulator of an interactive information retrieval system is introduced. The simulation environment implements a model of an interactive information retrieval system that uses an adaptive hypertext interface to present documents to the user. Links within such an interface are generated at runtime based on previous link selections. In addition to the model of the hypertext interface, several agents have been implemented which embody prototypical information exploration styles for such an interactive information retrieval interface. The simulation environment is designed to allow researchers to conduct exploratory investigations that can help to narrow the focus of future human subject studies by showing which differences in information exploration style and functionality within the interface or underlying search algorithms that are likely to produce significant differences in future human subject studies.

An experiment was carried out to demonstrate how the simulation environment could be used to predict performance when using different search strategies in a dynamic hypertext environment. Analyzes were conducted on both agent performance (i.e., precision and recall) and behaviour (i.e., link selections and documents visited). The analyses of both the performance and behavioural measures showed significant differences for a number of experiment parameters, such as query difficulty (a post-hoc measure), newness (controls the ability to revisit a document), and query tail size (an indicator of how easy it is to modify the topic during the search). Overall, the agents differed in terms of their behaviour compared to one another and in terms of their interaction with the simulator parameter of newness and the dynamic hypertext control parameter of query tail size. The analysis of the performance measures showed the same pattern as found in the behaviour measures, with query tail size having a strong influence on performance. The results of this study are discussed in terms of their implications for future automated evaluation of adaptive hypertext search interfaces.

Contents

1 Introduction

In Smeaton *et al.* (2002), the authors reviewed 25 years of papers from past SIGIR conferences. In the paper, the authors found that the number of papers focused on evaluation has been growing since 1993 (which is correlated with the start of the first TREC conference). As more research in the field of information retrieval is focused on evaluation, there is a need to consider the role of the user in the search process. Previous evaluation methodologies, either automatic or manual, do not focus on the individual searcher, but primarily on the system being investigated. Such system-centric models of evaluations are in contrast with most models of information exploration, which are user-centric. Overall information retrieval evaluation models have been slow to incorporate the user and in particular individual differences between users.

Many interactive information retrieval (IIR) systems have been developed and there is a need to evaluate the systems using standard text collections and methodologies. IIR systems place the user in active role of identifying relevant documents, via the interface, in the search process. Thus, user-centric IIR systems are difficult to evaluate a systematic manner due the limitations of user population samples available (typically small and homogeneous samples). While there has been considerable progress made in developing system-centric information retrieval test collections, methodologies, and measurement methods (e.g., precision and recall), the variability of human subjects remains a problem in assessing interactive information retrieval systems.

Adaptive hypertext interfaces for searching are a class of interactive information retrieval system. The non-linear nature of hypertext makes it even more of a challenge to use system-centric methodologies for evaluation. In the past we have developed and test systems based on our dynamic hypertext model in which links are generated at runtime based on the content of previous link selections. We have found that the benefits of such a hypertext search interface varies depending on the user's information exploration style (search strategy). Most notably, search novices benefit from the recognition-based querying provided by dynamic hypertext interfaces. Although past results have shown some benefits to certain users groups, we have not been able to extensively measure various versions of the model due to limitations in obtaining large, representative samples of users across a wide range of experimental conditions. Our participation in the Interactive Track at TREC-7 (see Bodner and Chignell 1999 for more details) further illustrated the difficulties of evaluating interactive information retrieval systems in a systems-centric evaluation framework. Improvements were made to the TREC evaluation framework for the Interactive Track in terms of identifying relevant "chunks" of information at the instance level as opposed to the document level. Unfortunately due to the overall constraints of the evaluation framework employed only limited differences between a control and an experimental system, and any resulting change in impact on users, could be investigated.

Although some work is beginning to be carried out on how individual differences can affect system performance (see Over, 1999), little work has been done on how individuals or groups differ in their search patterns or usage of hypertext interfaces. In addition, there is no standardized means to compare interactive information retrieval systems with one another. For example, different systems might be evaluated using either first year university students or professional librarians, thus making comparisons difficult. This is a problem for many people who wish to deploy these tools. It would be valuable for them to know detailed facts such as: "System A, with interface B can improve novice retrieval performance over system A, with interface C".

The framework presented in this paper attempts to bridge the gap between the advantages (i.e., feature coverage, and repeatability) of system-centric evaluation methodologies and user-centric human subject experiment results by developing a framework for the automatic evaluation of interactive information retrieval systems, in particular hypertext search interfaces. The framework employs the use several prototypical search strategies for automatic evaluation of an interactive information retrieval system. The simulation environment (SIIRS) based on the framework allows a research to evaluate an IIR system using five different search strategies. Both performance and behavioural measures can be observed and analyzed using SIIRS. This automated interaction with an IIR system provides a researcher with the advantages of the system-centric evaluation methodologies, in all of the features of a system can be explored in combination with different search strategies in a repeatable manner. This provides the researcher with a powerful tool during system development where multiple human subject experiments are not cost effective. In addition, the results of the analyses can assist a researcher to focus a human subject experiment on users who employ search strategies that were found to show significant differences in the simulation. This focusing helps to eliminate some of the noise in the human subject experiment and focuses such an experiment on the search strategies and system features of interest.

The remainder of this paper describes previous work on information retrieval evaluation and automatic evaluation of user interfaces. Following that related research section, the framework for automatic evaluation of hypertext search interfaces is presented. A simulation environment, SIIIRS, which implements the framework using search agents, is also described. Finally, the use of SIIIRS is illustrated in a simulation experiment that compares the retrieval performance of different search agents implemented within SIIIRS.

2 Related Research

Information exploration in a hyperlinked document space such as the World Wide Web is cognitively demanding. Cognitive overload can occur when users are forced to memorize the structure of a large and complex collection and become disoriented - the "lost in hyperspace" dilemma (cf, Conklin 1987; Nielsen 1990; Parunak 1989). As a result, information exploration is often a mix of search and browsing. Intermixed searching and browsing tends to lead to a spiky navigation pattern (Campagnini and Ehrlich 1989; Parunak 1989) where users navigate in two modes: searching (e.g., querying the system) and browsing (e.g., reviewing the documents retrieved based on the previous query). The resulting mode switching can be problematic for users. The mode switching problem is still an issue for the World Wide Web. Search engines are used in information exploration tasks (e.g., find citations for a given paper) to locate relevant documents. A user switches from searching mode to browsing mode to review a document. The contents of a document may lead to revised or new queries (user switches back to searcher mode). Thus many of the problems identified for hypertext are relevant to the World Wide Web.

Waterworth and Chignell (1991) suggested that information exploration can be performed more effectively if browsing and searching are seamlessly blended together (i.e., the user does not need to switch modes). The idea of dynamic hypertext is based on a model of information exploration that blends these tasks into a single interface. One example of a dynamic hypertext system is ClickIR (Bodner and Chignell 1999).

Models of dynamic hypertext generally employ either link filtering or link enhancing methods. In the filtering model, all links are created a priori and some are then filtered out at runtime based on any number of parameters (e.g., prior lessons completed by a student). Link enhancing methods (the focus of this paper) add links that did not previously exist, to a document when a reader requests it. The potential advantage of link enhancement based is that links can be defined at run-time, based on the context of the collection and the interests of the user. In the ClickIR approach, when a user selects a dynamic link, he/she is really sending a query to search the collection (for a more detailed description of how this works see Bodner *et al.* 1997 and Tam *et al.* 1997).

2.1 Individual Differences

The dynamic hypertext approach to information exploration is motivated in part by the presence of individual differences in search strategies that affect information seeking performance. Bates (1990) reported that an individual search strategy could change relatively quickly within a single search session. Hancock-Beaulieu (1990) tracked how users in a text retrieval task switched from one search strategy to another. The users were libraries who began with a search for a particular item only to discover that the desired item was not available in the library. They then changed their strategy from search to scan (or browse). Belkin *et al.* (1990) also observed user behaviour in libraries and similarly noted a number of transitions from one kind of search strategy to another.

In addition to dynamic switching between strategies, there are also individual differences in searching ability (e.g., Borgman 1989; Fenichel 1981; Saracevic 1991). Iivonen (1995) found that searchers working in different types of search environments (e.g., public vs. non-public organizations) had different types of work experience and that differences in searchers' experiences caused them to prefer different search methods, different terminological styles and different search strategies. Borgman (1989) observed that people brought different skills and talents to the task of information retrieval. She found high variability in searching behaviour, even when the same system and the same database were used. Fenichel (1981) found that even in a group of users with the same experience levels, the difference between the search process and outcome measures for the same search topics sometimes varied by a factor of ten or more.

The presence of strong individual differences in information exploration behaviour raises challenges for the evaluation of interactive information retrieval systems. Consequently, the topic of how to evaluate information exploration systems will now be considered.

2.2 Evaluation Methods

Evaluation has been an integral part of information retrieval research from its early days with the Cranfield experiments (Cleverdon *et al.* 1966) that used pre-defined queries that were run against a test collection in batch mode. In those early studies, the focus of evaluation was to test and compare components of information retrieval systems, such as an indexing mechanism. These system-centric evaluations were conducted in batch since test collections were developed which did not require human interaction.

As IR systems came into wider general use, there was also a focus on user-centric evaluations. Not only were the system-centric questions of reliability, computational effectiveness and efficiency being addressed, but also the questions regarding displaying search results, feedback, etc. The need arose for new evaluation methods that dealt with interactive, rather than batch mode information retrieval. Saracevic (1995) described six levels of evaluation in information retrieval research:

- Engineering: addresses questions about hardware and software performance,
- Input: addresses questions about the inputs and contents of the system being investigated,
- Processing: addresses questions regarding how the inputs are processed (e.g., performance of indexing algorithms, etc.),
- Output: addresses questions about interaction with a system's outputs (e.g., presenting and selecting retrieval results),
- Use and User: addresses questions of related to the application to given problems and tasks (e.g., evaluation of task difference between question-answering and shopping, or differences in search strategies between domain experts and novices), and
- Social: addresses questions about the effects that IR systems might have on social environments (e.g., effects on productivity or decision-making).

The first three levels can be roughly categorized as system-centric evaluation questions and the last three levels can be categorized as user-centric evaluation. Typically, laboratory tools, such as the batch mode evaluations used in the National Institute of Standards and Technology's (NIST) Text Retrieval Conference (TREC), addressed the system-centric questions. A number of questions arose regarding the applicability of such tools for real-world problems. Voorhees (1998, p. 315) suggested that the relative effectiveness of two IR systems should be "insensitive to modest changes in the relevant document set since individual relevance assessments are known to vary widely". She analyzed past TREC results and found that comparative evaluations between systems were stable despite variations in relevance judgments. She concluded that batch mode evaluations, such as those used in TREC, could be used as a laboratory tool for evaluation.

Due to the inherent benefits of using a controlled laboratory tool for batch mode evaluation (e.g., cost, repeatability, feature coverage, and experimental control), others have tried to use such tools to evaluate interactive information retrieval systems. Beaulieu *et al*. (1996) described the difficulty the Okapi system, which is an experimental interactive information retrieval system, experienced while participating in the first couple of TREC conferences. The authors described how they had to devise a method that used the interactive search formulation mechanism of the Okapi system to create a static query that could be used in the ad hoc search task at TREC. The authors had concerns regarding differences in domain knowledge between the subjects they recruited for their studies and the TREC analysts. They suggested that the "topics and data have specific characteristics and may reflect certain type of information seeker, whose motivation is not necessarily easy to replicate" (Beaulieu *et al.* 1996, p.88).

In summary, system-centric batch mode evaluations have the benefits of controlled, repeatable, experimental environments, which allow researchers to evaluate specific system features in detail. In contrast, user-centric evaluation techniques use human subjects and real-world tasks. Although it is difficult for user-centric research to focus on a specific system feature for evaluation, user studies have the advantage of being able to investigate how the user interacts with, and is affected by, the system. However, users typically have varying degrees of domain knowledge, cognitive abilities, search experience, etc. and it may be that interactive user studies are evaluating these properties of the user as much as they are evaluating properties of the information retrieval system.

2.3 Automation in Usability Studies

West and Emond (2002) identified a number of difficulties concerning usability testing with users. First, there is no way to control a subject's background in terms of how it might relate to the interface being tested. Nor is there any control over the subjects' motivation (subjects who are paid may not have the same motivation to find solutions to problems, as would a user who needs to use the system). Subject availability is also an issue for user studies. Due to the time and cost of using human subjects, often only a small number of subjects are used. In addition, due to the cost and the small number of subjects used, only a restricted set of functions or behaviours can be investigated in a single study.

Ivory and Hearst (2001) reviewed various methods for employing automation to evaluate user interfaces. The authors suggested that automation could assist with one of the main problems with usability studies, i.e., the lack of consistency between evaluators studying the same interface (cf. Jeffries *et al.* 1991; Kessner *et al.* 2001; Molich *et al.* 1998, 1999; Nielsen, 1993). In addition to issues related to consistency, the authors suggest several potential advantages of using automation in evaluation:

- Reducing cost of evaluation by introducing automation in the capture, analysis, or critique phases of evaluation,
- Increasing test coverage and consistency of errors discovered. Non-automated or automated random tests do not cover all possible combinations generated by expert and novice users. They noted that “usability evaluation typically only covers a subset of the possible actions users might take” (Ivory and Hearst 2001, p. 471) ,
- Prediction of time and error costs throughout the entire design and development processes.
- Enable more comparison between alternative designs. Due to the reduction in the time and cost to complete an automated evaluation, multiple design alternatives can be compared.

While simulation models have not been used much for modelling interactive information retrieval, they have been used quite extensively in modelling of human-computer interaction. The Goals, Operators, Methods, and Selection (GOMS) model (Card *et al.* 1983) and related methods can be used to analyze graphical user interfaces from a user perspective, and make predictions about quantities such as the speed of typical user performance. Peck and John (1992) described a Soar-based model, called Browser-Soar, which attempted to model how users interacted with (browsed) an online help system.

Although it seems unlikely that traditional user studies can ever be replaced by automated techniques, automated evaluation through simulation seems like a useful technique for finding additional insights about the properties of information exploration systems. As an example of this approach, we will now introduce an evaluation framework for simulating and assessing the effect of search strategies on interactive information retrieval performance when using dynamic hypertext.

3 Evaluation Framework

In this section, a framework for evaluation of interactive information retrieval systems (IIR) is presented. The framework proposes the use of idealized individual differences in search strategies as a means to analyze how an IIR may affect human users in terms of both performance and behaviour. We then report on the development of a simulator (based on this framework) that is used for automatic evaluation of an interactive information retrieval system using a dynamic hypertext interface.

This evaluation framework is designed to investigate how changes to IIR systems affect users. The framework was constructed to allow for automated evaluation of an IIR system (thereby reducing the need for costly user studies where the outcomes may be difficult to interpret due to the large amount of noise and variation in complex human behaviour such as information retrieval). Since the framework is intended to evaluate automatically interactive information retrieval systems, it was designed to model the interaction between a user and the IIR system. This approach does not attempt to model the higher-level cognitive processes (e.g., visual memory, working memory, attention span, fatigue, system model, etc.). A similar approach to that adopted in this dissertation was proposed by de Bruijn *et al.* (1999) in which the authors describe an interaction model that embodied only a single bottom-up model of a user's interaction behaviour. The cornerstone of the framework introduced here is the addition of individual differences, in terms of search strategies, to the interaction model.

The evaluation framework is presented in Figure 1. The figure shows a high-level view of the framework structure. The high-level components within the framework are the Search Agent and the Information IIR System boxes. These components comprise the automatic evaluation aspect of the framework. Since the framework is designed to evaluate IIR systems, it is assumed that there is an iterative search process between the interaction model (referred to as the *Search Agent* in the figure) and the IIR system. This iterative search process will continue until the search agent has satisfied its information need or cannot continue searching due to lack of relevant information. Based on the interaction between the Search Agent and the IIR System, an evaluation of the search performance (in terms of traditional information retrieval measures, such as precision and recall) can be conducted. In addition, link selection behaviours can be investigated. Typically, a search agent's link selections are compared against other agents' behaviour, or against link selections from the same agent but with different configurations for the IIR system. Using performance and behaviour measures, the evaluation framework can be used to analyze the effect that changes in an IIR system's configuration have on an agent's information retrieval behaviour.

The *Interactive Information Retrieval System* box represents an IIR system that employs a dynamic hypertext interface to an information retrieval system. The interaction between the search agent and the IIR system occurs when the user, in this case the search agent, selects a link from a hypertext document. The link is transformed into a query and sent to an information retrieval system. The IIR system returns a resulting hypertext document for the query.

The search agent is intended to mimic an interaction model for a specific search strategy. The interaction model is comprised of two components: the search strategy and the process of identifying potential solutions (relevant documents) to an information need. The search strategies are idealized (algorithmic) abstractions of strategies identified by studying individual differences in human searchers.

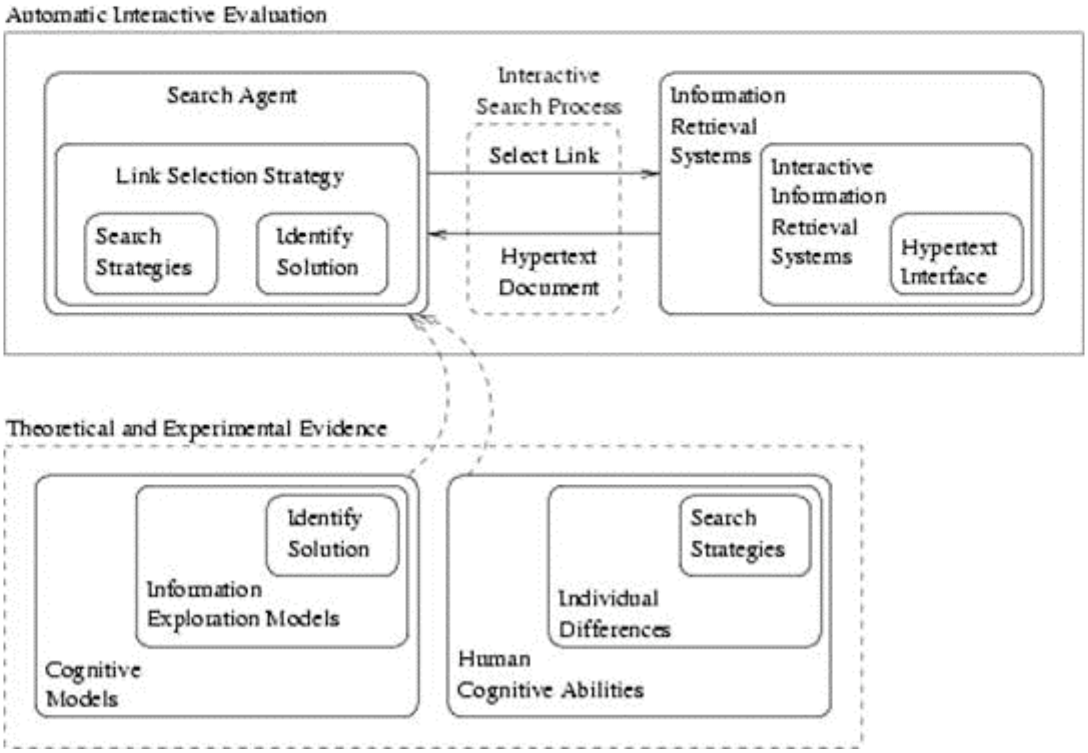
Automatic Interactive Evaluation
Search Agent
Link Selection Strategy
Search Strategies
Identify Solution
Interactive Search Process
Select Link
Hypertext Document
Information Retrieval Systems
Interactive Information Retrieval Systems
Hypertext Interface
Theoretical and Experimental Evidence
Identify Solution
Information Exploration Models
Cognitive Models
Search Strategies
Individual Differences
Human Cognitive Abilities

Figure 1. Evaluation Framework for IIIR Systems with Hypertext Interfaces.

Brandt and Uden (2003) describe six high-level tasks in performing a search: identify search topic, perform research, select search tool, create search query, execute search, and identify solution. It is the task of each search agent to identifying a solution to its information need. A mixture of declarative and procedural knowledge accomplishes this. Pirolli and Fu (2003) suggested that the act of search is the interaction between these two types of knowledge. In terms of searching in a hypertext environment, declarative knowledge represents hypertext links or the content of a document. Procedural knowledge represents how declarative knowledge is transformed into actions (i.e., link selection). Thus, the interaction model for the search agent only contains knowledge required to search out and follow relevant information to a specified query.

The agent is not intended to consider things that influence a user's information need, nor is it concerned with the prior processes of identifying a search, or creating a search query.

Although studies focusing on search strategies are available in the literature, many focus on high-level information seeking strategies rather than the hypertext search strategies (interaction models) that are the focus of this framework (c.f. Bates 1989; Belkin *et al.* 1990; Kuhlthau 1991; Oddy 1977; Xie 2000, 2002). Literature that does focus on web browsing typically propose general models of browsing, such as the Choo *et al.* (2000) empirical model of web use and the Huberman *et al.* (1998) general predictive model of browsing a given web site.

Information seeking strategies refer to how people look for information in both the physical (e.g., books in libraries) and computerized environments. Many of these papers also focus on strategy switching within a single search session. The search strategies discussed in the following section are stable interaction models that do not contain a switching component to their strategy. The decision to use a stable model, rather than a model that changes over time (e.g., a learning model), was due to the focus of the framework, which is on the evaluation of an interactive information retrieval system. The framework was developed for use in making comparisons between different search strategies interacting with information retrieval systems/algorithms/etc. If learning models were used, such comparisons would be more difficult. For example, it might be difficult to state that a system benefited from a novice search strategy if during a search session that novice agent started to exhibit behavioural characteristics of an expert searcher.

In addition, strategy switching is mediated by many high-level cognitive factors (e.g., user's motivation, type of search (informative or question-answering), fatigue, etc.) and external factors about the information retrieval system (i.e., type of documents contained in collection, search interface, etc.) which may interact with each other when a user decides to change strategies. Much research would need to be done in order to identify the factors and how they contribute to switching strategies. This was outside the scope of the research reported in this paper. By focusing on prototypical strategies, the framework can be used to evaluate distinct search strategies in a search session that employs multiple strategies.

Belkin *et al.* (1993) proposed four dimensions for classifying information seeking strategies. The dimensions are goal of interaction, method of interaction, model of retrieval, and type of resource interacted with. These dimensions interact with each other and can affect a user's information seeking strategy. Using the proposed agent simulation framework, the method of interaction, model of retrieval, and type of resource can be investigated. The goal of interaction is assumed to be to find as many relevant documents as possible based on the information need (predefined query) given to the search agent.

3.1 Search Strategies

This section presents five prototypical search strategies. The strategies were identified based on the research literature and on prior experimental results (Charoenkitkarn 1996 and Golovchinsky 1997a). The strategies are domain novice, domain expert, collection expert, skimmer, and reader. The domain novice and expert strategies will be discussed together since they are closely related, and are often discussed together in the literature. The skimmer and reader strategies will also be discussed together since they are related.

These five strategies are not intended to be an exhaustive list of search strategies used in an interactive information retrieval system using a hypertext interface. On the contrary, it is possible that many more prototypical strategies exist and that there exist many combinations of these strategies. Instead, these strategies were selected to test the validity of the evaluation framework.

3.1.1 Domain Novice and Expert

Much of the literature that describes user studies of information retrieval systems focuses on the differences between domain novices and experts (Bhavnani 2002; Goldstein-Hirsh 1995; Tam *et al.* 1997). Essentially, the main difference, in terms of the interaction model, between these types of users is in the level of knowledge about the problem domain. Toms *et al.* (2003) found that searchers tend to compare information currently available to them with their previous knowledge. Domain novices are assumed to have little or no knowledge of the problem domain. Due to lack of task domain knowledge, novices who do not use search intermediaries, such as librarians, "use simple and direct strategies" (Marchionini 1989, p. 55).

The novice agent's search strategy is to select a link, which is most similar to the query. This strategy is designed to reflect the actions of a user who is not an expert searcher and whose only piece of knowledge is the query itself.

In a hypertext-based information retrieval system, the domain expert agent ranks links based on their relevancy to its knowledge of the problem domain. Due to difficulty in modeling domain knowledge, a document from the test collection, deemed as authoritative with respect to the query, will be used as the agent's *domain knowledge*. This is intended to simulate a user who can differentiate the possible informational value between links in a document based on prior knowledge of the problem domain.

3.1.2 Collection Expert

The collection expert agent simulates a user with prior knowledge (i.e., the structure and the content) of the document collection itself (similar to a librarian). This agent's strategy is to find a document with a high similarity to the query and use that document as a jump point to investigate the rest of the collection (similar to a librarian suggesting a reference book or section in the library from which to start searching).

In a hypertext environment, this search strategy produces a spiky navigation pattern (Campagnini and Ehrlich 1989; Parunak 1989), where *jump* documents appear as the center of a star pattern. The points of the star represent paths of inquiry that a user followed before returning to the jump document. Kleinberg (1999) referred to such documents as hubs. In a study of 3190 searchers using the Alta Vista search engine, it was found that 14.83% of all searchers wanted a document containing a collection of links as opposed to a single relevant document (Broder 2002).

3.1.3 Skimmer and Reader

The strategies for the skimmer and reader search agents were derived from a mix of literature review and prior experimental findings obtained in the Interactive Media Laboratory at the University of Toronto. The experiments were conducted as part of the TREC-3 and TREC-4 conferences. The research followed a mixture of exploratory and confirmatory analyzes that is characterized and further discussed in the paper by Golovchinsky *et al.* (1997). The first set of experiments used the BrowsIR system (described in Charoenkitkarn 1996) which allowed users to mark-up queries by clicking and dragging between words in a document. In the experiments, subjects were classified as either search experts or novices depending on whether or not they had formal training in information retrieval (i.e., librarians vs. on-line searchers). The experiment showed that overall the mark-up interface improved precision and recall for only the search novices (but without hindering the experts). The subjects clustered into two main groups: inclusive and exclusive. The inclusive group consisted of subjects who selected and viewed more documents while the exclusive group selected and viewed fewer documents. This distinction between human searchers was a better predictor of performance than search expertise and therefore shows how differences in search strategy can be an important predictor of performance. The second set of experiments involved a system called VOIR (as described in Golovchinsky 1997a and Golovchinsky 1997b). It employed a dynamic hypertext interface for querying. Golovchinsky carried out cluster analysis to differentiate between different search strategies. Like the previous study, he found two main clusters. Readers were described as users who spent much of their time reading, and making only a few queries and relevance judgments. Skimmers consisted of users who issued a large number of queries and made many relevance judgments. Allowing for differences in the interfaces used across the two studies, Charoenkitkarn's inclusive searchers are roughly equivalent to skimmers. Similarly, the exclusives are roughly equivalent to Golovchinsky's readers. For a more detailed description of the experiments and the benefits of interactive interfaces to information retrieval, see Bodner *et al.* 2001. In the above experiments, experts tended to be more exclusive (precision-oriented readers), while novices were more inclusive, carrying out more queries and judging more documents. Based on these experiments a number of prototypical search strategies can be developed.

3.2 SIIRS

SIIRS stands for “Simulated Interaction of Interactive Information Retrieval Systems” and is described in greater detail in Bodner (2005). The SIIRS system is an implementation of the evaluation framework for hypertext search interfaces mentioned in the previous section. SIIRS provides an abstraction of the dynamic hypertext interface implemented in the ClickIR system (described in Bodner and Chignell 1999). The search agents interact with the abstraction interface and SIIRS interprets those interactions as queries that are presented to an information retrieval search system.

The SIIRS system consists of three main components: the simulator, search agent interface, IR system interface. The search agent interface provides a common set of functions that the search agents use to interact with abstraction of the hypertext interface, while the IR system interface provides a common set of functions for the simulator to communicate with various information retrieval systems or search engines.

The simulator component contains the abstraction of the dynamic hypertext search interface. The abstraction is continually updated based on requests from the search agent and responses from the IR system. The simulator has two main parameters that affect the behaviour of the dynamic hypertext interface: query tail size and newness. The first is query tail size, which defines the number of previous queries that are to be combined with the current query. Typically, small query tails (e.g., <2) create a situation were new query terms are introduced very quickly into the query tail and old terms are removed just as quickly. This creates a system that jumps quickly from topic to topic. The problem is that the system may misinterpret a link and switch to a new topic when the user does not want to switch, but instead wishes to expand his search. The reverse case is found when the query tail is set too high (e.g., >5). In this situation, old terms are not removed quickly enough and generally over-shadow the link selection process of dynamic hypertext. Thus, a delicate balance must be struck between "evolution" and "revolution" in link selection.

Newness dictates the how quickly the system will show a document that has already been viewed. A pilot study showed that a small newness value (e.g. < 4) led to situations where agents were restricted to a small set of documents during the course of the simulation. The agent would quickly be stuck in a core set of highly relevant documents and would traverse between documents in this set until the simulation finished. Although the documents in the core set tended to be relevant, a small newness value limited the search, in terms of the size of the set of documents covered during the search. Thus, the potential to retrieve other relevant documents was also limited, resulting in low performance scores. In addition, a small value limited the serendipity effect while searching using browsing techniques (i.e., hypertext). Toms (1998, 2000) has argued for the importance of serendipity in information retrieval. The pilot study also found that large newness values (e.g., > 15) increased the number of non-relevant documents viewed during the search session. Since newness determines when the agent can review a previously viewed document, a large value caused the agent to choose many documents that were not relevant, resulting in lower performance scores.

As previously stated, five search strategies for a dynamic hypertext were identified: domain novice, domain expert, collection expert, skimmer, and reader. Based on these search strategies, six search agents (five strategies plus a control agent referred to as "Random") were implemented. The phrase "search agent" refers to the software implementation that formalizes the link selection behaviour exhibited by different link search strategies. These search agents implement the tasks of selecting hypertext links (Search Strategy) that are likely to lead to relevant documents (Identify Solution). The information need part of an information exploration model will be provided by predefined queries which are given to the simulator at the start of the simulated search process.

The agents were implemented using a mixture of random link selection and link selection based on similarity measures. Figure 2 illustrates this agent hierarchy. The sixth agent was a control search agent (Random) implemented for the purpose of comparing performance and behaviour characteristics of the other search agents with a type of baseline behaviour (random selection of links).

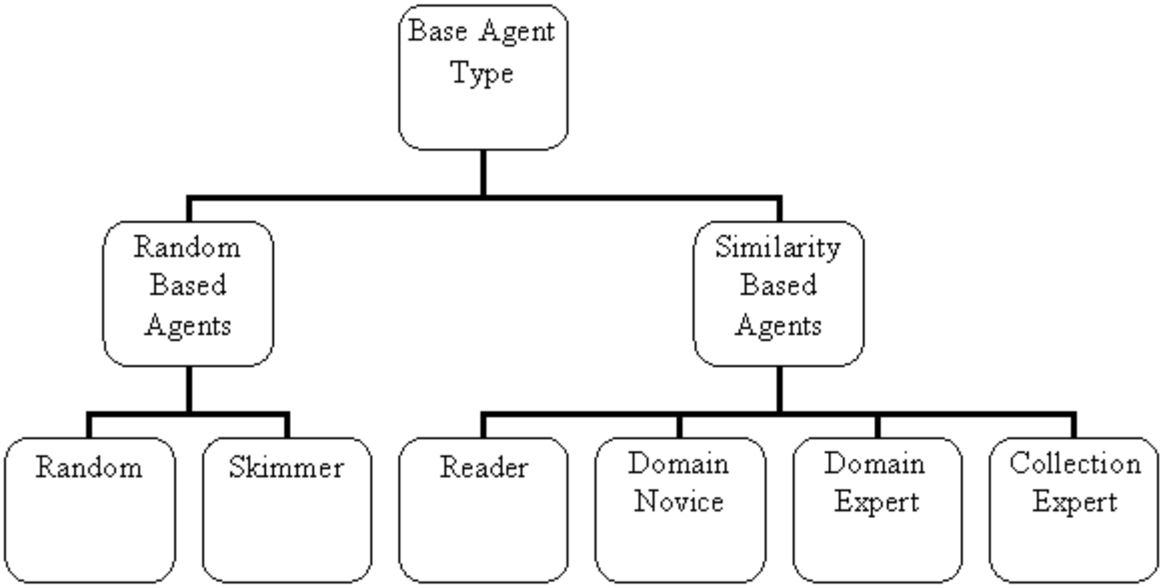

Figure 2. Search Agent Implementation Hierarchy.

4 Search Agent Experiment

As an initial test of the usefulness of SIIRS, an experiment was conducted to address two major research questions: “Is there a performance difference between the search agents?” and “Do the agent exhibit different search behaviours?”. In addition to these major research questions, two other research questions addressed by the experiment are “Does query difficulty affect the performance of the agent?” and “Are agents sensitive, in terms of performance and search behaviour, to simulator specific parameters, and if so, how?”. To answer these research questions, the six search agents were compared and contrasted based on the performance measures (e.g., precision and recall) and the measures that focused on search behaviour.

4.1 Experiment Measures

4.1.1 Performance Measurements

The standard equations for precision, average interpolated precision, recall, and Harmonic mean (F-Score) were used. Additional measure called *clicks to first relevant document* (CTFRD) was used to compare how quickly agents arrived at a relevant document.

4.1.2 Behaviour Measurements

Three behavioural measurements were used: redundancy, coverage, longest common click-path. Redundancy (see Equation 1) measures the amount of document revisits by calculating the ratio of duplicate documents over the total length of the agent's click-path. A click-path is the sequence of documents viewed during a run resulting for the agent's link selection behaviour. The path is represented as a directed graph where the nodes are the documents viewed by the agent and the edges are the links traversed from document to document.

$$RED_i = \frac{DupVerts_i}{TotalVerts_i}$$

Where:
RED_i = redundancy score for click path i
$DupVerts_i$ = number of duplicate vertices in click path i
$TotalVerts_i$ = total number of vertices in click path i

Equation 1. Redundancy Measure.

Coverage (see Equation 2) measures the overlap, at the node-to-node level, between click paths. Coverage does not look at the order in which edges are traversed, but at the number of single edge sets shared between in both agents' click paths. The coverage ratio is calculated by computing the intersection of unique edges between two click paths. The smaller number of unique edges from either click path is used to divide this intersection (this expression is analogous to the ratio of the intersection to the union, which corresponds to the “content” model of similarity as defined by Gregson 1975). This produces a value from 0 to 1.

$$COV(A_i, B_i) = \frac{UniqEdges_{A_i} \cap UniqEdges_{B_i}}{\min(UniqEdges_{A_i}, UniqEdges_{B_i})}$$

Where:
$COV(A_i, B_i)$ = coverage score between click path A and B for query i
$UniqEdges_{Ai}$ = number of unique edges for agent's A click path for query i
$UniqEdges_{Bi}$ = number of unique edges for agent's B click path for query i

Equation 2. Coverage Measure.

The other metric that is used to measure similarity between click paths is the Longest Common Click-path (LCC) ratio (see Equation 3). This metric calculates the longest subsequence of viewed documents (including duplicates) shared between two click paths. This measure is reported as a ratio of the longest shared sub-graph divided by the longest possible shared sub-graph, which is the length of the shortest click path. A similar measure was used in a study by Banerjee and Ghosh (2001) where path similarity was measured based on the Longest Common Subsequence (LCS). Schluep *et al.* (1998) used the intersection of two state transition vectors, normalized by the smaller vector, to measure similarity between users interacting with two database interfaces. The study investigated three methods for computing similarities between user interaction strategies (correlation, intersection, and exclusion) and found that the intersection (analogous to LCC) coupled with multi-dimensional scaling best clustered the subjects into the three groups of interaction strategies observed by the researchers. Pitkow and Pirolli (1999) also used the longest repeating subsequence to produce models of, and to predict, web surfing paths.

$$LCC(A_i, B_i) = \frac{LongestSharedRoute(Path_{A_i}, Path_{B_i})}{\min(PathLen_{A_i}, PathLen_{B_i})}$$

Where:
$LCC(A_i, B_i)$ = longest common click path between click path A and B for query i
$Path_{Ai}$ = click path for query i for agent A
$Path_{Bi}$ = click path for query i for agent B
$PathLen_{Ai}$ = number of vertices in the click path for query i for agent A
$PathLen_{Bi}$ = number of vertices in the click path for query i for agent B
$LongestSharedRoute$ = longest sequence of vertices and edges shared between paths

Equation 3. Longest Common Click-Path (LCC).

Although both coverage and LCC ratios measure the sequence of documents that is viewed based on link selection behaviour, they measure slightly different aspects of the graph. Coverage looks at similarity in edges traversed. For example, agent A might have the following path sequence: 1-2-4-9-7-2-3 and agent B might have the following sequence: 2-1-2-4-9-7-5-4. The number of common edges between the agents is 4 (1-2, 2-3, 2-4, and 9-7). The coverage metric is not concerned with the ordering of the edges, whereas LCC is concerned with the ordering of the edges. It looks at the longest common sequence of edges between the two agents. In this case, the sequence would be 1-2-4.

4.2 Apparatus

The experiment was conducted using the SIIRS simulator. The simulator parameters, either agent specific or simulator specific, remained constant for all simulation runs. For this experiment, the query tail parameter was set to 2, 3, and 4. The newness parameter was set to 7, 10, and 13.

Additionally, the Managing Gigabytes (MG) text retrieval system (version 1.2) was used as the IR System for all simulation runs in this experiment. The system was configured to use free text input and to rank the output based on its internal document similarity measure.

4.3 Procedure

Each search agent was given the task of searching the document collection for a maximum of 250 clicks (successive link selections) or until an internal stopping condition was reached. The simulator run consisted of a specific agent instance (i.e., one instance for each agent class, therefore six in total) combined with an initial query. The initial query was used to start the search session. The experiment consisted of 25 queries selected at random from NIST's TREC topics 51-100. Buckley and Voorhees (2000) suggested that 25 queries is the minimum number of queries for stable results in a batch mode experiment.

The queries were assigned either low, medium, high difficulty ratings. Query difficulty was calculated after the simulation runs were completed. A query's difficulty rating was determined by using the average query Harmonic mean (F-Score) to rank the queries. Once the queries were ranked, the set was partitioned into quartiles. The queries in the quartile with the highest F-Score values were assigned the low difficulty rating. Similarly, the queries in the quartile with the lowest F-Score values were assigned the high difficulty rating. The queries in the middle two quartiles were assigned the medium difficulty rating.

The document collection used for the experiment was the Wall Street Journal collection. The collection articles were taken from NIST's Text Research Collection Volume 1 and 2 compact discs. The collection consisted of 173,252 articles. Articles, on average were 2,793 characters in length. There were, on average, 439 terms per document and of those 172 terms were unique, and 174 were stop words

4.4 Results

The analysis of the data was divided in to two parts: the first part for the performance measures and the second part for the behavioural measures. The results for the performance measures will be presented first, followed by the results for the behavioural measures.

4.4.1 Search Agent Performance Measures

A multivariate analysis of variance (MANOVA) was performed with Search Agent, Query Tail Size, Newness, and Query Difficulty as independent variables and Recall, Precision, Average Interpolated Precision, Harmonic mean (F-Score), and CTFRD as dependent variables. When investigating the precision and recall scores for the three query difficulty groups (low, medium, and high), it was discovered that all of the combinations of agent, query tail size, and newness performed poorly for queries in the high query difficulty group. Due to this fact, further analyses of the performance-based measures focused only on the low and medium query difficulty groups.

There was an interaction between newness and query difficulty. Table 1 shows the dependant performance measures for the low and medium query difficulties for the three newness values (7, 10, and 13). Precision and average interpolated precision differed significantly (F[1,918]=9.331, p<.001 and F[1,918]=5.804, p=.003 respectively). Harmonic mean was also significant for this interaction (F[1,918]=4.063, p=.018). It can be seen, in Table 1, that newness had a greater effect on low difficulty queries in terms of precision and Harmonic mean, particularly for a newness value of seven. There was no significant difference between the recall scores for the query difficulty groups. The CTFRD column shows that the mean for the low query difficulty queries was one for all newness levels (i.e., in low difficulty queries a relevant document was always found on the first try). There was an inverse relationship between CTFRD score and newness value for queries of medium difficulty. As the newness values increased, the CTFRD score decreased.

There was also a significant main effect found for newness as measured by precision (F[1,918]=9.804, p<.001) and average interpolated precision (F[1,918]=6.101, p=.002). Figure 3 shows the overall precision, recall, Harmonic mean, and average interpolated precision means for newness. Overall agent performance, in terms of precision and average interpolated precision, decreased as newness increased.

Table 1. Performance for newness for low and medium query difficulty levels.

Newness	Precision*		Recall		F-Score**		Avg. Int. Prec.*		CTFRD	
	Low	Med.	Low	Med.	Low	Med.	Low	Med.	Low	Med.
7	0.320	0.045	0.104	0.013	0.109	0.014	0.416	0.041	1	116.0
10	0.253	0.048	0.110	0.015	0.095	0.017	0.370	0.044	1	105.5
13	0.214	0.043	0.108	0.019	0.087	0.019	0.334	0.040	1	93.4

* - significant difference (p<.005) and ** - significant difference (p<.05)

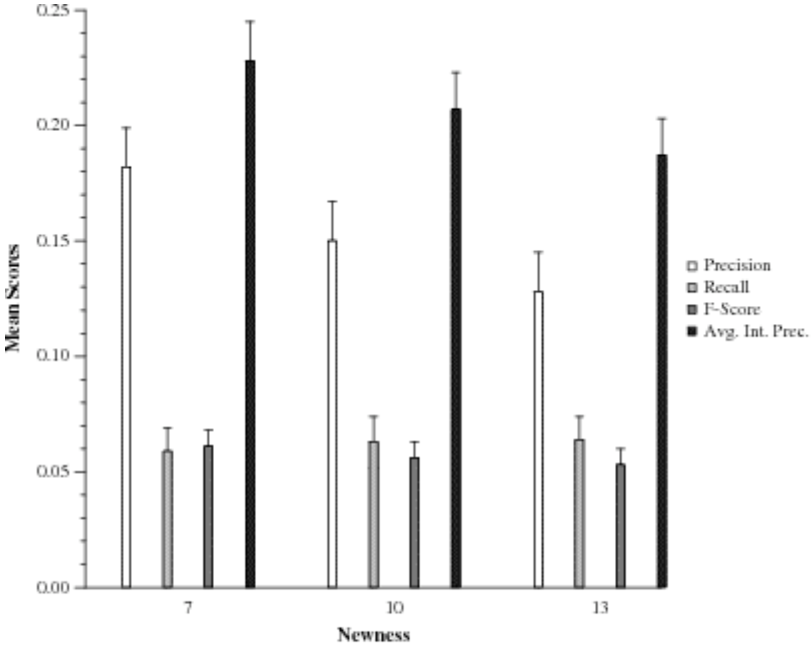

Figure 3. Effect of newness on agent performance.

The next significant interaction was between query tail size and query difficulty. Table 2 shows the various performance scores for the low and medium difficulty levels and the three query tail sizes (2, 3, and 4). The nature of the interaction can be seen by investigating the relative improvements in precision and average interloped precision for low and medium queries for the three levels of query tail size. The performance increased as query tail size increased for the low difficulty queries (improvement from 2 to 3 was 0.012 and 3 to 4 was 0.075 for precision), whereas there was very little improvement in performance between the difference values of query tail size for the medium difficulty queries (improvement from 2 to 3 and 3 to 4 was 0.009).

The precision (F[1,918]=4.913, p=.008), Harmonic mean (F[1,918]=3.705, p=.025), and average interpolated precision (F[1,918]=5.586, p=.004) measures were significantly affected by the interaction between query tail size and query difficulty. For both precision and Harmonic mean, a query tail of 4 performed the best, followed by 3 and 2 (see Table 4.2). An analysis was also conducted without query difficulty as a factor.

The main effect of query tail size also showed the same rankings for precision, recall, and Harmonic mean. There was a significant main effect for query tail size on precision (F[1,918]=10.190, p<.001), Harmonic mean (F[1,918]=6.441, p=.002) and on average interpolated precision (F[1,918]=6.101, p=.002). Figure 4 shows the overall means for precision, recall, Harmonic mean, and average interpolated precision for the three query tail sizes.

There were significant main effects for search agent in terms of precision (F[1,918]=60.204, p<.001), Harmonic mean (F[1,918]=21.865, p<.001), and average interpolated precision (F[1,918]=52.260, p<.001). Figure 5 presents the performance for the six search agents. As can be seen in the figure, the agents appear to form two groups.

The agents are separated based on their underlying link selection mechanism: random-based versus similarity-based.

Table 2. Performance for query tail size for low and medium query difficulty levels.

Query Tail Size	Precision**		Recall		F-Score**		Avg. Int. Prec.*		CTFRD	
	Low	Med.	Low	Med.	Low	Med.	Low	Med.	Low	Med.
2	0.229	0.036	0.096	0.014	0.084	0.013	0.340	0.033	1	100.3
3	0.241	0.045	0.106	0.017	0.093	0.018	0.353	0.042	1	109.6
4	0.316	0.054	0.120	0.016	0.114	0.018	0.427	0.050	1	104.9

* - significant difference (p<.005) and ** - significant difference (p<.05)

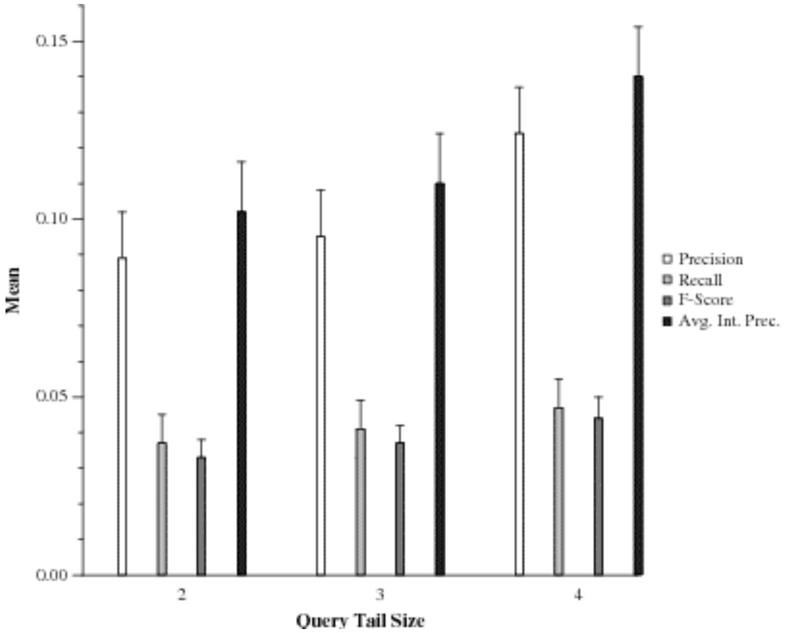

Figure 4. Effect of query tail size on agent performance.

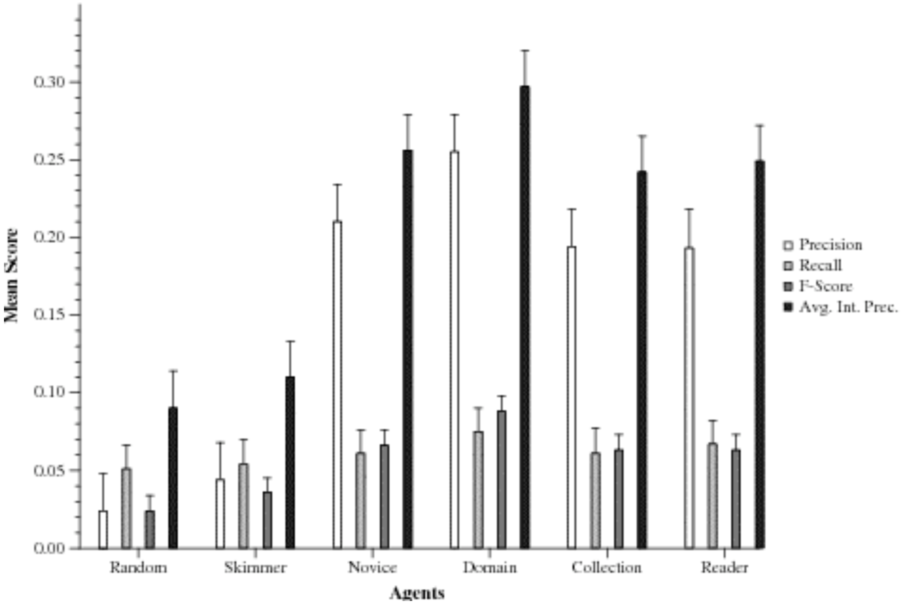

Figure 5. Overall mean performance measures for all agents.

4.4.2 Search Agent Behaviour Measures

The remainder of the results section focuses on the search agent behaviour as captured by the measures of redundancy, coverage, and longest common click path (LCC).

A three-way ANOVA was performed on the redundancy measure scores for all six search agents. The three factors were agents (6 levels), newness (3 levels), and query tail size (3 levels). There were significant main effects for agent (F[1,1296]=1053.598, p<.001), newness (F[1,1296]=62.026, p<.001), and query tail size (F[1,1296]=22.488, p<.001). Table 3 shows the redundancy ratios for the six search agents. A lower redundancy score indicates that there were fewer re-viewings of documents during the search session. It can be seen that the random-based agents had lower redundancy scores as compared to the similarity-based agents. Both the Random and Skimmer agents had the lowest redundancy score of 0.140. All of the similarity-based agents had a relatively high number of re-viewings and conversely had a small number of unique viewings. Although this measure shows that the similarity-based agents viewed a small number of unique documents, it does not reveal whether or not the link selected from a document was the same each time.

Table 3. Mean redundancy scores for all agents.

Agent	Mean Redundancy
Random	0.140
Novice	0.755
Skimmer	0.140
Domain	0.797
Collection	0.809
Reader	0.818

The mean redundancy scores were .637, .521, and .581 respectively for the three levels of newness (7, 10, and 13). For the three levels of query tail (2, 3, and 4), the mean redundancy scores were .539, .581, and .609, respectively. Given that these two parameters contribute to "randomness" in the agents' search behaviour, it was to be expected that the newness value of 13, which has the most random effect, would also have the lowest redundancy score. A low redundancy score was also expected for the smallest query tail size (2).

As mentioned above, although redundancy captures agent revisits to documents, what it does not reveal is whether a click path created by the agent's link selection strategy is the same each time an agent revisits a document. To address this question, the metrics of coverage and longest common click path were used.

The pattern of coverage and LCC ratios across agents and conditions was investigated by studying the patterns of similarities in coverage and LCC within agents for different pairwise combinations of query tail size and newness. Two 9x9 similarity matrices were constructed for each search agent based on the experimental data. One matrix was calculated for the coverage ratio and the other matrix was calculated for the LCC ration. A cell represented either the coverage/LCC score computed between runs (specific agent/simulation configuration) for a given agent. Each row and column in the matrix represented a combination of newness (values: 7, 10, and 13) and query tail size (values: 2, 3, and 4) resulting in nine combinations. The matrices were symmetrical along the diagonal. Multi-dimensional scaling was performed on all six agent's matrices at once using individual difference scaling (INDSCAL), as described in Carrol and Chang (1970). The individual differences chart shows relationships between agents within a space. The stimulus chart can be used to ascertain how parameters map to dimensions in the solution space.

Figure 6 shows the individual differences scaling (two-dimensional solution) based on the Euclidean distances between the six search agents for the coverage metric. The Domain, Collection, and Reader agents formed one cluster while the Random and Skimmer formed another cluster. From an investigation of the one-dimensional plot of the individual differences, the Novice agent was found not to match with either of these two clusters. This separation between groupings of the agents was also found in the analysis of the performance measures. Table 4 provides the mean coverage scores for all agents.

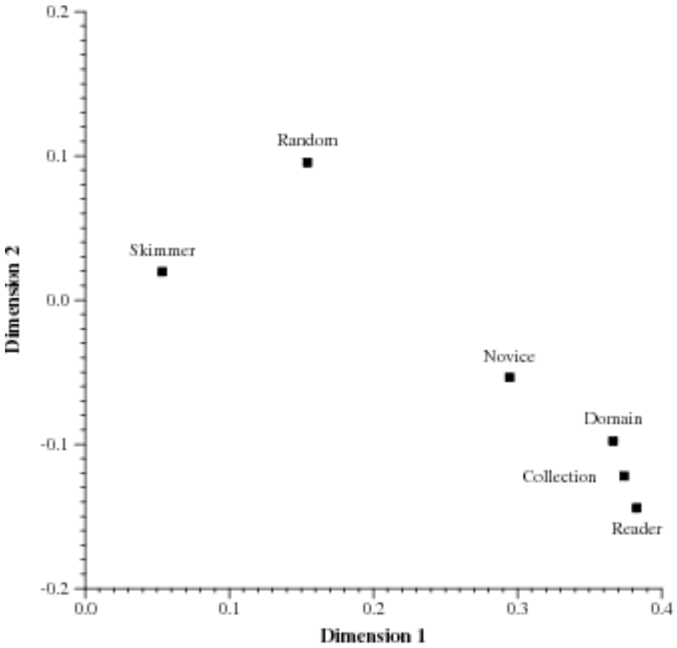

Figure 6. Individual difference between agents based on coverage.

Table 4. Mean coverage scores for all agents.

Agent	Mean Coverage
Random	0.133
Novice	0.528
Skimmer	0.138
Domain	0.496
Collection	0.443
Reader	0.463

To further investigate the agent's link selection behaviour, an analysis was conducted which compared the longest common click path (LCC) ratios for the agents paired with the various combinations of newness and query tail size. The analysis started by using multi-dimensional scaling on the six stacked 9x9 matrices for the agents. The individual difference based on the Euclidean distances produced the derived subject weight graph found in Figure 7. Again, the agents formed two main clusters: Random and Skimmer, versus Domain, Collection, and Reader. The Novice agent did not belong to either cluster.

Table 5 details the mean LCC scores for the six agents. The ordering of the agents was maintained when the means were calculated for the three levels of query difficulty. The various combinations of newness and query tail did not affect the random-based agents. For the similarity-based agents, the combinations of newness levels of 10 and 13, and a query tail size of 2, produced the largest LCC scores. The Novice agent behaviour was most consistent in terms of LCC, averaged over all combinations.

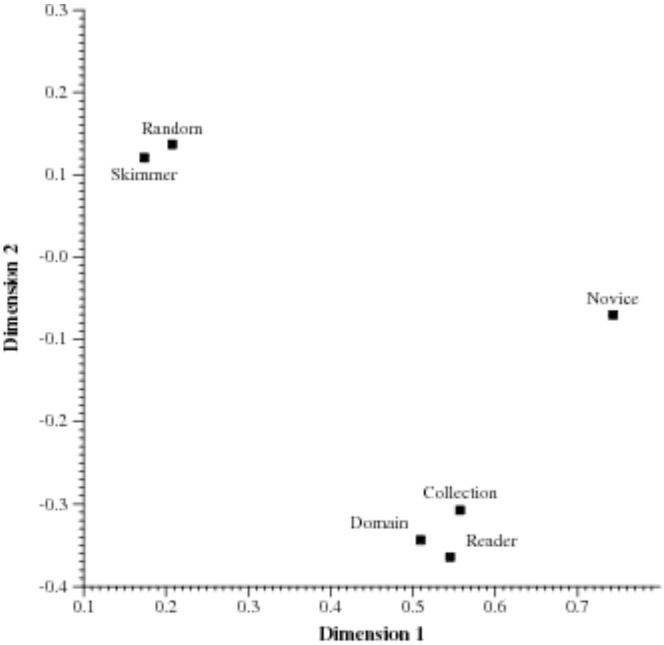

Figure 7. Individual difference between agents based on LCC.

Table 5. Mean LCC scores for all agents.

Agent	Mean LCC
Random	0.119
Novice	0.247
Skimmer	0.125
Domain	0.167
Collection	0.146
Reader	0.155

4.5 Discussion

The analysis of the both the performance and behavioural measures showed significant differences in how an agent performs due to different combinations of query difficulty, newness, and query tail size, as discussed in the following sections.

4.5.1 Search Agent Performance Measures

The analysis of the attribute-based performance measures of precision, Harmonic mean, and average interpolated precision showed significant differences for the three main effects of Newness, Query Tail Size, and Agent. The significant interactions were for Query Difficulty and Newness, Query Difficulty and Query Tail Size, and Query Difficulty and Search Agent.

The newness value controls how soon a document can be re-viewed by an agent. Thus as the newness value increased, the chance of viewing a new, unseen document also increased since the ability to re-view old documents is reduced. Therefore, the ordering of the CTFRD values for newness is to be expected (see Table 1).

The query tail size affects the dynamic hypertext mechanism, more specifically the terms used for links, and it affects how the query that is sent to the IR System is built. The query tail size directly relates to the number of previous link selections made by the agent that the dynamic hypertext mechanism uses. A small query tail size tended to cause the system to “jump” from topic to topic too quickly. A large query tail size had the opposite effect where there was a diminished tendency for the system to “switch” to a new topic. Thus, it can be seen that the smaller query tail size of two created somewhat more topic instability, allowing the agents to explore more documents, thereby increasing the chances of viewing a relevant document.

As the query tail size increased, so did the mean CTFRD score for both query difficulty levels. As the query tail size increases, it has the effect of “stabilizing” the search terms from iteration to iteration, allowing previous search terms to stay in the INFERRED query for longer. This stabilizing effect reduces randomness in the agent's search pattern (click-path). Thus, if an agent starts a search with a non-relevant document and a larger query tail size, it will take longer for the agent find a relevant document.

As mentioned, there was a significant interaction between agent and query difficulty. The agents formed two main groups that matched the two main agent types (random-based and similarity-based link selection mechanisms) developed for the simulator. Interestingly, the two random-based agents evidently found relevant documents sooner for medium difficulty queries as opposed to the similarity-based agents. This was the only aspect of performance in which these agents did well as compared to the other agents.

4.5.2 Search Agent Behaviour Measures

Overall, the agents differed in terms of their behaviours compared to one another and in terms of their interaction with the simulator parameter of newness and the dynamic hypertext mechanism control of query tail size. The analysis of the behavioural measures showed the same pattern as found in the performance measures. The query tail size, especially a query tail size of 2, had the most effect on the agent's behaviour.

The analysis of redundancy showed that agents were affected by changes in the size of the query tail. For the Random, Skimmer, Novice, and Reader agents, as the size of the query tail increased, so did the agent's redundancy scores. This behaviour is to be expected since the stability (difference in search terms) of the agent's query is improved as the query tail increases. The Novice showed an extreme example of this behaviour with the largest difference (0.144) between redundancy scores for query tail sizes of 2 and 4. All of the other similarity-based agents had differences roughly half the size of the Novice difference. This result can be attributed to how the Novice agent selects links. It uses its previous link selections as a gauge to rank links in a document. Therefore, as more past links are considered due to the increase in query tail size, the Novice agent's behaviour showed that the agent was able to focus on a core set of documents then the other agents. This ability of an agent to focus on a core document set means that the documents in the core set are more frequently re-viewed, thus increasing the agent's redundancy score. Conversely, redundancy decreased for the Collection agent when the size of the query tail was increased. It had a negative difference (-0.24) between scores for query tails 2 and 4.

The Collection agent selects jump documents by comparing a document to the current to past link selections. If the similarity between the document and the past links is above a specified threshold, the document is considered a jump document. By improving the stability of the past link selection from iteration to iteration in the Collection agent's search session; it also improves the quality of the jump document. There was a significant difference (p=.012) between the number of jump documents found by the Collection agent for the three levels of query tail size. As the query tail increased, so did the number of jump documents found by the agent, suggesting that the agent was better able to judge the similarity of a document as the stability of the query increased.

Redundancy scores for both the Random and Skimmer agents increased as query tail size increased. The similarity-based agents (Novice, Domain, Collection, and Reader) had varying responses to the change in query tail size. For a query tail size of 2, the Novice had the lowest redundancy score of the four similarity-based agents. As the query tail size increased, the difference between the similarity-based agents decreased. In contrast to its effect on the other agents, increasing query tail size tended to reduce redundancy for the Collection agent.

As found in the analysis for redundancy, the analysis for the agent's coverage and LCC scores showed the same agent groupings of random-based vs. similarity-based agents. Overall, the random-based agents performed less consistently with respects to the LCC metric. This is to be expected since LCC measures the length of the longest common path for paired agent combinations of various query tail and newness settings. Since these agents randomly select from a set of links, these parameters have little effect. The LCC scores were not zero since dynamic hypertext is essentially a sentence level relevancy feedback mechanism. Such a mechanism has the benefit of guiding the searcher to relevant document and thus random-based agents still manage some consistency between configurations.

5 Conclusions

Most automated evaluation tools focus primarily on the system oriented measures of precision and recall. Although these measures can provide valuable insight into the performance characteristics of the search process, they are typically used in a batch evaluation manner and provide only a high-level overview of an information retrieval system. When these measures are applied to interactive information retrieval systems, experiments with human participants are required. Often such experiments reveal no differences among systems or configurations, but the subjects or the query tasks instead show significant differences. Perhaps because of these difficulties, past research on interactive information retrieval has tended to focus primarily on the system being investigated, rather than on user attributes and search strategies.

As more research in the field of information retrieval is focused on evaluation, there is a need to consider the role of the user in the search process. Although most models of information exploration are built around the searcher, evaluation models and methods have been slow to incorporate the user, and to account for individual differences between users.

The evaluation framework presented is an attempt to incorporate individual differences within an automated evaluation environment. Although the search strategies presented do not capture every facet of the class of searcher they represent, they provide the means to test the potential of the evaluation framework and its embodiment within the SIIIRS simulator. Addressing the research question regarding differences between search strategies, the simulation of the search agents worked well. The analyses of the performance and behavioural measures showed significant differences between the agents. Agents showed strong affinity to other like-implemented agents (random-based vs. similarity-based agents). Query difficulty was also shown to have an effect on agent performance. Additionally, similarity-based agents were sensitive to newness and query tail size parameters.

Due to inconsistencies between subjects, document collection, system configurations, etc., it is difficult to compare system attribute measures such as precision and recall across different experiments. The application of a framework in the form of the SIIIRS simulator can be used to provide a consistent method of analyzing the performance between IIR systems. The experiment has shown that the SIIIRS simulator can be used to investigate the affects of changes to the agent and simulator configuration parameters on performance (in terms of performance measures, such as precision and recall) and search behaviour (in terms of measures, such as redundancy, coverage, and LCC).

The framework can be utilized by the research to investigate a number of aspects related to interactive information retrieval. First, from the perspective of automatically evaluating algorithmic changes in terms of individual differences in search strategies, the framework and SIIIRS can be used to explore various forms of correlation between human users' perceptions of an IIR system and performance. The issues surrounding the selection of the various simulator parameters illustrate the importance of this research. For example, values used for the query tail size were based on previous studies involving human subjects. In those studies only the users' qualitative responses were measured (i.e., users reported how they felt about the dynamic hypertext system when different values for the query tail size were used). Given the results from the above experiment concerning query tail size, a human subjects study should be conducted which measures quantitatively the affect of the query tail size on the utility of the dynamic hypertext mechanism.

The concept of correlation between perception and performance is an area where the SIIIRS simulator could be of value by focusing and reducing the number of human experiments needed, since the simulator can be run multiple times to try to match the experience of human subjects. Once a match is found, the parameters that contribute to the match can be investigated further using human subjects to assess the effect of changes to system parameters.

Another aspect that the simulator can be of value is in the application having the software search agents provide search assistance for a user (e.g., by providing configuration or link selection recommendations). The results from previous simulator analyses could be used to assist users in selecting an appropriate IIR system and/or system configurations for their task, search strategy, and interaction style. For example, a user could pre-select a similar search strategy to their own or the system could monitor their interactions and adapt the presentation of information or the underlying IR tools to match that user's search strategy.

In conclusion, the evaluation framework and in particular the SIIRS simulator, allows researchers to conduct many exploratory studies which can help to narrow the focus of future human subject studies by showing which differences in information exploration style and functionality are likely to produce significant differences in future studies with human subjects.

Acknowledgments

Our approach to dynamic hypertext has been influenced by the work on dynamic hypertext that Gene Golovchinsky carried out while doing his PhD research at the Interactive Media Laboratory at the University of Toronto. This research was supported by CITO, a division of the Ontario Centres of Excellent, and by an NSERC Discovery grant to the second author.

References

Banerjee, A. and Ghosh, J. (2001) Clickstream clustering using weighted longest common subsequences. In *Proceedings of the 1st SIAM International Conference on Data Mining: Workshop on Web Mining.*

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

Bates, J. M. (1990) Where should the person stop and information search interface start? *Information Processing & Management*, Vol. 26, No. 5, 575–591.

Beaulieu, M., Robertson, S., and Rasmussen, E. (1996) Evaluating interactive systems in TREC. *Journal of American Society for Information Science*, Vol. 47, No. 1, 85–94.

Belkin N. J., Chang, S., Downs, T., Saracevic, T., and Zhao, S. (1990), Taking into account of user tasks, goals and behavior for the design of online public access catalogs. In *Proceedings of the 53rd Annual Meeting of the American Society for Information Science (ASIS'90)*, pp. 69–79.

Belkin, N. J., P. G. Marchetti, P. G., and Cool, C. (1993), BRAQUE: design of an interface to support user interactions in information retrieval. *Information Processing & Management*, Vol. 29, No. 3, 325–344.

Bhavnani, S. K. (2002) Domain-specific search strategies for the effective retrieval of healthcare and shopping information. In *Proceedings of the SIGCHI conference on human factors in computing systems (CHI'02)*, pp. 610-611.

Bodner, R. C. (2005) Automated Evaluation of Hypertext Search Strategies. Unpublished PhD thesis, Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Ontario, Canada.

Bodner, R. C. and Chignell, M. H. (1999) ClickIR: text retrieval using a dynamic hypertext interface. In *Proceedings of the Seventh Text REtrieval Conference (TREC-7)*, pp. 573–582.

Bodner, R. C., Chignell, M. H., and Tam, J. (1997) Website authoring using dynamic hypertext. In *Proceedings of Webnet'97*, pp. 59–64.

Bodner, R. C., Chignell, M. H., Charoenkitkarn, N., Golovchinsky, G., and Kopak, R. W. (2001) The impact of text browsing on text retrieval performance. *Information Processing & Management*, Vol. 37, No. 3, 507–520.

Borgman, C. L. (1989) All users of information systems are not created equal: an exploration into individual differences. *Information Processing & Management*, Vol. 25, No. 3, 237–252.

Brandt, D. S. and Uden, L. (2003) Insight into mental models of novice Internet searchers. *Communications of the ACM*, Vol. 46, No. 7, 133–136.

Broder, A. (2002) A taxonomy of web search. *SIGIR Forum*, Vol. 36, No. 6, 3–10.

Buckley, C. and Voorhees, E. M. (2000) Evaluating evaluation measure stability. In *Proceedings of SIGIR'00, the 23rd Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pp. 33–40.

Campagnini, F. R. and Ehrlich, K. (1989) Information retrieval using a hypertext-based help system. *ACM Transactions on Office Information Systems*, Vol. 7, No. 3, 271–291.

Card, S. K., Moran, T. P., and Newell, A. (1983) *The psychology of human-computer interaction*. (Hilsdale, NJ, USA: Lawrence Erlbaum Associates).

Carrol, J. D. and Chang, J. J. (1970) Analysis of individual differences in multidimensional scaling via an N-way generalization of 'Eckart-Young' decomposition. *Pyschometrika*, Vol. 35, 283–319.

Charoenkitkarn, N. (1996) The effect of markup-querying on search pattern and performance in large-scale text retrieval. Unpublished PhD thesis, Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Ontario, Canada.

Choo, W. C., Detlor, B., and Turnbull, D. (2000) Working the web: an empirical model of web use. In *Proceedings of the Hawaii International Conference on System Science*.

Cleverdon, C. W., Mills, J., and Keen, E. M. (1966) *An inquiry in testing of information retrieval systems (2 vols.)* (Cranfield, UK: Aslib Cranfield Research Project, College of Aeronautics).

Conklin, J. (1987) Hypertext: an introduction and survey. *Compute*r, Vol. 20, No. 9, 17–41.

de Bruijn, B., Holte, R., and Martin, J. (1999) An automated method for studying interactive systems. In *Proceedings of the 58th Annual Meeting of the American Society for Information Science (ASIS'99)*.

Fenichel, C. H. (1981) Online searching: measures that discriminate among users with different types of experience. *Journal of American Society for Information Science*, Vol. 32, No. 1, 23–32.

Goldstein-Hirsh, S. (1995) The effect of domain knowledge on elementary school children's search behavior on an information retrieval system: the science library catalog. In *Proceedings of the SIGCHI conference on human factors in computing systems (CHI'95)*, pp. 55–56.

Golovchinsky, G. (1997a) From information retrieval to hypertext and back again: the role of interaction in the information exploration interface. Unpublished PhD thesis, Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Ontario, Canada.

Golovchinsky, G. (1997b) Queries? Links? Is There a Difference? In *Proceedings of the SIGCHI conference on human factors in computing systems (CHI'97)*, pp. 407–419.

Golovchinsky, G., Chignell, M. H., and Charoenkitkarn, N. (1997) Formal experiments in causal attire: case studies in information exploration. *New Review of Hypermedia and Multimedia*, Vol. 3, 123–158.

Gregson, R. A. M (1975) *Psychometrics of Similarity* (New York, NY , USA: Academic Press).

Hancock-Beaulieu, M. (1990) Evaluating the impact of an online library catalogue on subject searching behaviour at the catalogue and at the shelves. *Journal of Documentation*, 46, 318–338.

Huberman, B. A., Pirolli, P. L. T., Pitkow, J. E., and Lukose, R. M. (April, 1998) Strong regularities in World Wide Web surfing. *Science*, 280, pp. 95–97.

Iivonen, M. (1995) Searchers and searchers: differences between the most and least consistent searchers. In *Proceedings of the 18th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR'95)*, pp. 149–157.

Ivory, M. Y. and Hearst, M. A. (2001) The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys*, Vol. 33, No. 4, 470–516.

Jeffries, R., Miller, J. R., Wharton, C., and Uyeda, K. M. (1991) User interface evaluation in the real world: A comparison of four techniques. In *Proceedings of the SIGCHI conference on human factors in computing systems (CHI'91)*, pp. 119–124.

Kessner, M., Wood, J., Dillon, R. F., and West, R. L. (2001) On the reliability of usability testing. In *Proceedings of the SIGCHI conference on human factors in computing systems (CHI'01)*, pp. 97–98.

Kleinberg, J. (1999) Authoritative sources in a hyperlinked environment. *Journal of the ACM*, Vol. 46, No. 5, 604–632.

Kuhlthau, C. C. (1991) Inside the search process: information seeking from the user's perspective. *Journal of American Society for Information Science*, Vol. 42, No. 5, 361–371.

Marchionini, G. (1989) Information-seeking strategies of novices using a full-text electronic encyclopedia. *Journal of American Society for Information Science*, Vol. 40, No. 1, 54–66.

Molich, R., Bevan, N., Butler, S., Curson, I., Kindlund, E., Kirakowski, J., and Miller, D. (1998) Comparative evaluation of usability tests. In *Proceedings of the UPA Conference*, (Washington, DC: Usability Professionals Association), pp. 189–200.

Molich, R., Thomsen, A. D., Karyukina, B., Schmidt, L., Ede, M., Van Oel, W., and Arcuri, M. (1999) Comparative evaluation of usability tests. In *Proceedings of the SIGCHI conference on human factors in computing systems (CHI'99)*, (Pittsburgh, PA: ACM Press), pp. 83–86.

Nielsen, J. (1990) The art of navigation through hypertext. *Communications of the ACM*, Vol. 33, No. 3, 296–310.

Nielsen, J. (1993) *Usability Engineering* (Boston, MA, USA: Academic Press).

Oddy, R. N. (1977) Information retrieval through man-machine dialogue. *Journal of Documentation*, Vol. 33, No. 1, 1–14.

Over, P. (1999) TREC-7 interactive track report. In *Proceedings of the Seventh Text Retrieval Conference (TREC-7)*.

Parunak, H. V. D. (1989) Hypermedia topologies and user navigation. In *Proceedings of Hypertext'89*, pp. 43–50.

Peck, V. A. and John, B. E. (1992) Browser-Soar: a computational model of a highly interactive task. In *Proceedings of the SIGCHI conference on human factors in computing systems (CHI'92)*, (Monterey, CA , USA: ACM Press), pp. 165–172.

Pirolli, P. and Fu, W. T. (2003) SNIF-ACT: a model of information foraging on the World Wide Web. In *Proceedings of the 9th International Conference on User Modeling*.

Pitkow, J. E. and Pirolli, P. (1999) Mining longest repeating subsequences to predict World Wide Web surfing. In *Proceedings of USITS'99: The 2nd USENIX Symposium on Internet Technologies & Systems*.

Saracevic, T. (1991) Individual differences in organizing, searching, and retrieving information. In *Proceeding of the 54th Annual Meeting of the American Society for Information Science (ASIS'91)*, edited J. Griffiths, pp. 82–86.

Saracevic, T. (1995) Evaluation of evaluation in information retrieval. In *Proceedings of the 18th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR'95)*, pp. 138–146.

Schluep, S., Fjeld, M., and Rauterberg, M. (1998) Discriminating task solving strategies using statistical and analytical methods. In *Proceedings of Cognition and Co-operation (ECCE9)*, pp. 121–126.

Smeaton, A. F., Keogh, G., Gurrin, C., McDonald, K., and Sodring, T. (2002) Analysis of papers from twenty-five years of SIGIR conferences: what have we been doing for the last quarter of a century? *SIGIR Forum*, Vol. 36, No. 2, 39–43.

Tam, J., Bodner, R., and Chignell, M. (1997) Dynamic hypertext benefits novices in question answering. In *Proceedings of the Human Factors and Ergonomics Society 41st Annual Meeting*, pp. 350–354.

Toms, E. G. (1998) Information exploration of the third kind: the concept of chance encounters. In *Proceedings of the CHI'98 Workshop on Innovation and Evaluation in Information Exploration Interfaces*.

Toms, E. G. (2000) Serendipitous information retrieval. In *Proceedings of the Workshop Information Seeking, Searching and Querying in Digital Libraries*, pp. 1–10.

Toms, E. G., Freund, L., Kopak, R., and Bartlett, J. C. (2003) The effect of task domain on search. In *Proceedings of CASCON 2003*, pp. 1–10.

Voorhees, E. (1998) Variations in relevance judgments and the measurement of retrieval effectiveness. In *Proceedings of the 21st Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98)*, pp. 315–323.

Waterworth, J. A. and Chignell, M. H. (1991) A model of information exploration. *Hypermedia*, Vol. 3, No. 1, 35–58.

West, R. L. and Emond, B. (2002) Can cognitive modeling improve rapid prototyping. Cognitive Science Technical Reports, Carleton University, Ottawa, Canada.

Xie, H. (2000) Shifts of interactive intentions and information-seeking strategies in interactive information retrieval. *Journal of American Society for Information Science*, Vol. 51, No. 9, 841–857.

Xie, H. (2002) Patterns between interactive intentions and information-seeking strategies. *Information Processing & Management*, Vol. 38, No. 1, 55–77.