Available online at www.sciencedirect.com

SCIENCE DIRECT®

Information Processing and Management 41 (2005) 909–928

INFORMATION PROCESSING & MANAGEMENT

www.elsevier.com/locate/infoproman

Seeking and implementing automated assistance during the search process

Bernard J. Jansen

School of Information Sciences and Technology, The Pennsylvania State University, University Park, PA 16801, USA

Received 20 October 2003; accepted 16 April 2004
Available online 15 July 2004

Abstract

Searchers seldom make use of the advanced searching features that could improve the quality of the search process because they do not know these features exist, do not understand how to use them, or do not believe they are effective or efficient. Information retrieval systems offering automated assistance could greatly improve search effectiveness by suggesting or implementing assistance automatically. A critical issue in designing such systems is determining when the system should intervene in the search process. In this paper, we report the results of an empirical study analyzing when during the search process users seek automated searching assistance from the system and when they implement the assistance. We designed a fully functional, automated assistance application and conducted a study with 30 subjects interacting with the system. The study used a 2G TREC document collection and TREC topics. Approximately 50% of the subjects sought assistance, and over 80% of those implemented that assistance. Results from the evaluation indicate that users are willing to accept automated assistance during the search process, especially after viewing results and locating relevant documents. We discuss implications for interactive information retrieval system design and directions for future research.

Keywords: Automated assistance; Intelligent information retrieval systems; Explanation systems; Contextual help; Adaptive interfaces; Implicit feedback

1. Introduction

Considerable research has investigated automated searching assistance in hopes of resolving some of the issues users have when interacting with information retrieval (IR) systems. These searching issues include failure to find sufficient relevant information and not understanding searching features of the system.

E-mail address: jjansen@acm.org

0306-4573/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ipm.2004.04.017

Although many IR systems contain advanced features that may improve the IR process, users seldom utilize these features or have difficulty implementing them. Automated assistance systems attempt to assist the user during the search process by either executing search tactics for or offering assistance to the user in order to locate relevant information. The success of these intelligent systems depends in part on their ability to generate appropriate advice or action at opportune times.

There has been little empirical research into when in the search process to provide automated assistance. Thus, it is not clear whether or to what extent that automated assistance is of actual benefit to searchers at various stages of the information searching process. We also do not know when searchers actually desire assistance from the system. The research results presented in this article address the frequency of user interaction with automated assistance, when they view this assistance and when they implement the assistance. This knowledge can improve the design of future IR systems by personalizing and timing assistance to improve the effectiveness of the search process.

We begin with a review of literature concerning intelligent IR systems, explanation systems, contextual help systems and intelligent IR interfaces. Using various but related methods, these systems attempt to aid the searcher during the searching process. We then provide a short description of the automated assistance system we developed and utilized in an empirical study. Following this, we discuss the evaluation identifying the interactions during the search process between the users and automated assistance. We present the results of our evaluation and the implications for IR system design, along with a discussion of directions for future research.

From a synthesis and analysis of the literature presented in the next section, we provide a general definition of automated assistance. Automated assistance as a concept within information searching (IS) is not clearly articulated despite the significant body of research on IS models (e.g., Beaulieu, 2000; Belew, 2001; Belkin, Cool, Stein, & Theil, 1995; Brajnik, Guida, & Tasso, 1987; Ingwersen, 1992; Kuhlthau, Spink, & Cool, 1992; Saracevic, 1996) and IS within hypertext environments (Choo & Turnbell, 2000; Marchionini, 1995), along with search tactics (e.g., Bates, 1979, 1989; Belkin, Oddy, & Brooks, 1982) and user studies (e.g., Siegfried, Bates, & Wilde, 1993). Discussing explanation systems, Johnson and Johnson (1993) note that few researchers have defined "explanation". Examining intelligent interfaces, Mizzaro (1996) notes that there is no theoretical foundation for the functionalities of intelligent interfaces for IR, and Bates (1990) argues for searcher systems that are inline with searcher preferences.

Within the system development area, researchers generally do not define automated assistance, or they present a description that is inline with a particular system. Meadow, Hewett, and Aversa (1982b) state that their system can "tell when the human makes a mistake, provide supporting information, or offer, general advice on how to proceed." Belkin (1988, p. 138) defines explanation as "primarily prompted justification of intermediary activities, or unprompted explanation of system capabilities and characteristics." Johnson and Johnson (1993, p. 159) define intelligent interfaces as having the "ability to assist users in their tasks and can be seen to display context sensitive behavior." Oakes and Taylor (1998, p. 645) state that automated assistance "assist the nonexpert user in the formulation of ... search statements." Taking a script-based approach, Belkin et al. (1995, p. 10) say, "these scripts would characterize the most usual, or most effective, or in some sense standard means by which the user and the rest of the system interact." Mizzaro (1996) presents a matrix of functions that intelligent interfaces for IR typically possess. Belkin et al. (2001, p. 1) sees intelligent IR as "the machine (or program) doing something for the user, or the machine (or program) taking over some functions that previously had to be performed by humans (either user or intermediary)."

We believe that these and similar definitions of automated assistance are narrowly focused on a particular methodology, system, or viewpoint. Therefore, one cannot easily integrate the concept of automated assistance into existing models of IS. As information systems become more active participants in the searching process, models of this process must begin including system actions and responses. What is needed is a more general definition that is not system based and is inline with existing IS models. With this in mind, we define automated assistance as:

> Automated assistance is a temporal, goal-driven dialogue of expressions, actions or responses by an IR system with the aim of improving the information searching experience for the user as measured by some external metric.

A view of automated assistance containing these elements of evaluation, dialogue and time will facilitate the incorporation of automated assistance into IS models. Many IS models view the interaction between a searcher and system as a dialogue (Saracevic, 1997) with a temporal element (e.g., Beaulieu, 2000; Belew, 2001; Belkin et al., 1995; Brajnik et al., 1987; Ingwersen, 1992; Kuhlthau et al., 1992; Saracevic, 1996). External metrics measuring the effectiveness of the IS process are usually relevance-based (Saracevic, 1975), such as precision, recall (Salton & McGill, 1983) or shortest path (Cooper, 1968).

2. Literature review

The IS process is a special form of human computer interaction, with its own unique operational tasks (Beaulieu, 2000), and IR systems are a special form of information systems. People use these IR systems in an attempt to locate information relevant to their information need. Many searchers have difficulty effectively utilizing IR systems (Jansen, Spink, & Saracevic, 1998; Yee, 1991). Issues include finding appropriate query terms, retrieving too many results, not retrieving enough results, and retrieving zero results (Yee, 1991), among many others. These issues occur across the spectrum of IR systems, including online public access catalogs (Peters, 1993) and Web systems (Jansen et al., 1998). Although there has been considerable research and development incorporating advanced searching features into these IR systems, users generally do not use these features (Hunter, 1991) and have problems with them when they do (Jansen et al., 1998). Researchers have referred to systems designed to assist the user with searching issues and to better utilize advanced searching methods by a variety of names, including intelligent IR systems, explanation systems, intelligent interfaces, agent-based systems, contextual help systems, recommender systems, and relevance feedback systems. We collectively refer to all of these as *automated assistance* systems.

Meadow, Hewitt, and Aversa (1982a) and Meadow et al. (1982b) present one of the first accounts of a system using contextual help. The system provides searching instructions and diagnostic help. Croft and Thompson (1986) developed a user modeling system where the user supplied a natural language query or relevant document as a seed. Chen and Dhar (1991) developed a system for key word selection and thesaurus browsing. Oddy and Balakrishnan (1991) developed a networked–modeled system where an approximately one million node-and-edge network represented 10,000 document abstracts. In what appears to be the first published use of the phrase *automated assistance* in the IR literature, Oakes and Taylor (1998) designed an automated assistance system for pharmacology offering query formulation options.

In efforts using new interfaces, researchers (Brajnik et al., 1987) implemented an adaptive IR interface that utilizes natural language queries. OAKDEC (Meadow, 1988) is a front end to a database management system that suggested to users what searching tactic to employ. Also utilizing intelligent interfaces, Gauch and Smith (1993) developed an expert system interface for a rudimentary IR system. In one of the few user evaluations of these types of systems, the researchers noted no improvement in precision, but they did note a reduction in the number of queries to achieve comparable results.

Herlocker, Konstan, and Riedl (2000) examined methods to design intelligent systems. Within the field of automated collaborative filtering, the researchers examined the optimal degree of transparency for systems offering automated assistance. They concluded that automated assistance could be a valuable component of a system. However, their user evaluation did not indicate any improvement in system performance. Brajnik, Mizzaro, Tasso, and Venuti (2002) take a collaborative coaching approach in designing the FIRE system.

Focusing on Web systems, Google (http://www.google.com) offers spelling assistance with a *Did you mean* function. Alta Vista (http://www.altavista.com) offers spelling assistance, also with a *Did you mean* function, and term relevance feedback with its *Prisma* feature. Middleton, Roure, and Shadbolt (2001) investigate the issue of capturing user information preferences in the hypermedia environment. The researchers took the approach of unobtrusively monitoring users' browsing behaviors. They then used a machine learning approach coupled with an ontology representation in an attempt to extract user information preferences. Their system calculated a correlation between browsed Web pages and information topics, calculating a topic history weights.

Using agents, several researchers have explored intelligent IR systems for the Web, including Letizia (Lieberman, 1995) and Alexa (Kahle, 1999) to aid in the browsing process. ResearchIndex uses agents to recommend articles based on a user profiles (Lawrence, Giles, & Bollacker, 1999). Chen, Meng, Fowler, and Zhu (2001) developed an intelligent Web meta-indexer for Web searching.

From a review of the literature, it is apparent that there has been much work into developing IR systems that offer some type of automated assistance. However, there has been little evaluation of these systems. Evaluation is absolutely critical in order to understand when and how searchers utilize these systems during the searching process. From this understanding, one can develop systems that provide the proper type of assistance, offer it when it is most beneficial to or at least desired by the user, and also identify system features that are not helpful to users.

In this research, we developed a system that provides personalized and timed automated assistance for searchers using IR systems. In an evaluation, we then analyzed the user interactions with the automated assistance in order to investigate when in the IS process users are open to intervention by the system.

3. Research questions

In our research, we are interested in improving the search experience for the user by developing automated assistance systems that provide the proper type of assistance and offer it during the search process when it is most beneficial to the user. Our rational is that the user will be more open to assistance from the system and may then engage some of the advanced searching features that current IR systems offer. In order to achieve these goals, we investigate three research questions, which are:

a. How often do users seek and implement automated assistance in the search process?
b. Where in the search process do users seek automated assistance?
c. Where do users implement automated assistance in the search process?

4. System development

We designed and developed a software application to integrate with a variety of existing IR systems. This application gleans information solely from typical user–system interactions during the search process, using these interactions to determine what assistance to provide. In this respect, our approach is similar to previous work of Kamba, Bharat, and Albers (1993) who used user actions to personalize an online newspaper, Göker (1999) who used user context to help determine information need, and Oard and Kim (2001) who classified types of implicit feedback for recommender systems. We present a brief overview of the system with a complete description presented in Jansen and Pooch (2003).

4.1. Development

The system builds a model of user–system interactions using a technique similar to that utilized in some adaptive hypermedia systems. In adaptive hypermedia, a model of the user is represented by a set of pairs (c, v) where c is a *concept* and v is a *value* (De Bra & Calvi, 1998). We altered this approach for use in the IR area. In our system, an action–object pair (a, o) captures an instance of a user–system interaction, where a is an action taken by a searcher and o is the object that receives that action. A series of (a, o) pairs models a searcher's chain of interactions during the session. Using this series, the system can make determinations of the user's information need and provide appropriate assistance by associating certain actions with specific types of assistance.

Using the (a, o) pairs methodology has several advantages compared to other methods of gathering information from a user during a session. One advantage is time. Some automated systems build a model of the user's information need and then take action or provide suggestions for the user (Maes, 1994). The basis of this approach is usually relative long-term interaction between the user and the system. Unfortunately, most interactive sessions between searchers and Web IR systems are typically extremely short both in terms of the number of queries and user interests are extremely varied (He, Göker, & Harper, 2002; Jansen & Spink, 2003). For example, Jansen and Spink (2003) have shown that 26% Web search sessions are approximately 5 min or less. Relying solely on user–system interactions, the (a, o) pairs methodology can immediately begin forming a model of the user's information need.

A second advantage is that the (a, o) pairs methodology utilizes the full range of searcher–system dialogue. In the traditional view of IR interaction, the query is usually the only source of information from the user. Other techniques for gathering information (e.g., answering questions, completing profiles, judging relevance judgments) require the user to take additional actions beyond those typical of user interactions during an online search (e.g., Croft & Thompson, 1986; Gauch & Smith, 1993; Koenemann & Belkin, 1996). Using (a, o) pairs, the user's query is not the sole representation of the information need. The system obtains addition information is obtained with user actions, such as bookmarking, printing, emailing, etc. without requiring the user to take additional actions that divert attention from the search process.

The application currently monitors the searcher's interactions with the system, logging actions of *bookmark*, *copy*, *print*, *save*, *submit*, and *view results*. There are currently three objects that the system currently recognizes, which are *documents*, *passages from documents*, and *queries*.

When a session begins, the system monitors the user for one of the six actions, via a communication line using an application program interface (API), a technique commonly used to integrate software programs. When the system detects a valid action, it records the action and the specific object receiving the action. For example, if a searcher was viewing *this_document* and saved it, the system would record this as (*save this_document*). The system then offers appropriate search assistance to the user based on the particular action and the system's analysis of the object. In this example, the system would offer the searcher relevance feedback terms to the user from *this_document*. The more (a, o) pairs the system records and integrates these pairs, the more complex the model of the information need.

4.2. Assistance offered

We focused on five user–system interaction issues and corresponding system assistance, which we present in Table 1.

Our system design aim was the development a system with automated assistance representing the general state of the art in order to provide a realistic environment for the evaluation of searcher–system interaction. We believe the methods we use are generally effective for our research goals.

Table 1
Web searching issues and areas of automated assistance

Interaction issue	Discussion	Automated assistance
Structuring queries	Searchers usually have problems properly structuring queries, namely applying the rules of a particular system when using query operators (Spink et al., 2002). This is especially true with Boolean operators (e.g., AND, OR, NOT) and term modifiers (e.g. '+', '!') (Spink et al., 2002)	When the user submits a query, the application records this as a (submit query) pair, checks the query's structure based on the system's syntactic rules, correcting any mistakes
Spelling	Searchers routinely misspell terms in queries (Jansen et al., 2000; Yee, 1991), which usually drastically reduces the number of results retrieved. However, it is often difficult to detect these spelling errors because these queries frequently retrieve results from large document collections. Thus, the user may not realize the query contains a spelling mistake	A (submit query) pair alerts the application to check for spelling errors. The application separates the query into terms, checking each term using an online dictionary. The automated assistance application's current online dictionary is *ispell* (Gorin, 1971), although the application can access any online dictionary using the appropriate API The assistance is naturally based on the terms, phrase, and contextual data within the dictionary, including acronyms, multiple word senses, and proper names. In terms of multiple senses, we chose to display the noun sense based on prior researching showing that over 90% of searchers use only noun query terms (Jansen et al., 2000)
Query refinement	Searchers do not refine their query, even though there may be other terms that relate directly to their information need (Bruza et al., 2000). Studies show that searchers seldom modify their queries, or do so incrementally (Jansen et al., 2000), and then typically only one or two times	With a (submit query) pair and a thesaurus, the application analyzes each query term and suggests synonyms and the contextual definitions of the query terms. The system currently uses WordNet (Miller, 1998), but one can modify it to utilize any online thesaurus
Managing results	Searchers have trouble managing the number of results (Gauch and Smith, 1993). Generally, user queries are extremely broad, resulting in an unmanageable number of results. Few searchers view more than the first ten or twenty documents from the result list	Using the (submit query) pair and the number of results, the application provides suggestions to improve query. For this research, if the number of results is greater than twenty, the application provides suggestions to restrict the query (e.g., query with AND or PHRASE operators). If the number of results is less than twenty, the application provides advice on ways to broaden the query (e.g., query with OR operators or no Boolean operators) For our evaluation, we always wanted the application to provide assistance. Naturally, one could adjust the cut-offs to provide a range where no assistance is provided
Relevance feedback	Relevance feedback is an effective search tool (Harman, 1992); however, searchers seldom utilize it when offered. In this study, we extend previous research by automating the process using term relevance feedback (Mitra et al., 1998). When a (*bookmark document*), (*print document*), (*save document*), or (*copy passage*) pair occurs, the system implements a version of relevance feedback using terms from the document or passage object. The system provides suggested terms from the document that the user may want to add to the query	When a (bookmark document), (print document), (save document) or (copy passage) pair occurs, the application implements a version of relevance feedback using terms from the document or passage object. For example, if the user examines a document from the results list and performs one of the actions (i.e., bookmarking, printing, or saving), the application provides suggested terms from the document that the user may want to add to the query

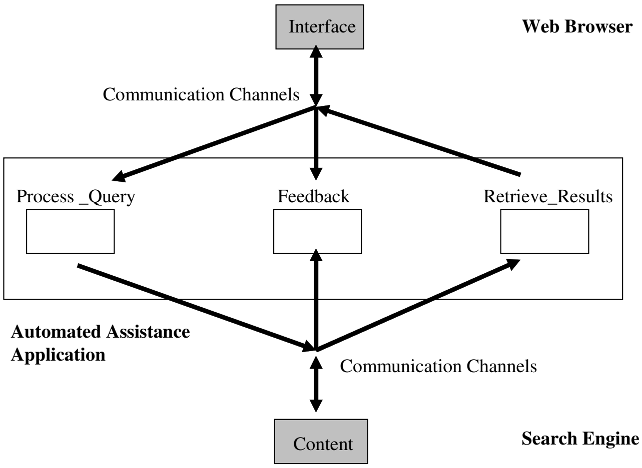

Fig. 1. Automated assistance interface modules.

4.3. Application modules

The automated assistance system has three major modules, as Fig. 1 displays.

The *Process Query* module records (a, o) pairs in a transaction log. When a session begins, the module checks to see if the transaction log is empty and if not, it empties it. While the user is logged in, the user log saves the user's actions and source to the file log, adds to the list the action query and the corresponding query for that particular action, and appends to the list if the user bookmarks, saves, prints, or copies a section of text and the source of that action. The *Process Query* module stores these actions and objects until the *Feedback* module accesses the transaction log and retrieves the actions and the objects.

The *Retrieve Results* module receives results from the search engine and passes the results to the *Feedback* module. One can modify the second of two modules for the particular search engine and graphical user interface. The *Retrieve Results* module accepts the results from the search engine. It then reformats the search results to a form suitable for the particular interface, which the application does not need if the interface already does this. Once the search engine returns the results, the *Feedback* module begins its analysis.

The *Feedback* module provides five types of information to the user. It offers (1) spelling suggestions for query terms, (2) terms from the query that have not appeared in the current results list nor in any former results list (i.e., a special case of out-of-vocabulary terms), (3) synonyms for query terms along with contextual definitions, and (4) suggestions to improve query structure. Once the user looks at a document from the results list, the application provides (5) relevance feedback on that document and returns a list of terms from the document that the user may want to add to the query. In order to provide minimal interruption of the searching process, the assistance is display with the search results.

4.4. Assistance interface

The system communicates with the user via a button appearing on the interface. If the user selects the automated assistance button, the assistance appears in a dialog box, along with a brief explanation of each type of assistance. Once the user views what the system has to offer, the button disappears until the system has more assistance to offer. The user can ignore the assistance with no impact on the normal operation of

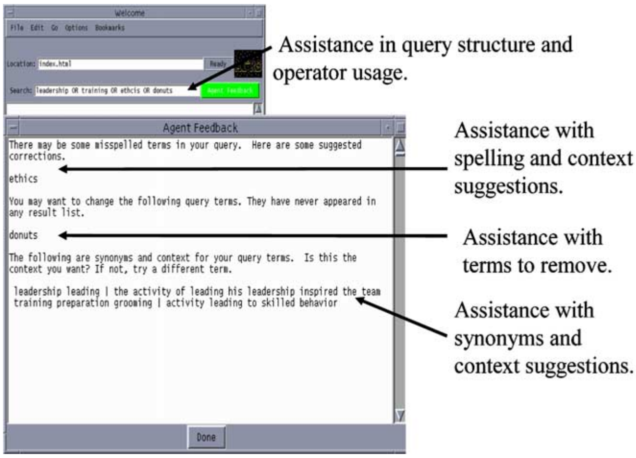

Fig. 2. Browser and automated assistance interface and dialog box.

the interface or IR system. For this experiment, the system offered aid whenever assistance was available. Fig. 2 shows the interface, dialog box, and text blocks with explanations.

5. Empirical evaluation

In this section, we describe the empirical evaluation we conducted to investigated our three research questions. We pilot tested the application using four subjects to ensure the effectiveness of the system and the evaluation design. We made changes in the system, primarily dealing with the presentation of the assistance.

5.1. Evaluation design

The IR system we utilized for the empirical study was MG (Witten, Moffat, & Bell, 1994) with a TCL/TK hypertext Web browser. We modified no MG code to integrate the automated assistance application with the MG system, other than establishing the API. We installed the IR system on a SPARC book 3 running Solaris 2.5. The subjects for the evaluation were 30 freshman college students (26 males and 4 females) in their second semester at a four-year undergraduate university. For this experiment, the system offered assistance whenever assistance was available.

The document collection we utilized for the evaluation was the Text REtrieval Conference (TREC) volumes number 4 and 5. The document collection is approximately 2 GB, containing approximately 550,000 documents. The randomly selected topics we used for this evaluation were *Topic 301: International Organized Crime* and *Topic 340: Land Mine Ban*. There were 474 relevant documents for *Topic 301: International Organized Crime* in the document collection. There were 81 relevant documents for *Topic 340: Land Mine Ban* for total of 555 (0.01%) relevant documents in the collection. We chose two topics to provide some searching variety in terms of focus and difficulty.

We provided each of the subjects a short statement instructing them to search on a given topic in order to prepare a report and the TREC definition of relevance judgment for TREC documents. The subjects had 5 min to find as many relevant documents as possible. We determined the length of the search session by measuring the length of time it would take to implement a "typical" Web search session, as outlined in Jansen, Spink, and Saracevic (2000). Web research has also noted a large percentage of Web sessions of 5 min or less. Jansen and Spink (2003) report that more than 26% of all Web sessions were 5 min or less. In a study of AltaVista searchers, Jansen, Spink, and Pederson (in press) note that more than 70% of searcher sessions are 5 min or less.

We notified each of the test subjects that the system contained an automatic feature to assist them with their searching. When the system had searching advice to offer, an assistance button would appear on the browser. The user could access the assistance by clicking the button, or they could ignore the offer of assistance with no detrimental effect on the system. We gave each subject one of the two search topics and the one paragraph explanation provide by the TREC collection. We directed the subject to search as when conducting online research.

We videotaped the users during the searching process, and a transaction log recorded user–system interactions. In order to add further robustness to the analysis, we instructed the subjects to think out loud during the searching process. Although transaction logs are excellent data collection tools, one should use transaction logs in conjunction with other data collection methods whenever possible (Borgman, Hirsh, & Hiller, 1996). All 30 subjects utilized the full 5 min for a total of 150 min of video for analysis. The transaction log recorded 503 interactions for the 30 subjects.

In analyzing the video, we coded the utterances using verbal protocol analysis (Ericsson & Simon, 1984), specifically the thinking-aloud protocol where the verbalization occurs in conjunction with a task. We used these coded utterances to clarify user interactions with the system recorded by the transaction log. After the search session, each searcher completed a subjective evaluation of the automated assistance. The combination of the protocol analysis, TL, and user evaluations provided a robust data source to conduct our analysis.

For the analysis presentation, we used 10 s for the temporal granularity, which permitted us to annotate each interaction within its own time hack. With these time periods, we could conduct cross sectional analysis of interactions that may have occurred ten seconds or less apart. A period larger than 10 s would include more than one user–system interaction within some periods. A period smaller than 10 s introduces unnecessary empty periods in the analysis. Other researchers have noted this very short temporal interaction period. Kelly and Belkin (2001) also reported a 10 s interaction period. Jansen and Spink (2003) noted that a sizable percentage of Web search engine users spend less than 30 s interacting with the search engine per session. If an interaction exceeded 10 s in duration, we counted it only once, in the period that the searcher initiated the interaction.

In analyzing the transaction logs and video, we explored the use of tools specifically designed for observational data analysis, specifically MacSHAPA (Sanderson, Scott, Johnston, & Mainzer, 1994); however, modern spreadsheet programs have much of the needed functionality for this type of empirical analysis. We coded each interaction between the user and the IR system. For our analysis, we performed the pattern recognition of the codes and follow-on calculations using Microsoft Excel and Visual Basic for Applications (VBA) scripts. We used Microsoft Bayesian Network (MSBNx) for the manipulation and creation of the Bayesian probability models.

6. Results

We first present data on the categories of user–system interaction and frequencies of occurrence. The results of a pattern analysis of the user–system interaction follows.

6.1. User–system interaction taxonomy

We identified the specific user actions on the system using the transaction log, supplemented by coded protocols from the video analysis. Jansen and Kroner (2003) reported initial results of this section. These

Table 2
Taxonomy of user–system interactions

	Taxonomy category	Number of occurrences	Percentage of all occurrences
1	*View Results Page* (*V*)	122	24.3
2	*View Document* (*D*)	105	20.9
3	*View Assistance* (*A*)	76	15.1
4	*Execute Query* (*E*)	68	13.5
5	*Implement Assistance* (*I*)	63	12.5
6	*Navigation* (*N*)	57	11.3
7	*Action Indicating Relevance* (*R*)	10	2.0
Total actions		501	99.60

Notes: (1) E—execute query; (2) V—view results; (3) A—view assistance; (4) I—implement assistance; (5) D—view document; (6) R—relevance action; (7) N—navigation.

interactions relate to searcher tasks during the search process. From this task analysis, we developed a taxonomy of user–system interactions, as shown in Table 2.

We identified seven categories, shown in Table 2, which account for 99.66% (501) of all user interactions with the system. There were 2 user actions that this taxonomy does not describe. There was one action to resize the browser and one action to access the normal help features of the system. Accounting for these two actions, there were 503 user interactions with the system representing 55.9% of the 899 possible observation periods. There were 396 periods (44.1%) where no interaction occurred. During these periods, the users were involved in some individual process, such as evaluating the assistance, processing information from a document, or determining the next interaction, etc., that did not involve physical interaction with the system.

We did not engage multiple reviewers in the classification exercise since the interactions are nearly binary (e.g., a particular interaction occurred or it did not occur). Descriptions of the taxonomy categories are:

- *View Results Page*: Interaction in which the user viewed or scrolled one or more pages from the results listing. If a results page was present and the user did not scroll, we counted this as a View Results Page.
- *View Particular Document*: Interaction in which the user viewed or scrolled a particular document in the results listings.
- *View Offered Assistance*: Interaction in which the user viewed the assistance offered by the application.
- *Execute Query*: Interaction where the user entered, modified, or submitted a query without visibly incorporating assistance from the system. This category includes submitting the original query, which was always the first interaction with system.
- *Implement Assistance*: Interaction where the user entered, modified, or submitted a query utilizing assistance offered by the application.
- *Navigation*: Interaction where the user activated a navigation button on the browser, such as Back or Home.
- *Action Indicating Relevance*: Interaction such as print, save, bookmark, or copy.

Addressing our first research question (e.g., How often do users seek and implement automated assistance in the search process?), Table 2 shows the users accepted the offer of assistance 76 times (15.1% of all user–system interactions) or 54% of the 141 times the system offered assistance. All 30 subjects viewed the assistance at least once.

The mean number of interactions with the application per subject was 3.67 with a standard deviation of 2.91 interactions. The searchers implemented the assistance 63 times (12.5% of all user–system interaction and 82% of the times viewed) for an average of 2.1 interactions per searcher with a standard deviation of 2.41. There were three users (10%) of the 30 that did not implement any assistance offered by the application.

The search process is composed of multiple interactions. A searcher enters a query, evaluates the results list or individual results. If the searcher does not locate documents or enough documents to satisfy the information need, the searcher may mentally reformulate the query, repeating the process (Robertson, 1977). This search process unfolds sequentially over time. Each interaction is discrete, but not necessarily independent. Therefore, in order to understand how searchers interact with automated assistance, we also analyzed these interactions within a process unfolding overtime.

We present a temporal view of when in the search process searchers interacted with the system in Fig. 3. As we see in Fig. 3, there are corresponding peaks and valleys between viewing the assistance and implementing the assistance. There is also a noticeable lag between the two categories. There was a reduction in the viewing of the automated assistance after the initial period representing a drop of approximately 9% between the first and second periods. The implementation of assistance held relatively steady until the last period, where it dropped off substantially. Table 3 displays the numbers of occurrences of *View Assistance* and *Implement Assistance* by minute of the search process.

6.2. Patterns of user–system interaction

In order to address the second (e.g., Where in the search process do users seek automated assistance?) and third (e.g., Where in the search process do users implement automated assistance?) research questions, we examined the transitions between interaction categories using exploratory sequential data analysis (Sanderson & Fisher, 1994). We examined these first at the single transition (i.e., users moving from one category to another category) and then at multiple transitions (i.e., users moving from one category to a second category to a third category, etc.).

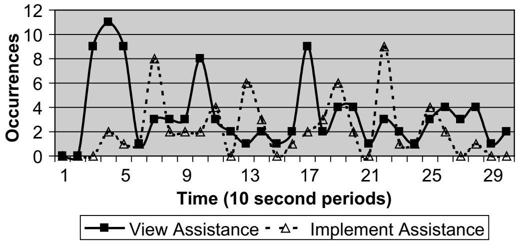

Fig. 3. Temporal view of interaction with automated assistance.

Table 3
Occurrences of *View Assistance* and *Implement Assistance* by period

Period	*View Assistance*		*Implement Assistance*	
	Occurrences	Percentage	Occurrences	Percentage
1	23	30.3	14	22.2
2	16	21.1	17	27.0
3	13	17.1	14	22.2
4	11	14.5	17	27.0
5	13	17.1	1	1.6
	76	100.0	63	100.0

Table 4
Examples of analysis of single and two transition patterns

Complete sequence of user interactions	Sub-patterns									
	1	2	3	4	5	6	7	8	9	10
EVADNEVEAIV										
Single transition	EV	VA	AD	DN	NE	EV	VE	EA	AI	IV
Two transition	EVA	VAD	ADN	DNE	NEV	EVE	VEA	EAI	AIV	

Notes: (1) E—execute query; (2) V—view results; (3) A—view assistance; (4) I—implement assistance; (5) D—view document; (6) R—relevance action; (7) N—navigation.

Table 4 illustrates an example of the transition analysis we conducted. Column 1 is the entire code sequence of one user's actions during the session (see Notes below Table 4). The other columns along the rows contain the set of all sub-patterns derived from that sequence at the single transition and two transaction levels of analysis.

In Table 4, the user sequence EVADNEVEAIV is composed of a sequence of 10 single-transition patterns (e.g., EV, VA, AD, etc.) and a sequence of 9 two-transition patterns (e.g., EVA, VAD, ADN, etc.). We conducted this analysis for all 30 users. We utilized an automated VBA script to identify the transition patterns within each sequence. To test the accuracy of the script, the single and two transitions patterns were manually generated for two randomly selected user sequences. The outcomes were identical. So, we feel comfortable that our script is accurate.

In addition to the one and two transition patterns, we also identified the three transition patterns; however the most common *View Assistance* pattern occurred less than 3% of the time and the most common *Implement Assistance* occurred approximately 4% of the time of all patterns. Given these low levels of occurrences, we did not examine these or lengthier transition patterns further. Prior research has also noted little gain from lengthy patterns (Marchionini, 1989).

6.3. Single transitions

Using our taxonomy, we identified the action immediately preceding a user requesting assistance and the action immediately preceding a user implementing the assistance. We coded each transaction from the first category to the second category taken by each user, assigning a code to each *category-to-category* pair. The codes we used are: (1) E—execute query; (2) V—view results; (3) A—view assistance; (4) I—implement assistance; (5) D—view document; (6) R—relevance action; and (7) N—navigation.

There were 37 unique patterns with 780 total pattern occurrences. From these occurrences, we isolated the patterns that terminated with a *View Assistance* (*A*) or *Implement Assistance* category (*I*). There were six unique *View Assistance* transition patterns with a total of 76 occurrences. For *Implement Assistance* (*I*), there were five unique transition patterns with 59 total occurrences. Table 5 displays the results of this analysis.

In Table 5, column 1 is all occurring initial categories. Each cell in column 2 is the number of unique patterns that begin with a particular category. Column 3 is the number of occurrences for those unique patterns. Column 4 is the number of pattern occurrences terminating with *View Assistance* (*A*), along with the percentage relative to the total occurrences with that Initial State. Column 5 is the number of pattern occurrences terminating with *Implement Assistance* (*I*), along with the percentage relative to the total occurrences with that Initial State. For example, row 1 shows that there were six unique single-transition patterns beginning with the category *View Document* (*D*). For these six unique patterns, there were 76 total occurrences, of which 11 (14.5%) were of the pattern *View Document–View Assistance* (*DA*) and none were of the pattern *View Document–Implement–Assistance* (*DI*).

Table 5
Single-transition patterns with terminating states of *View Assistance* and *Implement Assistance*

Initial categories	Unique patterns	Total occurrences	Number terminating with *View Assistance* (%)	Number terminating with *Implement Assistance* (%)
View Document (*D*)	6	76	11 (14.5%)	0 (0.0%)
Search (*E*)	5	67	4 (6.0%)	2 (3.0%)
View Assistance (*A*)	5	67	0 (0.0%)	45 (67.2%)
Implement Assistance (*I*)	4	57	4 (7.0%)	0 (0.0%)
Navigation (*N*)	6	40	10 (25.0%)	7 (17.5%)
View Results (*V*)	6	113	43 (38.1%)	4 (3.5%)
Relevance (*R*)	4	11	4 (36.4%)	1 (9.1%)
Total	36	431	76 (17.6%)	59 (8.1%)

Notes: (1) E—execute query; (2) V—view results; (3) A—view assistance; (4) I—implement assistance; (5) D—view document; (6) R—relevance action; (7) N—navigation.

The most common *View Assistance* pattern was the *View Results–View Assistance* (*VA*) pattern, which represented 38% of all *View Assistance* pattern occurrences. The *Relevance–View Assistance* (*RA*) and *Navigation–View Assistance* (*NA*) patterns also had relatively high occurrence rates of 36% and 25%, respectively; although numerical the *RA* only occurred four times.

For implementing assistance, the most commonly occurring pattern was *View Assistance–Implement* (*AI*) pattern, which occurred 67% of the time. The *Navigation–Implement* (*NI*) also occurred fairly frequently at 17%.

Figs. 4 and 5 display the Bayesian graphs of the single-transition patterns, terminating with *View Assistance* (*A*) and *Implement Assistance* (*I*) respectively.

6.4. Multiple transitions

We next analyzed dual transition patterns, where the searcher went from one category to a second category to a third category. There were 92 unique two-transition patterns and 445 two-transition occurrences. We were interested in the two-transition patterns that terminated with the user viewing the automated assistance and implementing the automated assistance. The results of our analysis are displayed in Tables 6 and 7.

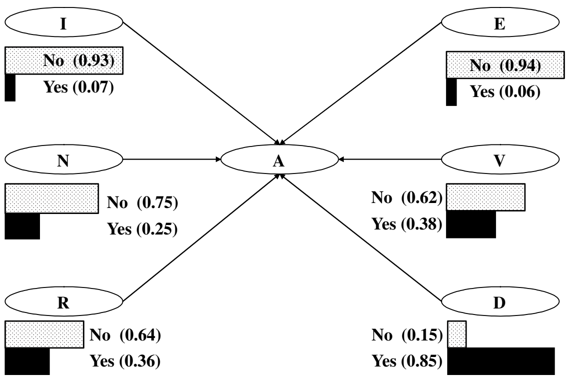

Fig. 4. Bayesian graph of single-transition patterns terminating with *View Assistance*. Notes: (1) E—execute query; (2) V—view results; (3) A—view assistance; (4) I—implement assistance; (5) D—view document; (6) R—relevance action; (7) N—navigation.

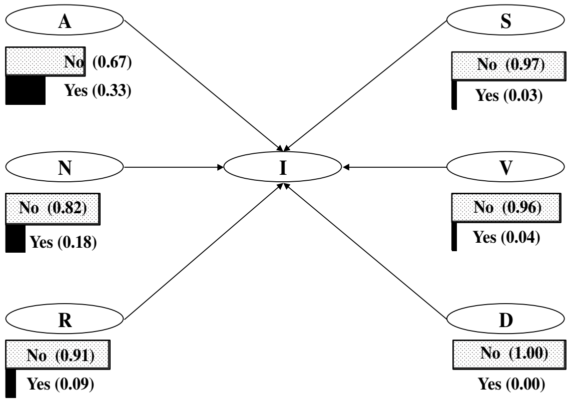

Fig. 5. Bayesian graph of single-transition patterns terminating with *Implement Assistance*. Notes: (1) E—execute query; (2) V—view results; (3) A—view assistance; (4) I—implement assistance; (5) D—view document; (6) R—relevance action; (7) N—navigation.

Table 6
Multiple transition patterns with terminating state of *View Assistance*

Two-transition patterns	Unique patterns	Total occurrences	Number terminating with *View Assistance* (%)
EV	5	55	29 (52.7%)
IV	6	48	14 (29.2%)
VD	9	50	9 (20.9%)
DN	5	22	7 (17.1%)
DR	3	9	4 (44.4%)
ND	6	21	2 (9.5%)
EI	1	2	2 (100.0%)
VN	4	5	2 (40.0%)
VS	2	11	2 (18.2%)
Total	41	223	71 (31.8%)

Notes: (1) E—execute query; (2) V—view results; (3) A—view assistance; (4) I—implement assistance; (5) D—view document; (6) R—relevance action; (7) N—navigation.

Table 7
Multiple patterns with terminating state of *Implement Assistance*

Two-transition patterns	Unique patterns	Total occurrences	Number terminating with *Implement Assistance* (%)
VA	3	34	24 (58.5%)
DA	3	9	7 (70.0%)
NA	4	9	6 (66.7%)
SA	1	4	4 (100.0%)
DN	6	41	3 (7.3%)
IA	1	3	3 (100.0%)
IV	6	48	2 (4.2%)
RN	3	5	2 (40.0%)
Total	27	153	51 (33.3%)

Notes: (1) E—execute query; (2) V—view results; (3) A—view assistance; (4) I—implement assistance; (5) D—view document; (6) R—relevance action; (7) N—navigation.

In Table 6, column 1 lists multiple transition patterns we identified where the next category was a *View Assistance* state (*A*). We include only the patterns that occurred more than once. Column 2 is the number of unique patterns that begin with the transition pattern in column 1. Column 3 is the total number of occurrences for these unique patterns. Column 4 is the number of pattern occurrences terminating with *View Assistance* and the percentage relative to the total occurrences for that single-transition pattern. For example, row 1 shows that for that there were five unique two-transition patterns beginning with *Execute Query–View Results* (*EV*). Of these five unique patterns, there were 55 total occurrences of which 29 (52.7%) were of the pattern *Execute Query–View Results–View Assistance* (*EVA*). Table 7 displays similar information for the *Implement Assistance* patterns.

For *View Assistance*, the most frequently occurring pattern was EVA representing 52% EV patterns. IVA was also a frequent occurrence at 29%, as was VDA at 20%. Other patterns, such as EIA had high percentages but occurred very infrequently. For *Implement Assistance*, the most frequently occurring pattern was VAI at 58% of all VA patterns. The remaining seven patterns terminating with *Implement Assistance* had very low occurrence rates, although some of the percentages of occurrences were very high (e.g., DVI with seven occurrences representing 70% of all DV patterns).

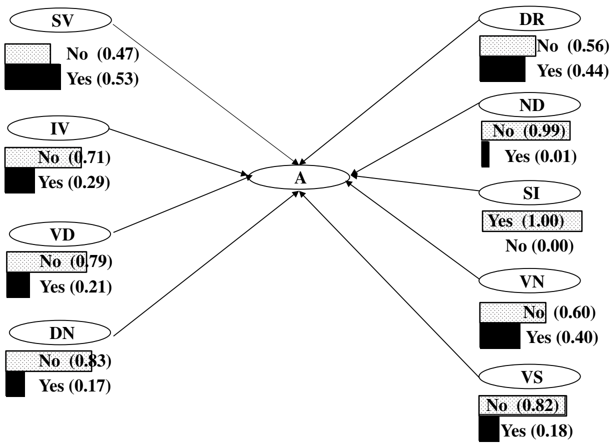

Fig. 6. Bayesian graph of two-transition patterns terminating with *View Assistance*. Notes: (1) E—execute query; (2) V—view results; (3) A—view assistance; (4) I—implement assistance; (5) D—view document; (6) R—relevance action; (7) N—navigation.

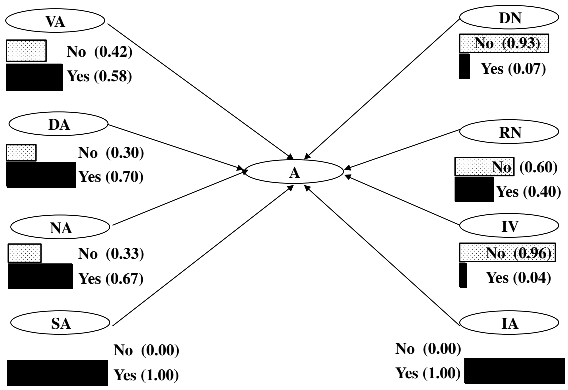

Fig. 7. Bayesian graph of two-transition patterns terminating with *Implement Assistance*. Notes: (1) E—execute query; (2) V—view results; (3) A—view assistance; (4) I—implement assistance; (5) D—view document; (6) R—relevance action; (7) N—navigation.

Figs. 6 and 7 display the Bayesian graph of the two-transition patterns terminating with *View Assistance* and *Implement Assistance*.

7. Discussion

For this research, we were interested in when in the search process searchers desire active involvement from the system, versus preferring to search without assistance. We investigated and analyzed users interacting with an automated assistance application. Returning to our research questions, we first investigated how often users seek and implement automated assistance. In this evaluation, the searchers interacted with the automated searching assistance 54% of the time (76 of the 141 offers). All 30 subjects viewed the assistance at least once. The implication of these results is that users will seek assistance if offered but also have a preference to work through the searching process on their own.

Searchers reject the offer of assistance nearly 50% of the time (46); however, searchers also implemented the assistance 82% (63) of the times they viewed the assistance. Therefore, although users may exhibit tendencies suggesting that they prefer to search without assistance, if the system provides assistance, users generally implement it. It also indicates that the users perceived the offered assistance as beneficial.

The temporal examination of the data shows that the searchers viewed and utilized the automated assistance more in the beginning of the search process, relative to later in the search process. After the initial surge, the level of interaction stabilizes. Users appear to settle into a comfortable level of interaction after the initial period of exploration. The initial surge in viewing may be due a novelty factor of the assistance. There was a drop in the final periods of searching, probably due to the effect of the time factor.

Investigating where in the search process users view assistance and when they implement it, there are certainly re-occurring patterns. Fifty-seven percent (57%) of all accesses to the automated assistance occurred after viewing the results listings. This would indicate that the users make some determination of the relevance of the results based solely on the meta-data presented in the results list. So, one might be able to develop a system using the meta-data from the result page and a searcher's reaction to that meta-data. Twenty-five percentage of the time, searchers viewed assistance occurred after a *Navigation* action, indicating that navigation actions may indicate indecision, frustration, or a transition point for searchers. Interestingly, searchers viewed assistance 36% of time after locating a relevant document. The initial success may lead to a desire to locate more relevant documents.

Users most commonly implemented the assistance immediately after viewing the feedback. However, perhaps surprisingly, users took some other action 22% of the time before implementing the assistance. This would again indicate that a sizable set of users, even after viewing assistance, prefer to attempt to proceed in the IS process on their own first. Even in these cases, however, they will eventually implement the assistance.

Over 53% of the time, searchers will *View Assistance* following an *Execute Query–View Results* (*EV*) pattern, without viewing any documents. Again, this is an indication that the searchers are reviewing the meta-data from the results pages, and finding nothing that appears relevant, they seek assistance from the system. This apparent behavior occurs again with the *Implement Assistance–View Results* (*IV*), followed by View *Assistance*. In this pattern (29% of all *View Assistance* patterns), searchers seek addition assistance from the system based solely on the results listing. Again, there was a high occurrence of seeking system assistance after locating relevant information, with *View Document–Relevance Action–View Assistance* (*DRA*) being 44% of all DR patterns.

Using these results, we are updating the automated assistance component to offer assistance at the most appropriate location in the search process. In the updated version, system logs the (a, o) pairs, generating searching advice as before, but the system offers the assistance to the user only after detecting a *Execute Query–View Results* (*EV*) and *Implement Assistance–View Results* (*IV*) pattern occurrences. Further enhancements are planned based on the results of this research, namely further refinement of the categories include sub-categories (e.g., scrolling of results listings, scrolling of documents, view next results page, or view previous results page). We are also investigating how to incorporate the meta-data from the results pages and searcher interactions to provide personalized and targeted assistance.

8. Conclusion

Despite an impressive array of IR systems that have been developed, many with advanced assistance features, few searchers utilize these mechanisms for improving search results (Jansen et al., 2000). Algorithmically, many of these features have been shown to be effective tools for enhancing the search process (e.g., Cronen-Townsend, Zhou, & Croft, 2002; Ruthven, 2003). Unfortunately, if searchers outside of the lab environment do not use them, these features are of little practical benefit. Researchers have developed automated assistance IR systems that take a more active role in the search process rather than passively waiting for the searcher to access assistance (Meadow, 1988). However, few of these systems have been widely adopted, especially on the Web (Sparck-Jones & Willett, 1997). One reason may be that these systems do not support the manner in which searchers actually prefer to search, namely they do not provide assistance when a searcher actually wants it. Conversely, these systems may intervene in the search process at times when the user prefers to search alone.

Much of the research in developing automated assistance systems has focused on automating strategy, tactics, and features. There has been much less research on when searchers actually desire this automated

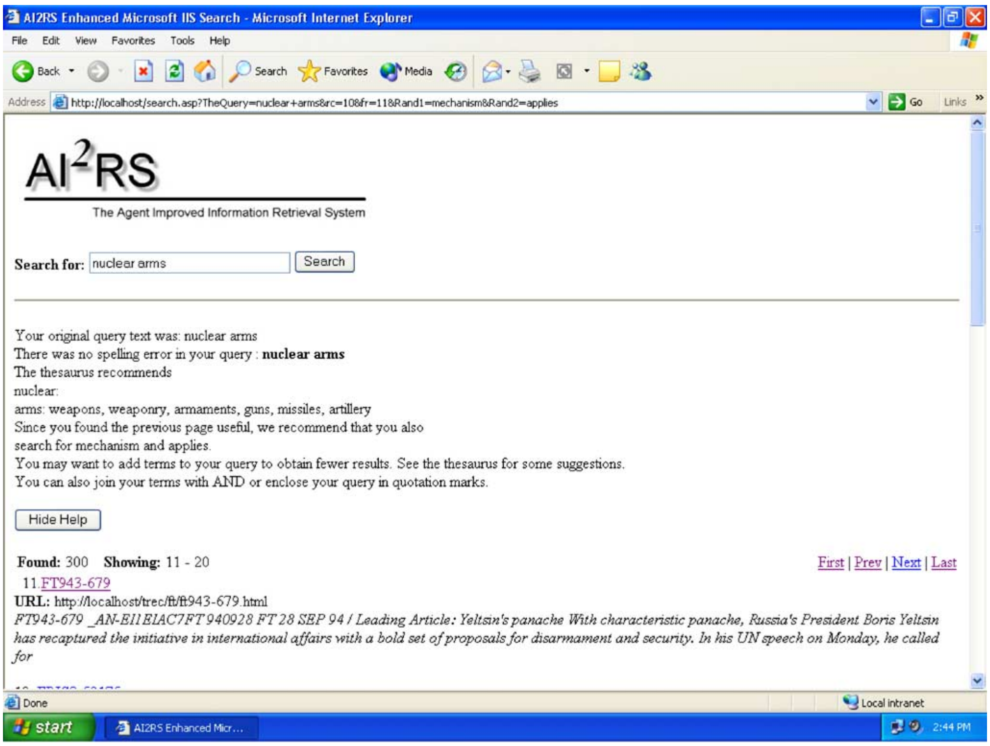

Fig. 8. Automated assistance integrated with Microsoft Explorer and Internet Information Service.

assistance. Other researchers have also raised this issue, calling for research into how these automated assistance system should interact with searchers (Bates, 1989, 1990; Belkin, 1988). It is this concern that this research addresses by examining how searchers interact with automated assistance while using IR systems.

Overall, the results of the research conducted so far shows potential to improve the searching process for the user. The results indicate that searchers have preferred points in the search process where they desire assistance and when they implement it. By detecting the patterns of user–system interaction, designers can tailor automated assistance systems to intervene at opportune temporal states when the probability is greatest that the user is willing to view the assistance or implement the assistance. With this knowledge, IR systems can more effectively assist users in finding the information they desire.

For future research, we are implementing the current version utilizing Microsoft Internet Information Service (IIS) as the backend IR system, running on an IBM-compatible platform using the Windows XP operating system and Microsoft Internet Explorer as the system interface. The system integration occurs via an API wrapper to the browser. This wrapper permits the monitoring of the user–system interactions and the presentation of the assistance to the user. Fig. 8 is a image of the current system, with automated assistance displayed.

References

Bates, M. J. (1979). Information search tactics. *Journal of the American Society for Information Science, 30*(4), 205–214.

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

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. (2000). Interaction in information searching and retrieval. *Journal of Documentation, 56*(4), 431–439.

Belew, R. K. (2001). *Finding out about: a cognitive perspective on search engine technology and the WWW*. New York: Cambridge University Press.

Belkin, N., Cool, C., Stein, A., & Theil, S. (1995). Cases, scripts, and information-seeking strategies: On the design of interactive information retrieval systems. *Expert Systems With Applications, 9*(3), 379–395.

Belkin, N., Oddy, R., & Brooks, H. (1982). Ask for information retrieval, Parts 1 & 2. *Journal of Documentation, 38*(2), 61–71, 145–164.

Belkin, N. J. (1988). On the nature and function of explanation in intelligent information retrieval. In *Proceedings of the 11th annual international ACM SIGIR conference on research and development in information retrieval, Grenoble, France, 13–15 June* (pp. 135–145).

Belkin, N. J., Cool, C., Kelly, D., Lin, S.-J., Park, S. Y., Perez-Carballo, J., & Sikora, C. (2001). Iterative exploration, design and evaluation of support for query reformation in interactive information retrieval. *Information Processing and Management, 37*(3), 403–434.

Borgman, C. L., Hirsh, S. G., & Hiller, J. (1996). Rethinking online monitoring methods for information retrieval systems: From search product to search process. *Journal of the American Society for Information Science, 47*(7), 568–583.

Brajnik, G., Guida, G., & Tasso, C. (1987). User modeling in intelligent information retrieval. *Information Processing and Management, 23*(4), 305–320.

Brajnik, G., Mizzaro, S., Tasso, C., & Venuti, F. (2002). Strategic help in user interfaces for information retrieval. *Journal of the American Society for Information Science and Technology, 53*(5), 343–358.

Bruza, P., McArthur, R., & Dennis, S. (2000). Interactive internet search: Keyword, directory and query reformulation mechanisms compared. In *Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval, Athens, Greece, 24–28 August 2000* (pp. 280–287).

Chen, H., & Dhar, V. (1991). Cognitive process as a basis for intelligent retrieval systems design. *Information Processing and Management, 27*(5), 405–432.

Chen, Z. X., Meng, X. N., Fowler, R. H., & Zhu, B. (2001). Real-time adaptive feature and document learning for Web search. *Journal of the American Society for Information Science, 52*(8), 655–665.

Choo, C., & Turnbell, D. (2000). Information seeking on the Web: An integrated model of browsing and searching. *First Monday* (Vol. 5). Available: http://firstmonday.org/issues/issue5_2/choo/index.html.

Cooper, W. S. (1968). Expected search length: A single measure of retrieval effectiveness based on weak ordering action of retrieval systems. *Journal of the American Society for Information Science, 19*(1), 30–41.

Croft, W. B., & Thompson, R. (1986). *I3: A new approach to the design of document retrieval systems* (COINS Technical Report 87-58). Amherst, MA: University of Massachusetts.

Cronen-Townsend, S., Zhou, Y., & Croft, W. B. (2002). Predicting query performance. In *Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval, Tampere, Finland, 11–15 August* (pp. 299–306).

De Bra, P., & Calvi, L. (1998). Aha: A generic adaptive hypermedia system. In *Proceedings of the ninth ACM conference on hypertext and hypermedia, Pittsburgh, PA, 20–24 June 1998* (pp. 283–284).

Ericsson, K. A., & Simon, H. A. (1984). *Protocol analysis: Verbal reports as data*. Cambridge, MA: MIT Press.

Gauch, S., & Smith, J. (1993). An expert system for automatic query reformulation. *Journal of the American Society for Information Science, 44*(3), 124–136.

Göker, A. (1999). Capturing information need by learning user context. In *Proceedings of sixteenth international joint conference in artificial intelligence: Learning about users workshop, Stockholm, Sweden* (pp. 21–27).

Gorin, R. E. (1971). Developer of Ispell. Retrieved 12 February, 2000, from the World Wide Web: http://fmg-www.cs.ucla.edu/geoff/ispell.html.

Harman, D. (1992). Relevance feedback revisited. In *Proceedings of the 15th annual international ACM SIGIR conference on research and development in information retrieval, Copenhagen, Denmark, 21–24 June* (pp. 1–10).

He, D., Göker, A., & Harper, D. J. (2002). Combining evidence for automatic Web session identification. *Information Processing and Management, 38*(5), 727–742.

Herlocker, J. L., Konstan, J. A., & Riedl, J. (2000). Explaining collaborative filtering recommendations. In *Proceedings of ACM 2000 conference on computer support collaborative work, Philadelphia, PA, 2–6 December* (pp. 241–250).

Hunter, R. (1991). Successes and failures of patrons searching the online category at a large academic library: A transaction log analysis. *Research Quarterly, 30*, 395–402.

Ingwersen, P. (1992). *Information retrieval interaction*. London: Taylor Graham.

Jansen, B. J., & Kroner, G. (2003). The impact of automated assistance on the information retrieval process. In *Proceedings of the ACM CHI 2003 conference on human factors in computing systems, Fort Lauderdale, FL, 5–10 April* (pp. 1004–1006).

Jansen, B. J., & Pooch, U. (2003). Assisting the searcher: Utilizing software agents for Web search systems. *Internet Research—Electronic Networking Applications and Policy, 14*(1), 19–33.

Jansen, B. J., & Spink, A. (2003). An analysis of Web information seeking and use: Documents retrieved versus documents viewed. In *Proceedings of the 4th international conference on Internet computing, Las Vegas, NV, 23–26 June* (pp. 65–69).

Jansen, B. J., Spink, A., & Pederson, J. (in press). Trend analysis of Alta Vista Web searching. *Journal of the American Society for Information Science and Technology*.

Jansen, B. J., Spink, A., & Saracevic, T. (1998). Failure analysis in query construction: Data and analysis from a large sample of Web aueries. In *Proceedings of the 3rd ACM conference on digital libraries, Pittsburgh, PA* (pp. 289–290).

Jansen, B. J., Spink, A., & Saracevic, T. (2000). Real life, real users, and real needs: A study and analysis of user queries on the Web. *Information Processing and Management, 36*(2), 207–227.

Johnson, H., & Johnson, P.E. (1993). Explanation facilities and interactive systems. In *Proceedings of the IUI'93 international workshop on intelligent user interfaces, Orlando, Florida, USA, 4–7 April* (pp. 159–166).

Kahle, B. (1999). *Alexa Internet Press Releases* [Web site]. Alexa Internet. Retrieved 12 February, 2000, from the World Wide Web: http://www.alexa.com/company/recent_articles.html.

Kamba, T., Bharat, K., & Albers, M. (1993). The Krakatoa chronicle: An interactive personalized newspaper on the Web. *The World Wide Web Journal, 1.* Available: http://www.w3j.com/1/Contents.html#19.

Kelly, D., & Belkin, N. J. (2001). Reading time, scrolling and interaction: Exploring implicit sources of user preferences for relevance feedback. In *Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval, New Orleans, LA, United States* (pp. 408–409).

Koenemann, J., & Belkin, N. (1996). A case for interaction: A study of interactive information retrieval behavior and effectiveness. In *Proceedings of the CHI '96 conference on human factors in computing systems, Vancouver, British Columbia, Canada, 13–18 April* (pp. 205–212).

Kuhlthau, C., Spink, A., & Cool, C. (1992). Exploration into stages in the information search process in online information retrieval: Communication between users and intermediaries. In *Proceedings of the 55th annual meeting of the American Society for Information Science, Medford, NJ* (pp. 67–71).

Lawrence, S., Giles, C. L., & Bollacker, K. (1999). Digital libraries and autonomous citation indexing. *IEEE Computer, 32*(6), 67–71.

Lieberman, H. (1995). Letizia: An agent that assists Web browsing. In *Proceedings of the 14th international joint conference on artificial intelligence, Montreal, Quebec, Canada, August* (pp. 924–929).

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

Marchionini, G. (1989). Information-seeking strategies of novices using a full-text electronic encyclopedia. *Journal of the American Society for Information Science, 40*(1), 54–66.

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

Meadow, C. T. (1988). Oakdec, a program for studying the effects on users of a procedural expert system for database searching. *Information Processing and Management, 23*(4), 449–457.

Meadow, C. T., Hewett, T. T., & Aversa, E. (1982a). A computer intermediary for interactive database searching I: Design. *Journal of the American Society for Information Science, 33*, 325–332.

Meadow, C. T., Hewett, T. T., & Aversa, E. (1982b). A computer intermediary for interactive database searching II: Evaluation. *Journal of the American Society for Information Science, 33*, 357–364.

Middleton, S. E., Roure, D. C. D., & Shadbolt, N. R. (2001). Capturing knowledge of user preferences: Ontologies in recommender systems. In *Proceedings of first international conference on knowledge capture, Victoria, British Columbia, CA* (pp. 100–107).

Miller, G. (1998). *Wordnet: An electronic lexical database*. Cambridge, MA: MIT Press.

Mitra, M., Singhal, A., & Buckley, C. (1998). Improving automatic query expansion. In *Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval, Melbourne, Australia, 24–28 August 1998* (pp. 206–214).

Mizzaro, S. (1996). *Technical Report Udmi/18/96/Rr: Intelligent interfaces for information retrieval: A review*. Technical Report, Department of Mathematics and Computer Science, University of Udine.

Oakes, M. P., & Taylor, M. J. (1998). Automated assistance in the formulation of search statements for bibliographic databases. *Information Processing and Management, 34*(6), 645–668.

Oard, D., & Kim, J. (2001). Modeling information content using observable behavior. In *Proceedings of the 64th annual meeting of the American Society for Information Science and Technology, Washington, DC, USA, November* (pp. 38–45).

Oddy, R. N., & Balakrishnan, B. (1991). Pthomas: An adaptive information retrieval system on the connection machine. *Information Processing and Management, 27*(4), 317–335.

Peters, T. (1993). The history and development of transaction log analysis. *Library Hi Tech, 42*(11), 41–66.

Robertson, S. (1977). Theories and models in information retrieval. *Journal of Documentation, 33*, 126–148.

Ruthven, I. (2003). Re-examining the potential effectiveness of interactive query expansion. In *Proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval, Toronto, CA* (pp. 213–220).

Salton, G., & McGill, M. J. (1983). *Introduction to modern information retrieval*. New York: McGraw-Hill.

Sanderson, P. M., & Fisher, C. (1994). Exploratory sequential data analysis: Foundations. *Human–Computer Interaction, 9*(1), 251–317.

Sanderson, P. M., Scott, J., Johnston, T., & Mainzer, J. (1994). Macshapa and the enterprise of exploratory sequential data analysis (Esda). *International Journal of Human–Computer Studies, 41*(1), 633–681.

Saracevic, T. (1975). Relevance: A review of and a framework for the thinking on the notion in information science. *Journal of the American Society of Information Science, 26*(6), 321–343.

Saracevic, T. (1996). Modeling interaction in information retrieval (IR): A review and proposal. In *Proceedings of the 59th American Society for Information Science annual meeting, Baltimore, MD, 19–24 October* (pp. 3–9).

Saracevic, T. (1997). Extension and application of the stratified model of information retrieval interaction. In *Proceedings of the annual meeting of the American Society for Information Science, Washington, DC, October* (pp. 313–327).

Siegfried, S., Bates, M., & Wilde, D. (1993). A profile of end-user searching behavior by humanities scholars: The Getty Online Searching Project Report No. 2. *Journal of the American Society for Information Science, 44*(5), 273–291.

Sparck-Jones, K., & Willett, P. (1997). Overall introduction. In K. Sparck-Jones & P. Willett (Eds.), *Readings in information retrieval* (pp. 1–7). San Francisco: Morgan Kaufman.

Spink, A., Jansen, B. J., Wolfram, D., & Saracevic, T. (2002). From E-Sex to E-Commerce: Web search changes. *IEEE Computer, 35*(3), 107–111.

Witten, I., Moffat, A., & Bell, T. C. (1994). *Managing Gigabytes: Compressing and indexing documents and images*. New York, NY: Van Nostrand Reinhold.

Yee, M. (1991). System design and cataloging meet the user: User interfaces to online public access catalogs. *Journal of the American Society for Information Science, 42*(2), 78–98.