Evaluation of an adaptive search suggestion system

Sascha Kriewel and Norbert Fuhr

University of Duisburg–Essen

Abstract. This paper describes an adaptive search suggestion system based on case–based reasoning techniques, and details an evaluation of its usefulness in helping users employ better search strategies. A user experiment with 24 participants was conducted using a between–subjects design. One group received search suggestions for the first two out of three tasks, while the other didn't. Results indicate a correlation between search success, expressed as number of relevant documents saved, and use of suggestions. In addition, users who received suggestions used significantly more of the advanced tools and options of the search system — even after suggestions were switched off during a later task.

1   Introduction

While nowadays information search technology is pervasive, and it is used by the larger public instead of just search professionals or search intermediaries, the effective use of these information retrieval (IR) technologies by end–users remains a challenge [10, 26]. Many modern information retrieval systems provide functionalities beyond those of low–level search actions, but rarely do they support users in finding the right tactic for the right situation [8], i.e. help searchers choose the most appropriate or useful action in a specific situation during a long search session to implement an overall search strategy.

Novices often try to express several concepts at once, and either over– or under–specify their requests [23]. If they reformulate their query, they mostly use parallel movements or add terms to their original query [24]. They have difficulties finding a set of search terms that captures their information need and brings a useful result set [11]. Rarely do they employ anything resembling sophisticated search strategies [9], and often they even use counter–productive moves. Search experts on the other hand have access to effective information finding strategies, that enable them to interact with the system to achieve good retrieval results, depending on the history of their interaction so far, and the result currently displayed [11]. Furthermore, users rarely know how and when to use advanced search features or tools to the best effect [21].

This is especially problematic when users work on complex search problems, such as researching viewpoints to support an argumentation. Users engage in

long interactive search sessions and don't expect to find all their answers as a result of a single query. And while there has been extensive work in supporting users in executing specific moves, tactics or stratagems (to use the terminology introduced by Bates [4]), such sessions are rarely supported by common systems.

This is where automatic search assistance can be of help. In this paper a system for situational search suggestions is described, and an evaluation of its usefulness is presented. Three main questions were examined: do users who receive suggestions search more successfully over the course of the complete session, do they use more advanced options, and can they learn from these suggestions.

2   Providing strategic help

Search strategies can be seen as plans for completing search tasks, encompassing many tactics and stratagems used in the process of an information search [4]. To use such a strategy, users need to select the appropriate stratagems or tactics supported by the system. However, searching is an opportunistic process and search goals can shift throughout the task [27]. It is often more fruitful for a searcher to follow promising opportunities that arise from previous results, instead of sticking to a straight path towards the perfect result set. Järvelin [17] showed that search sessions using individually ineffective queries may actually be more effective than long single queries, depending on how costs and benefits of interactive IR sessions are considered in the evaluation of search success.

Therefore, users need to recognize these strategic opportunities during a session and use the strategic options available to maneuver out of dead–ends. Unfortunately, users rarely exploit the advanced capabilities and features of modern search systems, even if these would improve their search. They might not be aware of their existence, might not understand them, or don't know in which situation they could be effectively employed. Search systems providing strategic assistance can improve search effectiveness by suggesting the use of these advanced features or options automatically [15].

Drabenstott [10] studied the use of domain expert strategies by nondomain experts, and concluded that there is a "need for system features that scaffold nondomain experts from their usual strategies to the strategies characteristic of domain experts." Wildemuth [26] pointed out that current IR systems offer little support to help searchers formulate or reformulate effective search strategies.

In [8] Brajnik et al. describe a strategic help system based on collaborative coaching, which tries to assist users by providing them with suggestions and hints during so–called critical or enhanceable situations. The system uses a hand-crafted knowledge base of 94 production rules to provide suggestions based on the tactics and stratagems proposed by Bates [3, 4]. The strategic help module was integrated into FIRE, a user-interface to a Boolean IR system. Only six people participated in the user evaluation, but the results showed promise for the usefulness of strategic suggestions.

Bhavnani et al. [7] introduce the idea of Strategy Hubs, which provide domain specific search procedures for web search. While domain specific strategies may

not be transferable to a general search system, they also noted the importance of procedural search knowledge in addition to system and domain knowledge, and the need to make such knowledge explicit within the search system.

Jansen and McNeese evaluated the effectiveness of automated search assistance in a within–subject study with 40 participants (although only 30 of them actually used the search assistance provided), and found that automated assistance can improve the performance of the searcher depending on how performance is measured. However, it was also noted, that about half of the participants actually did not perform better, and that automated assistance should to be targeted and possibly personalized to achieve the best results [16].

3   The suggestion tool

We developed a suggestion tool using case–based reasoning techniques for the DAFFODIL system to support users with useful strategic search advice. DAFFODIL [12] is a digital library search system offering a rich set of tools that experienced users can employ in their search. However, previous user experiments showed that inexperienced users have problems utilizing these tools [19].

For the experiment described in this paper, a stripped down version of DAFFODIL was used (shown in fig. 1). It contains a search tool for querying digital libraries using a fielded search form (the available fields are "title", "author", "year", and "free–text"). The search supports Boolean queries and phrases. Results from different sources are combined and ranked. A search history shows all previous queries of the user, and allows for re–use and editing of queries. Relevant documents (as well as queries, terms, or authors) can be stored in a clipboard. A detail viewer shows a short summary of a result document, including abstract and a full–text link where available. Details can be shown either for documents in the result list or for documents stored in the clipboard. The document details are interactive, and links for authors, keywords, or terms can be used to run new queries, modify the current one, or call other tools. In addition, a number of support tools were provided:

- extraction of popular terms and authors from the result list
- extraction of terms and authors from relevant documents in the clipboard
- display of related terms for the current query
- a thesaurus with synonyms, super– or subordinate concepts, and definitions
- an author network showing co–author relationships for a selected author
- a classification browser for the search domain

The suggestion system itself has been described previously [20], and consists of three main components. The *observing agent* triggers after every search to collect information about the current query, the returned results, and the search history of the user. From these, a case is constructed that is used by the *reasoning agent* to match against previously stored cases. Those suggestions that were found to be appropriate solutions for similar cases are collected and ranked.
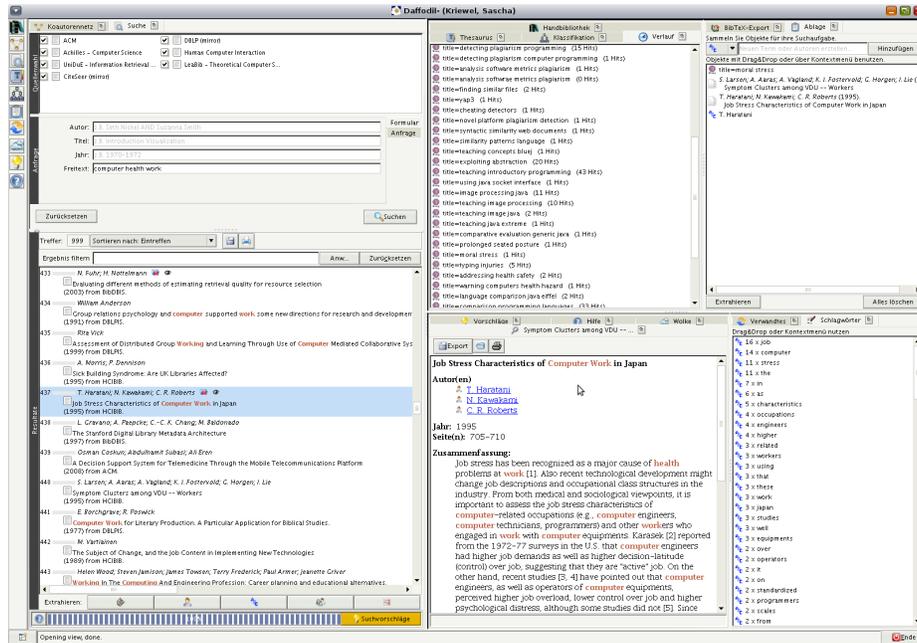
Fig. 1. The DAFFODIL desktop: search tool (left), search history (top center), clipboard (top right), detail view (bottom center), and Related Terms (bottom right)

The *suggestion tool* adapts these weighted suggestions for the current situation and offers them to the user. Users can either ignore the offer, or call up a ranked list of suggestions with a graphical bar for each list entry indicating the estimated likelihood of the suggestion being useful in the current situation. The suggestions come with a short explanation of their purpose and usage scenario, and most allow for direct execution by simple double–click.

3.1 Suggestions

The suggestions provided by the suggestion tool were compiled from the information science literature, e.g. from [3, 4, 14, 13], and supplemented by suggestions specific to problems discovered during user experiments with the DAFFODIL system [19, 25]. 22 suggestions have been included, which can be grouped as

- terminological suggestions (use of spelling variants, related terms, synonyms, subordinate or superordinate concepts)
- suggestions regarding use of operators and search fields (use of disjunction instead of conjunction, use of phrase operators, use of title or free–text field, restricting by years or authors)
- strategic suggestions (creating conceptual facets, pearl growing using a document previously saved as relevant, avoiding overspecification, author search)
- suggestions for advanced tools of the DAFFODIL system (computation of co–author network, browsing the classification, filters to extract result list, extracting terms from relevant documents or result sets)

3.2 Finding suggestions

Techniques from case–based reasoning are used to find and rank the most appropriate suggestions for the user's search situation. Belkin and colleagues [5] previously proposed a case–based reasoning approach for designing an interactive IR system. They applied this in the MERIT system to provide a script–based interaction with the system, based on cases that are associated with specific regions of the ISS space defined in [6].

In the scenario described in this paper cases are previous search situations of other users, and their solutions are specific suggestions that were found useful by those users. Users can judge suggestions as useful or not useful for their current situation, thereby adding to the case base. The cases are composed of a number of aspects which are used for comparing cases and to compute similarities. These aspects all fall within one of three categories: numeric values (e.g., number of results), sets of terms (e.g., the query terms), or vectors of term weights (e.g., popular terms extracted from a result set with number of occurrences per term).

After every search action of the user, a new situation is compiled and the reasoning component retrieves the most similar cases for each available solution from the database, with a cut–off value of 50% — i.e. no cases with a similarity score of less than 0.5 compared to the current situation are used.

The similarity $sim_T(a, b)$ of two situations a and b is computed as the *weighted mean* (using weights w_k) of the individual similarity scores sim_k of the various aspects of a situation.

For selecting and ranking the suggestions that will be presented to the user as search advice, both positive and negative cases are used. Positive cases are cases similar to the current situation for which a user has rated a specific suggestion as useful. Correspondingly, negative cases are those where a user has rated a specific suggestion as not useful. The use of negative cases is common for applications of case–based reasoning in medicine, where positive and negative indicators for treatments or diagnoses. Ontañón and Plaza [22] propose a justification based method for deriving a confidence score. A similar method has been used here.

$$p_v := s_v * \frac{\sum_{b_i \in F^+(v)} sim_T(a, b_i)}{\sum_{b_i \in F^+(v)} sim_T(a, b_i) + \sum_{b_i \in F^-(v)} sim_T(a, b_i)} \tag{1}$$

Be s_v the similarity of the most similar case for a suggestion v, $F^+(v)$ the set of similarities of all positive cases for the suggestion v and $F^-(v)$ the set of similarities for all negative cases for v. The total weight for this suggestion is then computed as shown in eqn. 1. The suggestions are ranked by decreasing weights p_v.

3.3 Pilot evaluation

A small scale pilot evaluation was conducted in 2007, during which 12 users worked with the suggestion tool on a simulated, complex work task. The evaluation pointed out a number of possible improvements, but overall user reception

of the suggestions was positive. The users found the automated, non–intrusive advice to be helpful, and implemented suggested tactics with success to further their search task [20]. Several problems that were found during the evaluation were fixed and the list of available suggestions has been extended. The system has been in use since then and user ratings have expanded the case base.

4   Experiment

A user experiment was conducted to evaluate if an adaptive suggestion tool using case-based reasoning techniques can help users search better and learn how to use the advanced capabilities of a complex search system.

4.1   Research questions

Three major research questions drove the experiment: (1) Do the suggestions lead to a more successful search? (2) Do the suggestions help users to use advanced search techniques and more of the advanced tools of the system? (3) Do users utilise these techniques and tools on their own, after having encountered them during their use of the suggestion tool?

For the first question, the number of relevant documents saved by the users was considered. For the second and third question, the search logs were analysed and the use of different DAFFODIL tools and search fields was counted.

4.2   Participants

For the study, 24 volunteers were recruited from among students of computer science, communication science, and related subjects at the University of Duisburg–Essen. The students, 20 male and 4 female, received extra course credit for their participation. The age of the participants ranged from 22 to 48, with an average age of 27.25 years (standard deviation 5.41, mode 24). Among other prestudy questions, the participants were asked to report their previous search experience. On a five-point Likert scale, two rated themselves as inexperienced (2), six as moderately experienced (3), 15 as experienced (4), and one as expert (5).

Since studies [1] have shown that self-evaluation of search skill can be a poor indicator for search performance (at least for some tasks), the participants were also asked to describe their use of search systems in years of experience and frequency of use. The average experience in years was 4.75 with a standard deviation of 2.45. Not all participants provided a useful estimation of their frequency of search engine use, but 6 reported less then daily use. Of the daily users that provided a more exact estimate, the frequency ranged between 1-35 distinct uses of search engines per day (average 11 times per day, standard deviation 11.84).

When asked about search systems with which they are familiar, 23 of the participants named Google, 4 named Yahoo, and 2 MSN Search or Bing. Three of the students reported that they used desktop search systems like Google Desktop or Beagle for Linux. Three students used digital library search systems.

4.3 Setup

For the study two systems were used, both based on the DAFFODIL system described in section 3. The systems were identical, except for the inclusion of the suggestion tool in one of them. Both contained a search tool configured to search six digital libraries from the computer science domain.

All participants were asked to work on three complex search tasks for 20 minutes each. During this time they had to find and save as many relevant documents as possible. Half of the participants (the assisted group) used the system with suggestions for the first two tasks, and the normal system for the last task. The other half (the unassisted group) used the normal system throughout the experiment. The participants were not told the purpose of the study beforehand, and were unaware of the fact that some received search assistance and some did not. They were assigned randomly to one of the two groups. No significant difference in average search experience (4.83 vs. 4.67 years, $p = 0.872$) or self-evaluation (both groups had a mode and median of 4, which corresponds to "experienced" on the Likert scale used) was found between the two groups.

The tasks were selected with the document collection in mind and with the goal of providing tasks that are similar to real work tasks. One of the tasks (1) was modified from TREC-6 adhoc topic no. 350, while the other two tasks were chosen to be similar in difficulty (see table 1). The order of the tasks was rotated.

Participants from both groups received an identical overview of the DAFFODIL search system. The assisted group was also shown the button to request suggestions, and given an explanation on how to execute a suggestion. Rating of suggestions was disabled during the experiment. Before each task the participants were given a textual description of the search scenario. If necessary the task was clarified. During the timed search only technical questions were answered, no search or terminological advice was given.

4.4 Evaluation

The DAFFODIL logging framework [18] was used to capture all user activities during the search. In addition, the searchers were encouraged to think aloud during their tasks and a note taker was present during the entire experiment. These notes were used to clarify ambiguous log entries.

From the logs two subsets were extracted for each user and task: (a) a history of all queries issued with the search tool, (b) a history of all uses of DAFFODIL's tools. The query log was hand–coded using the codes in table 2. The tool log was used to compute average use of tools (table 3). Since each request for assistance resulted in a list of suggestions from which more than one could be used, the number of executed suggestions is slightly higher than the number of requests.

Documents saved by the participants were pooled and two relevance assessors judged them for relevance. These blind judgements were used to calculate the number of relevant documents saved by each searcher for each task.

Task	Description
Health	Search for articles to answer if and how daily work with computer terminals can be hazardous to an individual's physical or mental health. Relevant documents describe physical problems (such as carpel tunnel, fatigue, eye strain) or mental problems (such as stress) caused or aggravated by regular work with computer terminals.
Plagiarism	Search for articles that describe or evaluate methods that can be used to detect or fight plagiarism of software source code. Relevant are descriptions of methods specific to detecting software plagiarism, application of existing methods to the domain of source code plagiarism, or comparisons of methods. Not relevant are documents dealing with plagiarism detection for text documents.
Java	Search for articles to argue for or against Java as a teaching language in the first year of computer science education. Relevant are articles dealing with Java as an educational tool for teaching basic computer science concepts, comparisons with other languages for this purpose, or general articles describing criteria for a good teaching language. Not relevant are articles dealing with learning or teaching Java programming.

Table 1. Task descriptions used in user experiment

Code	Move	avg. (stdv.) with sugg.	avg. (stdv.) without sugg.	Welch's t test t	df	p
T	add/drop term	7.50 (5.95)	12.00 (7.62)	-1.61	20.78	0.122
S	replace term	11.08 (4.89)	16.92 (8.54)	-2.05	17.51	0.055
R	replace whole query	3.33 (2.57)	6.00 (4.05)	-1.93	18.64	0.069
O	repeat a query	0.83 (1.11)	1.08 (1.31)	-0.50	21.44	0.620
F	use add./diff. fields	7.08 (5.04)	0.92 (1.78)	3.99	13.71	*0.001*
B	add/drop boolean op.	6.25 (6.16)	6.66 (6.92)	-0.16	21.71	0.877
C	spelling correction	2.25 (1.82)	2.42 (2.81)	-0.17	18.82	0.865
total	total use	38.25 (9.35)	46.58 (15.31)	-1.61	18.21	0.124
F_3	use of fields (task 3)	1.75 (1.42)	0.25 (0.62)	3.35	15.05	*0.004*

Table 2. Number of moves made to change the query (suggestions for tasks 1 and 2).

4.5 Results

Research question 1. The first question to be examined was if searchers who received assistance from the search suggestion tool were more successful in their searches. As can be seen in table 3, users in the assisted group issued fewer queries then those in the group without (46.5 vs. 51.33), but not significantly (using an unpaired Welch's two sample t test). They viewed more document details (79.17 vs. 51.67), but again the difference is not significant using Welch's t test. The difference in saved documents (38.17 vs. 23.25) is significant with $p = 0.019$, but it is not clear if this would be a good indicator for search success.

Therefore, relevance judgements of independent assessors were considered. The number of relevant documents saved on average for the three tasks by the two groups can be seen in table 4. Users within assisted group saved significantly more relevant documents for all tasks combined ($p = 0.002$) as well as for

Action	avg. (stdv.) with sugg.	avg. (stdv.) without sugg.	Welch's t test		
			t	df	p
Basic actions / Suggestions					
Run query	46.50 (9.76)	51.33 (19.89)	-0.82	17.01	0.423
View details	79.17 (45.47)	51.67 (28.50)	1.78	18.49	0.092
Save document	38.17 (14.86)	23.25 (14.18)	2.52	21.95	*0.019*
Request suggestions	8.33 (3.11)	— (—)	—	—	—
Execute suggestion	9.50 (4.01)	— (—)	—	—	—
Advanced actions	10.33 (6.53)	3.75 (3.89)	3.02	18.00	*0.007*
Extract terms	4.91 (4.60)	0.50 (1.73)	2.99	12.58	*0.011*
Use thesaurus	1.17 (1.75)	0.92 (1.56)	0.37	21.73	0.716
Change query from other tool	4.67 (2.87)	2.33 (2.31)	2.19	21.03	*0.039*
all adv. actions (only task 1)	3.42 (2.39)	1.16 (1.34)	2.84	17.26	*0.011*
all adv. actions (only task 2)	4.25 (1.24)	1.33 (1.44)	3.24	16.52	*0.005*
all adv. actions (only task 3)	2.67 (1.78)	1.25 (1.29)	2.48	21.52	*0.021*

Table 3. User actions taken during tasks (suggestions for tasks 1 and 2).

Task	avg. (stdv.) with sugg.	avg. (stdv.) without sugg.	Welch's t test		
			t	df	p
Health	8.25 (5.64)	3.00 (2.95)	2.86	16.61	*0.011*
Plagiarism	14.92 (5.30)	8.25 (7.62)	2.49	19.63	*0.022*
Java	2.42 (1.78)	1.42 (1.62)	1.44	21.81	0.165
Total	25.58 (9.61)	12.67 (8.95)	3.41	21.89	*0.002*

Table 4. Average number of relevant documents saved.

Health and Plagiarism tasks ($p = 0.011$ and $p = 0.022$). There was no significant difference in performance for the Java search task.

It seems that, although the search suggestions did help users to search more successfully, the assistance doesn't help for all tasks.

Research question 2. The second question concerned the actions that users perform during their search. The hypothesis was that users in the assisted group would employ advanced tools and search options of DAFFODIL significantly more often than users in the unassisted group.

Evaluation of the search logs showed no statistically significant differences between both groups with regards to moves to change the query by adding or removing terms or boolean operators, correcting the spelling of a term, re–using a previous query, replacing one or more terms, or replacing the entire query (see table 2). However, the differences for the later two moves were only barely not significant: users in the *unassisted* group replaced query terms on average 16.92 times per search versus 11.08 times ($p = 0.055$) and replaced the whole query 1.31 per search versus 0.83 times ($p = 0.062$).

On the other hand, assisted users restricted their queries with the year field, switched between free–text and title fields, or used the author field significantly more often then users in the other group (7.08 vs. 0.92, $p = 0.001$). In fact,

even though users in both groups received the same written introduction to the fielded search interface, and all search fields were visible at all times, users in the unassisted group generally ignored all other fields but the one chosen for their first query (either "title" or "free–text") and used this for all searches in all tasks.

The assisted group used less query moves in total, but instead employed more advanced actions using the other tools of the system. No significant differences could be observed in the use of the thesaurus for looking up more general or specific terms, or to get word definitions or synonyms. However, users in the assisted group made significantly more use of the extraction capabilities of DAFFODIL, to get new terms or common authors from result sets or relevant documents (4.91 vs. 0.5, $p = 0.01$). They also used more terms from other tools (like the displays of related terms, extracted terms, the thesaurus, or the classification) directly in their query (4.67 vs. 2.33, $p = 0.039$).

Research question 3. The last question examined was, if users of the suggestion tool would independently use advanced search actions after having seen them suggested for previous tasks. While this can really only be answered with a long–term experiment, a first test was done as part of this user experiment. For the assisted group, the suggestion tool was switched off for the last experiments, so that no suggestions were displayed to the users.

While there was a clear drop in advanced actions used by the group between tasks 2 and 3 (see table 3), they still used significantly more advanced actions than the unassisted group (2.67 vs. 1.25, $p = 0.02$). Similarly, the usage of additional search fields was higher among the users who had previously received search suggestions (1.75 vs. 0.25, $p = 0.004$).

This could be interpreted as there being a slight learning effect from receiving situationally appropriate suggestions, that led those users to employ them independently. Of course, the three tasks were performed in immediate succession, so it remains unclear if they would retain this knowledge for future search tasks. On the other hand, a more pronounced effect might have been observed during a long term study, where searchers use the suggestion tool over a much longer period of time than just during two 20 minute search tasks.

5   Summary and Conclusion

In this paper a system for providing adaptive search suggestions based on the situations of previous users was presented. The system uses case–based reasoning techniques and although it was implemented within the framework of the DAFFODIL search system for digital libraries, the basic principle is not dependent on the search system used, and has also been adapted for Google web search (implemented as an add–on for the open source browser Firefox) [2].

A user experiment was conducted to evaluate the suitability of the suggestion system for helping nonexpert users to search more successfully and to utilise more advanced tools and search tactics.

The results of the experiment were positive. Users with assistance were able to search more successfully, as measured by the number of relevant documents

saved — at least for some tasks. It seems that not all tasks benefit from the suggestions in their current form. If this results from a lack of suitable cases, is due to the specific search task, or points to a more general problem of the approach needs further examination. For those tasks where users were helped by the suggestions, the differences were very significant ($p = 0.011$ and $p = 0.022$).

Furthermore, users who received suggestions used more of DAFFODIL's tools and a greater variety of search fields in their tasks. This hold true even for the task which they performed unassisted. It would be interesting to examine how these differences in tool use change during a longer experiment covering multiple search sessions over several weeks. Additional analyses on the search and tool use logs could be done, in particular to identify search tactics and strategies from the query logs (as done e.g. in [9]). It is an open question if the use of search suggestions leads users to more sophisticated search strategies.

References

1. Anne Aula and Klaus Nordhausen. Modeling successful performance in web searching. *Journal of the American Society for Information Science and Technology*, 57(12):1678–1693, 2006.
2. Marcel Awasum. Suggestions for Google websearch using a firefox add–on. bachelor thesis, University of Duisburg-Essen, 2008.
3. Marcia J. Bates. Information search tactics. *Journal of the American Society for Information Science*, 30(4):205–214, 1979.
4. Marcia J. Bates. Where should the person stop and the information search interface start? *Information Processing and Management*, 26(5):575–591, 1990.
5. Nicholas J. Belkin, Colleen Cool, Adelheit Stein, and Ulrich Thiel. Cases, scripts, and information-seeking strategies: On the design of interactive information retrieval systems. *Expert Systems with Applications*, 9(3):379–395, 1995.
6. Nicholas J. Belkin, P.G. Marchetti, and Colleen Cool. BRAQUE: Design of an interface to support user interaction in information retrieval. *Information Processing and Management*, 29(3):325–344, 1993.
7. Suresh K. Bhavnani, Bichakjian K. Christopher, Timothy M. Johnson, R. J. Little, F. A. Peck, J. L. Schwartz, and V. J. Strecher. Strategy hubs: next-generation domain portals with search procedures. In *Proceedings of the conference on Human factors in computing systems*, pages 393–400. ACM Press, 2003.
8. Giorgio Brajnik, Stefano Mizzaro, Carlo Tasso, and Fabio Venuti. Strategic help in user interfaces for information retrieval. *Journal of the American Society for Information Science and Technology*, 53(5):343–358, 2002.
9. Carola Carstens, Marc Rittberger, and Verena Wissel. How users search in the german education index - tactics and strategies. In *Proceedings of the workshop Information Retrieval at the LWA 2009*, 2009.
10. Karen M. Drabenstott. Do nondomain experts enlist the strategies of domain experts. *Journal of the American Society for Information Science and Technology*, 54(9):8836–854, 2003.
11. Bob Fields, Suzette Keith, and Ann Blandford. Designing for expert information finding strategies. In Sally Fincher, Panos Markopoulos, David Moore, and Roy A. Ruddle, editors, *BCS HCI*, pages 89–102. Springer, 2004.

12. Norbert Fuhr, Claus-Peter Klas, André Schaefer, and Peter Mutschke. Daffodil: An integrated desktop for supporting high-level search activities in federated digital libraries. In *ECDL 2002*, pages 597–612, Heidelberg et al., 2002. Springer.

13. Stephen P. Harter. *Online information retrieval: concepts, principles, and techniques*. Academic Press Professional, Inc., San Diego, CA, USA, 1986.

14. Stephen P. Harter and Anne Rogers Peters. Heuristics for online information retrieval: a typology and preliminary listing. *Online Review*, 9(5):407–424, 1985.

15. Bernard J. Jansen. Seeking and implementing automated assistance during the search process. *Information Processing and Management*, 41(4):909–928, 2005.

16. Bernard J. Jansen and Michael D. McNeese. Evaluating the effectiveness of and patterns of interactions with automated searching assistance. *Journal of the American Society for Information Science and Technology*, 56(14):1480–1503, 2005.

17. Kalervo Järvelin. Explaining user performance in information retrieval: Challenges to ir evaluation. In *ICTIR 2009*, pages 289–296, 2009.

18. Claus-Peter Klas, Hanne Albrechtsen, Norbert Fuhr, Preben Hansen, Sarantos Kapidakis, László Kovács, Sascha Kriewel, András Micsik, Christos Papatheodorou, Giannis Tsakonas, and Elin Jacob. A logging scheme for comparative digital library evaluation. In *Proceedings of ECDL 2006*, pages 267–278, 2006.

19. Claus-Peter Klas, Norbert Fuhr, and André Schaefer. Evaluating strategic support for information access in the DAFFODIL system. In *Proceedings of ECDL 2004*, pages 476–487, 2004.

20. Sascha Kriewel and Norbert Fuhr. Adaptive search suggestions for digital libraries. In *Asian Digital Libraries: Looking Back 10 Years and Forging New Frontiers (ICADL 2007)*, pages 220–229, 2007.

21. Karen Markey. Twenty-five years of end-user searching, part 1: Research findings. *Journal of the American Society for Information Science and Technology*, 58(8):1071–1081, 2007.

22. Santiago Ontañón and Enric Plaza. Justification-based multiagent learning. In Nina Mishra Tom Fawcett, editor, *The Twentieth International Conference on Machine Learning (ICML 2003)*, pages 576–583. AAAI Press, AAAI Press, 2003.

23. Annabel Pollock and Andrew Hockley. What's wrong with internet searching. *D-Lib Magazine*, March 1997.

24. Soo Young Rieh and Hong (Iris) Xie. Patterns and sequences of multiple query reformulations in web searching: a preliminary study. In *Proceedings of the 64th Annual Meeting of the American Society for Information Science and Technology*, volume 38, pages 246–255, 2001.

25. André Schaefer, Matthias Jordan, Claus-Peter Klas, and Norbert Fuhr. Active support for query formulation in virtual digital libraries: A case study with DAFFODIL. In *Proceedings of ECDL 2005*, 2005.

26. Barbara M. Wildemuth. The effects of domain knowledge on search tactic formulation. *Journal of the American Society for Information Science and Technology*, 55(3):246–258, 2004.

27. Hong Iris Xie. Shifts of interactive intentions and information-seeking strategies in interactive information retrieval. *Journal of the American Society for Information Science*, 51(9):841–857, 2000.