M. Koch, A. Butz & J. Schlichter (Hrsg.): Mensch und Computer 2014 Tagungsband,
München: Oldenbourg Wissenschaftsverlag, 2014, S. 135-144.

Trail Building During Complex Search Tasks

Sebastian Franken[1], Ulrich Norbisrath[2]

Fraunhofer FIT, Sankt Augustin, Germany[1]
ulno.net, Gillette, WY, USA[2]

Abstract

Users frequently perform searches that do not only have simple fact retrieval in mind, but aim for more comprehensive results. These complex search tasks need to be distinguished from simple search tasks, where an information need can be fulfilled by simple fact finding. While search engine support for simple search tasks is satisfactory, it lacks in support for complex search tasks, leading to an increasing discontent about the interaction with the search engine of choice. In this paper we address the question, how to give tool support for complex search tasks. We present the logging framework 'SearchTrails', which captures the course of user actions of any search task and presents a corresponding graph visualization. We conducted a qualitative user study of SearchTrails and present first promising results.

1 Introduction

Literature review indicates discontent with popular search engines to fulfill some particular information needs. It turns out that this discontent is mainly visible in search tasks which are classified as exploratory or complex (Marchionini 2006; Aula & Russell 2009; Singer et al. 2012). In this paper we address the research question of how it is possible to give tool support for complex search tasks. We present a logging framework which is able to capture the course of any exploratory or complex search and allows presenting a graph visualization of it. We carried out an initial qualitative study using this new framework involving seven participants. The results indicate interest and acceptance of our chosen support method and outline benefits, shortcomings, and possible future improvements.

Users frequently perform searches that aim for more than simple fact retrieval. These searches aim for more comprehensive results, stretch over a certain amount of time, and are not completed within one session. Examples for search tasks like these could be: booking a holiday trip, building some technical device on your own, or staying informed about the market situation for one specific appliance over a period of time. Personal experience shows that we stumble upon interesting web sites during the course of such searches, but quickly move on from preliminary results to other web sites that seem more promising. Even if addresses of the visited pages can still be retrieved from the browser cache, the important pages are usually lost. Some days later, searchers might be interested again in the information on the intermediate sites, but are not able to get back to them. Our study participants also reported these situations. This is where an idea of (Bush 1945) comes into place: He suggests that navigating through the hypertext is stored in so-called 'trails' that could later be recalled and extended. This idea gets picked up by (Bates 1989), who already suggests that the search for information is comparable to the process of 'Berrypicking', in which many highly qualitative information bits get picked during a search. She opposes this to classic information retrieval, in which a query should be answered by only one perfectly matching document. Today's search engines get better and better in helping users to find fact-based information, but they lack support for gathering many pieces of information on a specific topic (e.g. Booking a holiday trip, which includes comparing different ways of traveling, different hotels, and related activities) (Singer et al. 2012). The well-supported fact searches are often part of connected complex search tasks. These sets of connected searches need to be supported. The approach presented here does exactly this: it visually logs the users' paths when traversing the web, offering possibilities to capture specific information and providing a way back to information alternatives seen in earlier stages of the search process.

Arguably, bookmarking the valuable pages might solve the mentioned problems. However, this does not work in the aforementioned cases, where users retroactively start appreciating the value of a visited website after they have already left it. We monitored that people hesitate to bookmark sites to avoid bookmarking too many sites. They also often end up being stored unsorted in the bookmark folder or a browser toolbar. Managing bookmarks as a part of the follow-up process of search is time consuming, especially with the help of the built-in browser bookmark managing tools. Social bookmarking services like Delicious[1], Tumblr[2], or Pinterest[3] also do not help in our cases, as they do not store the context in which a bookmark was created and require similar follow-up work as browser bookmarks. Recovering sites via the back button or the browser history is also tedious; especially when a user has visited many pages on a topic. If the visit lies back in time, relocating the page gets even harder.

(Singer et al. 2012a) show that the aforementioned discontent is based in the character of the search task, which is not supported well by todays' search engines. Current search engines perform well on lookup or fact finding tasks (Jansen 2006) and therefore yield positive user satisfaction in this discipline (Singer et al. 2012a). They are trained on fact retrieval and on delivering exactly matching documents to the searcher (Marchionini 2006). This type of search could also be described as a kind of one-shot-search, i.e. a 'search and forget' mode, in which for a given query the search results are returned, but then forgotten by the search engine. (Broder 2002) and (Rose & Levinson 2004) both analyzed and classified search tasks in the types of their underlying motivation. They showed that in 50-60% of search tasks, users are looking for more than simple fact retrieval or lookup of sources, but are rather looking for information that they assume not to find on one specific site, but a collection of pages, where every page adds to the overall result. Complex or exploratory search deals exactly

[1] https://delicious.com/, Accessed 2014/03/17
[2] https://www.tumblr.com/, Accessed 2014/03/04
[3] https://www.pinterest.com/, Accessed 2014/03/04

with these types of search, and in this contribution, we combine Bush's idea of logging and building visual trails through the Internet with the added benefit of metadata and the added possibility of storing selected parts of websites to support these types of search.

Consider as an example a search task about crypto currencies: 'What are they, what is their current status, how do you obtain them?' During a web search, people discover pretty early the page about crypto currencies from Wikipedia[4] and learn about bitcoins, bitcoin clients, and some of their properties, like being limited and community supported. In our set of test users, they often also discover some properties about currencies in general (fiat money, trading of currencies). Usually, test users study a lot about bitcoins and their inner workings and find some sites where one can purchase them. They usually also forget things they discovered in the beginning, like how a proof-of-work works or where they could find a simple introduction to the cryptography related to bitcoins. Also, the idea of mining becomes more interesting the longer one studies the topic of crypto currencies, which makes people want to go back to the sites they previously visited and gather relevant information from these pages.

In the following sections, we will look at the underlying theoretical concepts, as well as existing search loggers, then describe the architecture and implementation of our search support tool, present a small user study and its qualitative results. We will conclude with a discussion of the results and an outlook on the work to be done in the future.

2 Theoretical background and related work

First approaches to identify different natures of search tasks start with (Broder 2002), who identifies navigational, transactional, and informational search tasks by the query entered into the search engine. When performing navigational queries, users have in mind reaching a particular site, expecting to find information on that site. With transactional queries, users try to find a page where some action should be performed, e.g. accessing a database or downloading data. With informational queries, users try to find information that should be processed by reading and either satisfies the information need, or triggers a new query to refine the information. An interesting observation is that "in almost 15% of all searches the desired target is a good collection of links on the subject, rather than a good document" (Broder 2002). Broders' work got independently refined later by (Rose & Levinson 2004); their results match in their core with Broders' results. They also developed a trichotomy of user goals, consisting of navigational, informational, and resource-oriented (instead of transactional) goals. Additionally, the authors added a sub-classification of goals. While Broder states that 39-50% of all queries are informational, Rose and Levinson show that up to 60% of all queries are informational. (Lewandowski 2006) shows that almost 50% of all queries sent to search engines are informational, while navigational queries account for a much larger share of queries than transactional ones. Considering that informational search tasks tend to be time consuming and span over multiple sessions, these search tasks count for a large

[4] www.wikipedia.com, Accessed 2014/03/24

share of time consumed when searching (Singer 2012). This shows the importance of exploratory or complex search tasks.

Early research of structuring user goals in web search laid the foundations for more formal definitions of search tasks. Search activities were grouped by (Marchionini 2006) into the three overlapping categories of 'Lookup', 'Learn', and 'Investigate' (gray part in Fig. 1). 'Lookup' comes close to navigational and transactional search tasks, as it consists of e.g. fact retrieval, navigation, transaction, or verification. 'Learn' consists of more complex activities, such as e.g. knowledge acquisition, comparison, or aggregation, while 'Investigate' consists of analysis, synthesis, or evaluation. Exploratory search, as framed by Marchionini, can be seen as a combination of the search activities of Learn and Investigate. He states that 'investigative searching is more concerned with recall […] than precision […] and thus not well supported by today's Web search engines.' A definition of exploratory search is given in (White et al. 2008), stating that 'Exploratory search can be used to describe an information-seeking problem context that is open-ended, persistent, and multi-faceted; and to describe information-seeking processes that are opportunistic, iterative, and multi-tactical. In the first sense, exploratory search is commonly used in scientific discovery, learning, and decision-making contexts. In the second sense, exploratory tactics are used in all manner of information seeking and reflect seeker preferences and experience as much as the goal'.

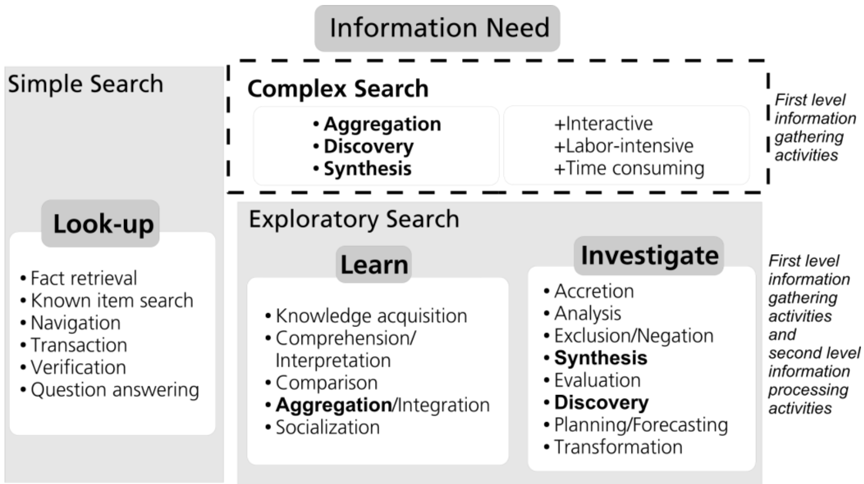

Figure 1: Complex and exploratory search (from (Singer 2012)).

Figure 1 shows that the core components of exploratory search are 'Learn' and 'Investigate', each of them again containing different subtasks. These subtasks imply relatively high cognitive effort, especially when it comes to subtasks like analysis, evaluation, or planning. The necessary cognitive effort is hard to measure by purely implemented solutions, like we are presenting here. Therefore, we stick to the definition of complex search, as presented in (Singer 2012), who defines complex search as "tasks where users are required to follow a multi-step and time consuming process that is not answerable with one query, requiring synthesized information from more than one retrieved web page or document to be solved. The process to work to complex search tasks usually comprises at least one of the process steps aggregation, discovery, and synthesis." This avoids the cognitively high loaded aspects of Marchionini's definition. It has the added benefit of covering complex search activities with limited cognitive loads into the definition, which would not be covered by Marchionini, e.g. checking the availability and prices of certain products in a number of web portals. However, complex search tasks often require synthesis of the information found or even yield discoveries which extend or alter the initial search task goal.

To find out how users cope with the challenges of complex search tasks, some systems have been developed which try to register the user interactions during the search process. In most cases, these search logging systems were constructed as browser plug-ins, storing information about the visited pages and sometimes posing questions to the users while performing certain tasks. An early approach was an Internet Explorer plug-in with built-in questionnaires (Fox et al. 2005) that was used to compare explicit and implicit measures of user satisfaction. Another approach was the Wrapper system (Jansen et al. 2006) which was installed to the user's system and logged the interaction with the browser and the browser's interaction with the system. It could be shown that users 'may seek information over an extended period [of time] and on multiple information systems'. The SearchLogger system was built by (Singer et al. 2011). It consists of a Firefox plug-in allowing configuring a list of search-tasks and selecting one of them, recording all user interactions with the browser, and storing this information centrally on a logging server to be evaluated. It comes with some guidelines how to evaluate the logged data. However, logged information is not visible to the user searching. Therefore, there is no direct feedback to the user, which could facilitate the search task. SearchLogger was mainly used to proof the existence of complex search and to allow its classification. Based on the observations derived from the user studies carried out with SearchLogger, a user centric model for increasing user satisfaction was developed.

Early approaches on collaborative search, such as the co-browsing idea in (Gross 1998) or collaborative bookmarking in the Social Web Cockpit (Prinz & Graether 2000) supported remotely controlled browsers for synchronous browsing or creation of community bookmarks, but no cooperative, context-preserving, or asynchronous exploration of information spaces. Faceted search can alleviate information complexity in very specific cases, e.g. when databases are preprocessed with large effort, or by relying on object-unspecific metadata (Franken & Prinz 2009). These approaches can only offer very little support if it comes to complex search cases, as these span over time and a multitude of resources.

The search logging systems presented here mainly focus on gathering data from the users via browser logging, system logging, and questionnaires. Our tool goes beyond that, as it uses the information gathered to actively support the users in their tasks. In the next section, we will present an overview over the newly developed system and outline key characteristics of our user study.

3 The SearchTrails tool

With our tool, we address the missing support during complex search. In contrast to the search loggers described before, our goal is not only to monitor the user's search behavior but also to provide immediate feedback and support to address discovery, aggregation, and synthesis needs while users carry out their search tasks.

To achieve this, we decided to build a plug-in for the chrome browser[5], which logs the users' interactions with the browser (Figure 2). This approach has several advantages: While providing full access to the Internet and its resources, the plug-in monitors all interactions with the browser (and the Internet) and stores the relevant actions unobtrusively. Therefore, we are not relying on any preprocessed data bases (like in faceted solutions) and allow free-form, open-ended search processes. We can omit intermediate and interrupting questions, and have very minor installation effort. The internal structure consists of three parts: The logging engine logs the events and builds a logical representation of the search process. The storage engine stores the log as a JSON-object and reports about the changes, which are visualized by the rendering engine. The storage engine also stores the log on a remote server.

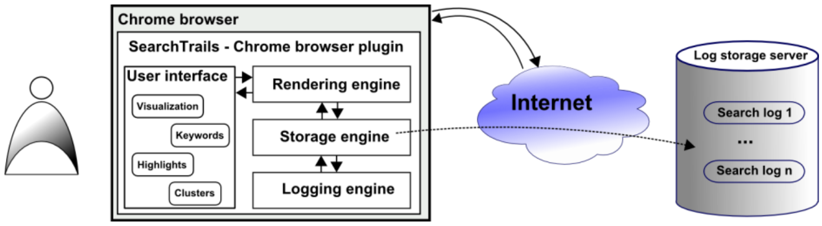

Figure 2: Schematic view of the SearchTrails architecture.

SearchTrails itself works as follows: It monitors the opening, closing, and switching between tabs and the change of URLs with the help of a logging engine that logs, filters, and interprets the user interactions and generates the metadata for the visited URLs. The visited URLs get transformed into nodes in the graph visualization. The visualization is done by the rendering engine and is shown in a separate tab, which is excluded from the logging itself. It is based on a bidirectional graph in a forced-directed layout (see Figure 3). For each visited URL, important keywords get stored and displayed in a table next to the visualization, if a keyword appears on two or more pages the user visited. This metadata gets also stored in the search logs. To ease the synthesis of information (as one of the key points of complex search), selected text can be stored on a key press and gets displayed with its original URL in a highlights table. Nodes belonging to the same host are clustered by a colored hull and can be reduced on click to reduce complexity of the visualization, but still being sensible to user interactions, even if not being visible. The storage engine stores the search logs remotely on a server, without requiring user interaction. As our tool does not judge the correctness of search results, it also stores irrelevant search results. This can be valuable in terms of showing that desired information was not present at certain pages. Fig. 3 shows an example visualization for the 'Beethoven' task, with the search trail visualized (1) (which shows a rather linear approach of searching, while trails for more complex search tasks may show more cyclic approaches), a colored node (2), symbolizing that there are highlights appended to it,

[5] https://www.google.de/intl/de/chrome/browser/, Accessed 2014/03/05

the highlights-overview being turned on (3), a list of keywords to the right (4), and a cluster with 3 nodes from the same host (5) (all nodes from Youtube).

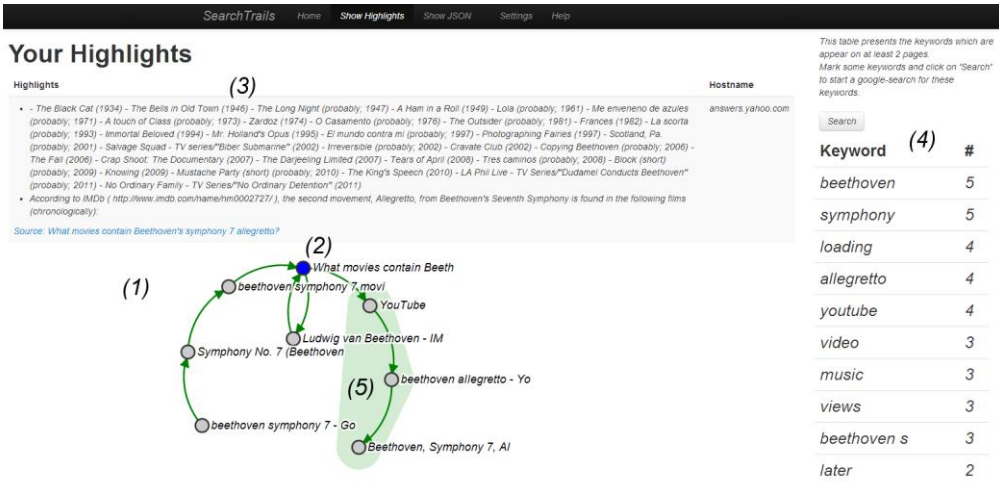

Figure 3: An example Search Trail and highlights for the ‘Beethoven’ search task.

4 User study

The evaluated version of SearchTrails evolved from a domain survey of related work and personal interviews with potential users of the system. A user study was needed to verify the design decisions made and to reveal advantages and flaws of the system. We did a study with seven participants with academic background, performing complex search tasks with and without the help of SearchTrails. The authors were not taking part in this study.

We generated a set of 4 search tasks. One is a simple search task (finding films using Beethoven’s 7th symphony as a score), and three tasks are examples for complex search (‘Building Your Own Video Glasses’, ‘Checking the Availability and Transport Possibilities for a Product’, and the aforementioned ‘Crypto Currency’ search task). We asked the participants to dismiss one uninteresting task for the evaluation to ensure that there is a personal engagement in the selected search tasks. As the creation of complex search tasks is sensible, we checked our search tasks against the four criteria given by (Kules & Capra 2009). First, the tasks should indicate uncertainty and ambiguity in the information need. We ensured this by posing open-ended questions that could be tackled from various starting points. Second, the tasks should suggest knowledge acquisition, comparison or discovery. As our tasks are from domains that our test users were unfamiliar with, we could ensure to fulfill this criterion. Third, the tasks should provide a low level of specifity about the information necessary and how to find the information. We ensure this by leaving open how and which information should be gathered. Fourth, the task should provide enough imaginative context in order for the users to relate and apply the situation. We generated tasks with a practical focus, to ensure our participants could easily engage in them.

The user study starts with the installation of SearchTrails and a practical introduction to make sure the users understand its features and know how to handle them. Of the three tasks the participants had to perform without supervision, we selected the simple search task for the start. This allows our participants to get acquainted with the system and to have a negative probe in terms of looking at complex tasks. Of the following two complex search tasks, one is carried out with the help of SearchTrails. This means that the users have the system on a second screen, while actually searching on the first screen. Another task is done without SearchTrails being visible; it is running in the background without the participants being able to look at it. Participants fill out a small questionnaire with 5-point Likert and open-ended questions after each search task. In addition, we analyzed the remotely stored search logs.

5 Evaluation

Doing an evaluation of a complex search support system with visualization is a difficult task: 'the evaluation of visualization systems that can support exploratory search tasks is not well understood' (Koshman 2006). Koshman also mentions testing metrics affecting the evaluation of exploratory search visualization systems, which helped us to make the results comparable with later evaluations. Among her metrics are: Training length, in our case long enough to make the user know how to handle the system; Task type, timing and purpose, in our case complex tasks for non-domain-expert users, with no time limit; Number of visualization system features, here an operational prototype with the aforementioned features; Type of visualization, here a force-directed graph layout; User types, here information professionals, but no domain experts considering the tasks; Response type, in our case a combination of 5-point Likert and open ended questionnaire items; and the speed of the system, which was asked in the questionnaire, and turned out to be sufficient.

In this initial, qualitative user study we already got some very interesting results. Task 1 is a simple search, as it could be fulfilled with the following search query *'wiki beethoven 7th symphony'* and visiting the first Wikipedia entry. Participants liked the highlighting function of our tool in that search task. We suppose that the highlighting stores important results and therefore enables the user to continue a search, although the core results are already found (User quote: '*pro: Saving results for later reuse*'). It might offer a feeling of security to the user to be able to go back to earlier findings ('*visualize the path especially when stepping back*').

As we expected, the search tasks were new to our participants ('*new topic for me, a lot of information*'), and they were motivated to explore the new topics ('*exploring a topic I've been wondering about for a while*'); therefore our example tasks matched the criteria by Kules & Capra mentioned above. Considering the second task, where participants should search without the help of SearchTrails, they reported that the task '*was quite tedious, [as] it required exploration of dozens of web sites, changing search queries several times, and subdividing queries.*' Comments to our third task (searching with the help of SearchTrails) were also very encouraging: Participants liked the '*Visualization of visited pages*', the possibility to store '*text snippets from the web pages*', and '*modifying search queries (→ keywords)*'. Participants also appreciated the clustering of nodes: '*Pro: Clustering nodes*'.

In general, the synthesis feature in SearchTrails yielded very positive comments. Five of our seven participants mentioned that the synthesis feature allows easy selection and compilation of relevant phrases. Highlighting was extremely helpful and alone justified the use of our tool ('*I will now miss the highlighter*'). Concerning keywords, the results were mixed. While one user mentioned that there are too many irrelevant keywords ('*keyword list was helpful, but contained irrelevant entries*'), two other users used the feature and found it interesting to be pointed to other relevant search terms ('*"Finding" new search queries based on keywords*' and '*It helped to identify the keyword related to the subject of crypto*'). The overall positive reactions of our study participants are encouraging to extend and improve SearchTrails.

6 Discussion and outlook

Information synthesis comprises the steps of information ordering, automatic editing, information fusion, and information compression (Singer et al. 2012). When it comes to synthesis, our literature survey showed that this aspect of complex search is not supported at all in the present search systems. With our system, we can offer fast and easy synthesis of diverse web-snippets and still capture their context. The responses of our participants show that this is the most important feature in the case of individual use. With respect to our research question, the results indicate that building a search trail is a means to support complex search tasks. The keyword suggestions and the clustering mechanism provide additional benefit to users. Nevertheless, some flaws were revealed in our study: Highlighting parts from pdf-files does not always work, there were irrelevant stopwords, manual adding and removing of highlights is missing, and the possibility of removing nodes from the graph is not yet provided. We will implement the support for recreating search trails, allowing extending a search and handing over a search trail in order to search collaboratively. Overall, the evaluation results are very promising and will positively influence the future work on SearchTrails.

Acknowledgements

Thanks to our study participants, notably Kamila and Zarina from Nazarbayev University, Astana, and Constantin, Nils, Wolfgang, Maren, and Urs from Fraunhofer FIT.

References

Aula, A., Russell, D.M. (2009). Complex and Exploratory Web Search. In: Marchionini, G. et al., Information Seeking Support Systems: An Invitational Workshop (06/2008, Chapel Hill), pp. 23-24.

Bush, V. (1945). As we may think. In: The Atlantic Monthly 176, July 1945.

Bates, M.J. (1989). The Design of Browsing and Berrypicking Techniques for the Online Search Interface. Online Information Reviews, Vol. 13, Issue 5, pp. 407-424.

Broder, A. (2002). A taxonomy of web search. In: SIGIR Forum, Fall 2002, Vol. 36, No. 2, pp. 3-10.

Fox, S., Karnawat, K., Mydland, M., Dumais, S., White, T. (2005). Evaluating Implicit Measures to Improve Web Search. In: ACM Transactions on Inf. Syst. Vol.23, No.2, Apr. 2005, pp. 147-168.

Franken, S., Prinz, W. (2009). FacetBrowse: Facettenbasiertes Browsen im Groupware Kontext. In: Wandke, H., Kain, S., Struve, D. Mensch & Computer 2009, Oldenbourg, pp. 123-132.

Gross, T. (1998). CSCW3: Transparenz- und Kollaborationsunterstützung für das WWW. In: Herrmann, T., Just-Hahn, K.: Proc. of D-CSCW'98, pp. 37-50.

Jansen, B.J. (2006). Using Temporal Patterns of Interactions to Design Effective Automated Searching Assistance. In: Communications of the ACM, April 2006/Vol. 49, No. 4, pp. 72-74.

Jansen, B.J., Ramadoss, R., Zhang, M., Zang, N., (2006). Wrapper: An Application for Evaluating Exploratory Search Systems Outside of the Lab. In: Proceedings of SIGR 2006, Aug. 6-11, Seattle.

Koshman, S., (2006). Exploratory Search Visualization: Identifying Factors Affecting Evaluation. In: Proc. of the ACM SIGIR 2006 Workshop on "Evaluating Exploratory Search Systems", pp. 20-23.

Kules, B., Capra, R., (2009). Designing Exploratory Search Tasks for User Studies of Information Seeking Support Systems. JCDL '09, June 15-19, Austin, Texas, p. 419.

Lewandowski, D. (2006). Query types and search topics of German Web search engine users, In: Information Services and Use, Vol. 26, Jan. 2006, pp. 261-269.

Marchionini, G. (2006). Exploratory Search: From Finding to Understanding. In: Communications of the ACM, April 2006/Vol. 49, No. 4, pp. 41-46.

Prinz, W., Graether, W. (2000). Das Social Web Cockpit: Ein Assistent für virtuelle Gemeinschaften. In: Reichwald, R., Schlichter, J.: Proceedings of D-CSCW 2000, pp. 127-138.

Rose, D.E., Levinson, D. (2004). Understanding User Goals in Web Search. In: Proceedings of WWW 2004, May 17-22, New York, pp. 13-19.

Singer, G. (2012). Web search engines and complex information needs. Dissertationes Mathematicae Universitatis Tartuensis, Order No. 446, ISSN: 1024-4212.

Singer, G., Danilov, D., Norbisrath, U. (2012). Complex search: aggregation, discovery, and synthesis. In: Proceedings of the Estonian Academy of Sciences, 2012, **61,** 2, pp. 89-106.

Singer, G., Norbisrath, U., Lewandowski, D. (2012a). Ordinary Search Engine Users assessing Difficulty, Effort, and Outcome for Simple and Complex Search Tasks. In: Proc.IiiX'12, Nijmegen.

Singer, G., Norbisrath, U., Vainikko, E., Kikkas, H., Lewandowski, D. (2011). Search-Logger - Analyzing Exploratory Search Tasks. In: SAC'11, March 21-25, Taichung. Taiwan, pp. 751-756.

White, R.W., Marchionini, G., Muresan, G. (2008). Evaluating Exploratory Search Systems. In: Information Processing and Management: An Int. Journal, Vol. 44/2, March 2008, pp. 433-436.

Contact information

Sebastian Franken, Fraunhofer FIT, sebastian.franken@fit.fraunhofer.de
Dr. Ulrich Norbisrath, ulno.net, research@ulno.net