Supporting Exploratory Search in User Interface

Ilkka Kiistala
Department of Computer Science
University of Helsinki
ilkka.kiistala@helsinki.fi

Tuire Peurala
Department of Computer Science
University of Helsinki
tuire.peurala@helsinki.fi

ABSTRACT

Here we'll describe the content of our essay.

Author Keywords
Exploratory Search; Information Retrieval; User Modeling.

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

General Terms
Human Factors; Design; Measurement.

INTRODUCTION

In this literature review we introduce the problem of exploratory search and some solutions for the problem. We cover three techniques that help the user in succeeding in his exploratory search task efficiently. We explain what exploratory search is and what distinguishes it from other information retrieval tasks.

To build a good system where machine and a human cooperate to perform a task it is important to take into account some significant characteristics of people [18]. User model is built of these characteristics and traditionally it has been a model of a typical user [18].

In reality often the users vary so much that a traditional user model is insufficient and there is a need for a user model of an individual.

Define ES

[23]: In order to accomodate to differing needs of users or usergroups over time, a system may use one of three basic approaches. System is called adaptive if it alters it structure, functionality or interface on the basis of a user model generated from *implicit* user input. Adaptable systems use *explicit user input* and need user's active participation. Personalized system is a hybrid of the two aforementioned.

In the following chapter we introduce basic concepts, solutions and systems that we found exploring the literature. In the Discussion chapter we summarize our findings and point out what is missing and what we expect from future research. The Conclusion chapter summarizes our findings.

LITERATURE REVIEW

(User Modeling) [18], [5]

Exploratory Search

(Ilkka)

Introduction to exploratory search. [14], [29], [22]

The user interface of an exploratory search system should be designed to fulfill the needs of most of its users. More information on what works and doesn't work can usually be collected from system evaluations.

However, evaluating exploratory search systems is difficult, because users have different starting positions. Their knowledge of the domain varies, they are interested in different aspects of the topic and they have previously encountered different information. [13]

Exploratory search tasks can be characterized as either learning oriented or investigative and they have common aspects like uncertainty, ambiguity and discovery distinguishing them from look-up oriented tasks [12].

User Modeling

(Tuire)

In the first ten years of Human-Computer interaction history the focus of research was in user interfaces and particularly the possibilities and design criteria of graphical user interfaces [5]. Later the focus has moved toward more broadly improving the way people use computers for example to work, think, communicate and learn [5]. To build a good system where computer and a human cooperate to perform a task it is important to take into account some significant characteristics of people [18]. User models are built of these characteristics and are used to adapt the system toward the user. We define user models in this literature review as ”models that systems have of users that reside inside a computational environment” [5].

A central objective of user modeling is to address the problem that system will be unable to interact with users cooperatively unless they have some means of finding out what the user really knows and does [5]. To find out this there are two communication channels the explicit user input and the implicit information that is gathered without any effort from the users. The data gathered by these means is then used to improve the interaction with the system with the background knowledge of the problem domain, communication processes and the user agents involved (see Figure 1).

Figure 1. Knowledge-based HCI [5].

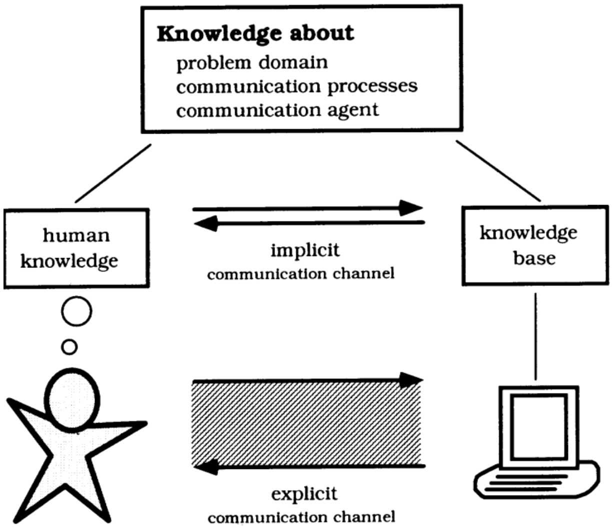
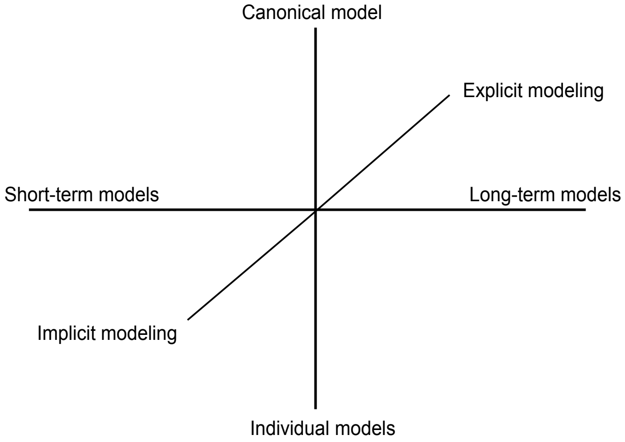


Figure 2. The dimensions of user modeling.



In order to accomodate to differing needs of users or user-groups over time, a system may use one of three basic approaches [23]. System is called *adaptive* if it alters it structure, functionality or interface on the basis of a user model generated from *implicit* user input. *Adaptable* systems use *explicit* user input and need user's active participation. *Personalized system* is a hybrid of the two aforementioned. In personalized systems the output or appearance differs for every user or user group in every context [23]. The adapted output has the potential to be of great benefit for users; it is geared towards the user's preferences, behaviour or needs and it can make interaction easier and a lot more fruitful. A life-cycle of a computational system cannot anymore be divided to the design time and use time as the behaviour of system can adapt to contextual information like the user goals only known at the use time. The differentiation between the design time and the use time of the system gets blurred with user modeling and the use time becomes design time [5].

The different types of user models can be mapped as a three dimensional space where the axes are: single model vs. collection of models, explicitly specified models vs. models inferred by the system on the basis of user behaviour and long-term vs. short-term user models [18]. The user model classification map is depicted in Figure 2. and the three dichotomies are explained further in this chapter.

The *single model vs. collection of models* polarity [18] has on it's other end the single model of single user. The modeled single user is called the canonical user and is actually not a real person at all but a modeled stereotypical user of the system. Traditional user models have been constructed by collecting data on an average user on various tasks and environments [18]. An example of a canonical user model is Fitt's law that suggests that the speed on which the user operates the machine can be increased by increasing the size of targets the user must hit. The major weakness of these kind of models is that they assume that all the users constitute a homogenous set. In most cases for the majority of users the system is better adapted to them that would be without any adaptation, but it isn't likely the best system that could be produced [18]. In the other end of this dichotomy is the collection of models of individual users. These users are real and the facts to build the individual model are gathered usually from the interaction of the user with the system. When choosing whether to use a canonical or individual user models one should think if the canonical model is sufficient or is the user community too heterogenous for it to be useful enough [18]. The decision on this axis influences the other aspects of user modeling. If one user model is used then it has to be designed only once and the system doesn't have to prepare for incorrect or conflicting user model information as the case is on the latter option.

The way how the user model is specified forms the second dichotomy mapping the user modeling techniques; *explicit vs. implicit models* [18]. On the explicit side the model is specified explicitly either by the system designer or by the users themselves. User model may be formed by letting the user modify the system but this leaves a lot on the hand of the user and fits badly to systems that expect to have lot of users that use the system only once or twice. Explicit user modeling can be done also by asking questions directly from the users but one problem with this is that users can't always answer on what the system needs to know. On the other end of the axis models are formed more subtly, they are inferred by the system only on the basis of user's behaviour. Implicit user models are built by gathering information of the user when the user interacts with the system. When constructing the user model implicitly the system must have some way of distinguishing incorrect information and resolving conflicts between gathered information.

In order to adapt soundly to the user the system must have wide variety of information about the user. The information might be ranging from short-term facts like the last query the user performed to more long-term facts like the level of mathematical sophistication of the user. This forms the third dichotomy: *long-term vs, short term user models* [18]. Long-term models can be built in the course of series of interactions between the system and the user. Long-term model could be built for example in a bookstore application where information of the user's purchases might be gathered to to recommend books of certain genre for the user. Short-term user model could be used in a search application. The model would be built by gathering recent queries to form a model of the user's search goal. Short-term user model is needed to respond to a situation where a user would need different search results when searching with keyword Java after searching information on programming or after the user has been searching beaches in Indonesia. There are also other differences in user models besides these introduced three dichotomies. Example of these differences is the model lifecycle; in a system where there is only a single model it can be permanently embedded with the system but in the case where there are models for each individual user the model is built on the fly. This and many other differences follow from the aforementioned three axes and are not distinguished as categorizing user models [18].

(oli: Personalization) Individualization of user models, Adaptive/Adaptable User Interfaces, intelligent user interfaces [2], [4], [1]

[23]: The evaluation of a personalized system is problematic because it is unclear if the results gathered from a few individuals who all used system personalized for them can be generalized to entire population of users.

Including adaptive elements in the user interface might increase the workload of the user. When users had two alternative interfaces to a menu, an adapted one and a full one, there was cognitive overhead in 1) having to decide which features to include in the personalized interface and 2) having to figure out a menu item was missing and that changing to the full interface would make it appear in the menu again [2].

(Kun tiedetn kyttjst jotain... jatketaan Solutions-kohtaan)

Search systems have evolved from simple "query to results" paradigm to a clever system that utilizes the user model to provide the user with better search results. Instead of providing results based on solely the search word, the search system can extend the search beyond the search word and, for example, include some synonyms in the query. Or the search system might recognize a common typo in the search query and replace the misspelled word with the correct one. The system might ask the user to rate the result of the search and then change accordingly. (faceted search jonnekin) In the following subsections we cover three different solutions that support exploratory search: faceted search, relevance feedback and query expansion.

Faceted search

Information seekers often express a desire for a user interface that organizes search results into meaningful groups, in order to help make sense of the results and to help decide what to do next [7]. There two ways of grouping search results; clustering and hierarchical faceted categories. Clustering is grouping of items based on some similarity and is fully automated process. It is good for clarifying a vague query but the clustering algorithms aren't yet perfect and the clustering can be unpredicted [7]. Category system is a set of labels that are organized to mirror the domain. Hierarchical faceted categories are a set of hierarchical categories that each represent a different dimension. Categories are usually created manually but can be partly automated.

[12]:The writers had done a user experiment to find out what the searchers are looking at in a faceted search UI. The test participants were university students and the system of interest was a library system. As a result of the eyetracker test the writers found out that participants looked a lot at the facets and 47,4% of the eye movement was between facets, breadcrums summarising the selected facets and the result list. In an interview the participants told they used facets to help organize their view on the topic domain and select sub-topics for further investigation. Of these results the researchers deduced that the facets played an important role in the exploratory search process. The article summarizes related study on faceted search and exploratory search and has many interesting leads on articles for our essay topic.

Exploratory search is a complex information seeking task and to support this it has become accepted to use faceted search or categorized overviews [12]. Structured metadata is used to provide the user with an overview of the results and clickable categories. With this UI approach the user doesn't have to reformulate the query to narrow and browse the results. Faceted search is used in practice in library catalogs, web search, online shopping and other domains [12]. Faceted search enables the user to change fluidly between search and browsing and searchers with partially defined or changing information needs can use the overview to understand the knowledge domain and refine their needs. It has been shown that when using faceted search the users explored their results more broadly than without facets and felt more organized about their searches. Still though the faceted search interfaces make the search more efficient the subjects don't always prefer it [12].

[20]The writers have developed a client-side web search agent to work on top of google and they claim it improves the accuracy of the results. In information retrieval the user model/information need is traditionally (2005) constructed only by the query keywords and improved by relevance feedback where the user is asked to implicitly specify which of the results are relevant. As the users are usually reluctant in doing this, the writers have developed an implicit model for improving the search user model. Usually user models are long-lived but this model is concentrated on short-term context. The search agent tries to respond to a situation where a user would need different results for searching with keyword Java after searching programming stuff or after the user has been searching beaches in Indonesia. Two challenges for information retrieval systems are identified: 1. different users may use the exactly same keywords to look for different things and 2. the information needs of a user may change over time. In their search agent the writers gather information from two sources; the previous query keywords and the viewed document summaries. The system responds immediately to each user action with a system action and the user doesnt have to

actively take part on adapting the search result to be more relevant.

- System examples

Relevance feedback

(also User Modeling)

The goal of user modeling for a search system is to model a user's information need. The obvious source of information is the query. But as the queries tend to be quite short, especially when using a mobile device, the search words may not be a very accurate source of information for the user's information need. Much more data is available if user is given the opportunity to give feedback on the search results.

It is a way to teach the system about users preferences. The user uses the explicit communications channel to give the system some information about the context. In practise, when a search result is shown to the user the system provides a way for user to respond if the result was sufficient or not. The user might rate the result, for example with stars to point out how successful the system was in finding the results for the users query. However, relevance feedback requires a system that incorporates and utilizes the given feedback and of course, it requires the user to use time and energy to evaluate the search result and how the result fills the need of the information retrieval task at hand. The extra effort has been shown to be too great a burden to users and thus relevance feedback is considered a scarce source for user model data.

This *relevance feedback* has been found to improve retrieval accuracy [19]. This, however, requires extra effort from the users and users are reluctant to make extra effort [8]. As [20] shows, user action data can be used to improve the search results without the extra effort. They collected all the actions the user did and used them to update the user model. This user model was used in customizing the ranking which the results were based on. - System examples

An example of relevance feedback in practise is demonstrated in Figure 3. The search system at dell.com displays a form at the end of the result list. The user is thanked if the form is submitted and a promise ”We will use your feedback to improve search on dell.com” is displayed to the user. Choosing the ”Write and Tell Us More” link extends the form (Figure 4) to include a text field to further explain the feedback. Similar examples of relevance feedback have been reported by (viitteet).

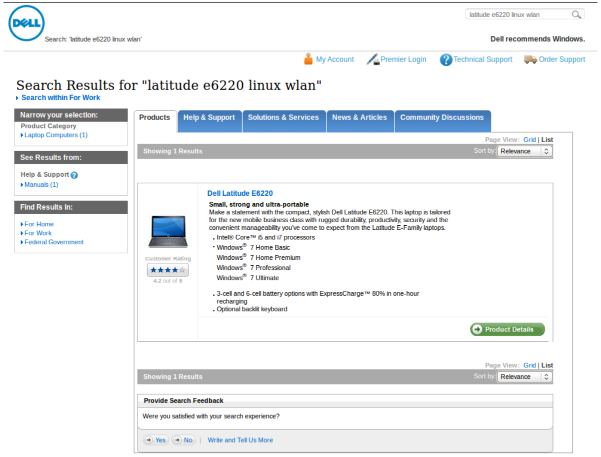
Query expansion

- System examples

Query Term Suggestion

Such search interfaces have appeared that suggest query terms dynamically, as the user enters them. In some of the systems the term suggestions appear before the searcher has seen any retrieval results, and in others, the system dynamically shows documents that match the characters typed so far, adjusting the results list as more characters are typed.

Figure 3. Relevance feedback



Search Results for "latitude e6220 linux wlan"

Dell Latitude E6220

Small, strong and ultra-portable

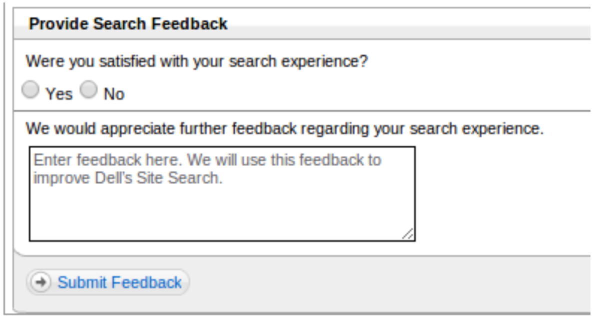
Make a statement with the compact, stylish Dell Latitude E6220. This laptop is tailored for the new mobile business class with rugged durability, productivity, security and the convenient manageability you've come to expect from the Latitude E-Family laptops.

- Intel® Core™ i5 and i7 processors
- Windows® 7 Home Basic
  Windows® 7 Home Premium
  Windows® 7 Professional
  Windows® 7 Ultimate
- 3-cell and 6-cell battery options with ExpressCharge™ 80% in one-hour recharging
- Optional backlit keyboard

Provide Search Feedback

Were you satisfied with your search experience?

Figure 4. Relevance feedback



Provide Search Feedback

Were you satisfied with your search experience?

○ Yes ○ No

We would appreciate further feedback regarding your search experience.

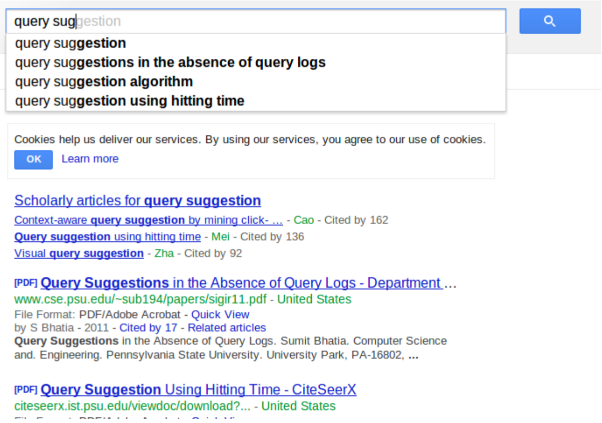
Enter feedback here. We will use this feedback to improve Dell's Site Search.

Dynamic query term suggestions (sometimes referred to as auto-suggest, autosuggest, or search-as-you-type) are becoming a widely used solution between requiring the user to know exactly how to spell the query terms and to think of relevant and connected terms overall. [Hearst]

Some dynamic term suggestion systems show only query suggestions whose prefix matches what has been typed so far. Figure 5 shows an example from Google's dynamic query suggestions interface, which shows queries where the prefix matches some words used in previously used queries. In this example, ”query sug” matches ”query suggestion”. Dynamic query suggestions are not restricted to matching the prefix of the query alone. In Figure 6 the system detects an upcoming spelling error in user's query term ”exansion” and suggests a query with a corrected term ”expansion”.

A less interactive system is displayed in Figure 7. The search system does not suggest any query terms and a user's misspelled query fails. However, as we see in Figure 8, only seconds after the initial use of the word ”morgage” the word is used again and spelled exactly the same, but the result page has changed. This helps user in a similar way as the interactive query suggestion method, but the first use of a word launches a background operation to handle future cases. The user model of the search system gets updated and after the initial use of the word ”morgage” the system will provide a suggestion in the otherwise empty result page (Figure 8).

Figure 5. One view of dynamic query term suggestion. An example from Google in which the prefix of a word in the query is matched against past queries. .


query suggestion

query suggestion
query suggestions in the absence of query logs
query suggestion algorithm
query suggestion using hitting time

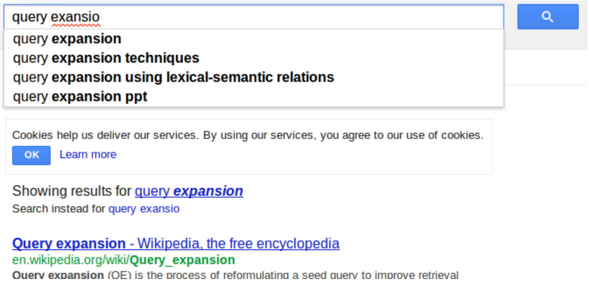
Cookies help us deliver our services. By using our services, you agree to our use of cookies.

Scholarly articles for query suggestion
Context-aware query suggestion by mining click- ... - Cao - Cited by 162
Query suggestion using hitting time - Mei - Cited by 136
Visual query suggestion - Zha - Cited by 92

[PDF] Query Suggestions in the Absence of Query Logs - Department ...
www.cse.psu.edu/~sub194/papers/sigir11.pdf - United States
File Format: PDF/Adobe Acrobat - Quick View
by S Bhatia - 2011 - Cited by 17 - Related articles
Query Suggestions in the Absence of Query Logs. Sumit Bhatia. Computer Science and. Engineering. Pennsylvania State University. University Park, PA-16802, ...

[PDF] Query Suggestion Using Hitting Time - CiteSeerX
citeseerx.ist.psu.edu/viewdoc/download?... - United States

Figure 6. Another view of dynamic query term suggestion in Google where the system detects an upcoming spelling error ”exansion” and suggests a query including a corrected word, ”expansion”.


query exansio

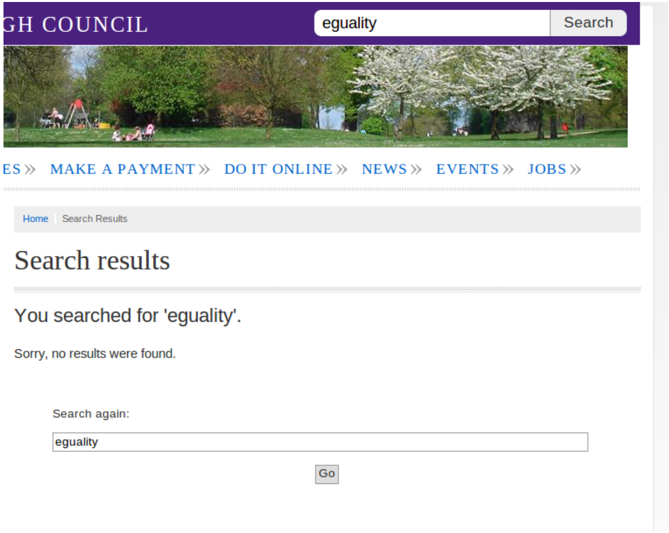
query expansion
query expansion techniques
query expansion using lexical-semantic relations
query expansion ppt

Cookies help us deliver our services. By using our services, you agree to our use of cookies.

Showing results for query *expansion*
Search instead for query exansio

Query expansion - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/Query_expansion
Query expansion (QE) is the process of reformulating a seed query to improve retrieval

Figure 7. Initial use of misspelled query term in Ipswich Borough Counsil website search system results in a page without suggestions.



Search results

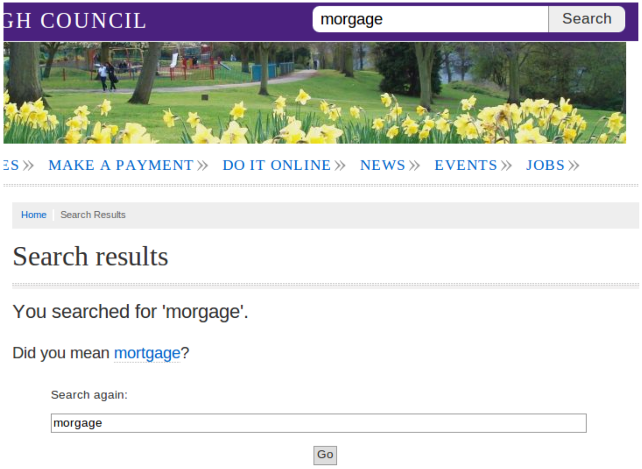
You searched for 'egualuty'.

Sorry, no results were found.

Search again:

egualuty

Figure 8. Only seconds after the initial use of misspelled ”morgage” query term in Ipswich Borough Counsil website search system, the system offers a suggestion of term ”mortgage”.



Search results

You searched for 'morgage'.

Did you mean mortgage?

Search again:

morgage

DISCUSSION

In the Discussion chapter we summarize our findings and point out what is missing and what we expect from future research.

Future challenges for user modeling : payoff, user modeling vs. task modeling, wrong/outdated info, adequacy of models, larger context, user’s loss of control, privacy concerns [5].

CONCLUSION

Goal, solution summary

Our goal was to explore the field of Exploratory Search and User Modeling. We found several articles that have some contribution to the topic.

Summary of results and their reliability

We found that: - Usage - Success - Failures

How much is it used in the real world, really?

Research impact - What has the research brought into software development?

REFERENCES

1. Brusilovsky, P. Methods and techniques of adaptive hypermedia. *User Modelling and User-Adapted Interaction 6*, 2-3 (1996), 87–129.
2. Bunt, A., Conati, C., and McGrenere, J. What role can adaptive support play in an adaptable system? In *Proceedings of the 9th international conference on Intelligent user interfaces*, ACM (2004), 117–124.
3. Dillon, A., and Watson, C. User analysis in hci the historical lessons from individual differences research. *International Journal of Human-Computer Studies 45*, 6 (12 1996), 619–637.
4. Findlater, L., and McGrenere, J. A comparison of static, adaptive, and adaptable menus. In *Proceedings of the*

SIGCHI conference on Human factors in computing systems, ACM (2004), 89–96.

5. Fischer, G. User modeling in human–computer interaction. *User modeling and user-adapted interaction 11*, 1-2 (2001), 65–86.
6. Hearst, M., Elliott, A., English, J., Sinha, R., Swearingen, K., and Yee, K.-P. Finding the flow in website search. *Communications of the ACM 45*, 9 (2002), 42–49.
7. Hearst, M. A. Clustering versus faceted categories for information exploration. *Communications of the ACM 49*, 4 (2006), 59–61.
8. Kelly, D., and Teevan, J. Implicit feedback for inferring user preference: A bibliography. *SIGIR Forum 37*, 2 (2003), 18–28.
9. Kobsa, A. Generic user modeling systems. vol. 11 (2001), 49–63.
10. Kobsa, A. *Generic user modeling systems*, vol. 4321 LNCS of *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2007.
11. Kuhlthau, C. C. Inside the search process: Information seeking from the user's perspective. *JASIS 42*, 5 (1991), 361–371.
12. Kules, B., Capra, R., Banta, M., and Sierra, T. What do exploratory searchers look at in a faceted search interface? In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries* (2009), 313–322.
13. Kules, B., and Shneiderman, B. Users can change their web search tactics: Design guidelines for categorized overviews. *Information Processing and Management 44*, 2 (2008), 463–484.
14. Marchionini, G. Exploratory search: From finding to understanding. vol. 49 (2006), 41–46.
15. O'Connor, B., Krieger, M., and Ahn, D. Tweetmotif: Exploratory search and topic summarization for twitter. *Proceedings of ICWSM* (2010), 2–3.
16. Pazzani, M., and Billsus, D. Learning and revising user profiles: The identification of interesting web sites. *Machine Learning 27*, 3 (1997), 313–331.
17. Pu, H. ., Chuang, S. ., and Yang, C. Subject categorization of query terms for exploring web users' search interests. *Journal of the American Society for Information Science and Technology 53*, 8 (2002), 617–630.
18. Rich, E. Users are individuals: individualizing user models. *International Journal of Human-Computer Studies 51*, 2 (8 1999), 323–338.
19. Salton, G., and Buckley, C. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science 41*, 4 (1990), 288–297.
20. Shen, X., Tan, B., and Zhai, C. Implicit user modeling for personalized search (2005). 824–831.
21. Sugiyama, K., Hatano, K., and Yoshikawa, M. Adaptive web search based on user profile constructed without any effort from users (2004). 675–684.
22. Tvarožek, M. Exploratory search in the adaptive social semantic web. *Information Sciences and Technologies Bulletin of the ACM Slovakia 3*, 1 (2011), 42–51.
23. Van Velsen, L., Van Der Geest, T., Klaassen, R., and Steehouder, M. User-centered evaluation of adaptive and adaptable systems: a literature review. *Knowledge Engineering Review 23*, 3 (2008), 261.
24. Wei, B., Liu, J., Zheng, Q., Zhang, W., Fu, X., and Feng, B. A survey of faceted search. *Journal of Web Engineering 12*, 1-2 (2013), 041–064.
25. White, R. W., Bennett, P. N., and Dumais, S. T. Predicting short-term interests using activity-based search context. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, ACM (2010), 1009–1018.
26. White, R. W., Drucker, S. M., Marchionini, G., Hearst, M., and Schraefel, M. C. Exploratory search and hci: Designing and evaluating interfaces to support exploratory search interaction. In *Conference on Human Factors in Computing Systems - Proceedings* (2007), 2877–2880.
27. White, R. W., Kules, B., and Drucker, S. M. Supporting exploratory search, introduction, special issue, communications of the acm. *Communications of the ACM 49*, 4 (2006), 36–39.
28. White, R. W., Marchionini, G., and Muresan, G. Evaluating exploratory search systems. introduction to special topic issue of information processing and management. *Information Processing and Management 44*, 2 (2008), 433–436.
29. White, R. W., and Roth, R. A. Exploratory search: Beyond the query-response paradigm. *Synthesis Lectures on Information Concepts, Retrieval, and Services 1*, 1 (2009), 1–98.
30. Wilson, M. L., Kules, B., Schraefel, M. C., and Shneiderman, B. From keyword search to exploration: Designing future search interfaces for the web. *Foundations and Trends in Web Science 2*, 1 (2010), 1–97.