

Supporting Exploratory Search with User Modeling

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

In this literature review we introduce exploratory search and cover three techniques that help users in succeeding in their exploratory search tasks efficiently. Exploratory search is an information retrieval strategy that combines learning and investigation activities. We explain the concept of exploratory search and what distinguishes it from other information retrieval strategies. Exploratory searchers can be supported in their tasks by means of user modeling. After shortly summarizing user modeling we present three practical techniques that combine user modeling with exploratory search. *Faceted search* is a method where structured metadata is used to provide the user with an overview of the results and clickable categories. *Relevance feedback* provides the user an option to rate the usefulness of the result. This information is used in ranking the search results next time. *Query term suggestion* displays choosable query suggestions as the user is typing their query.

Author Keywords

Exploratory Search; Information Retrieval; User Modeling. Faceted Search; Relevance Feedback; Query Term Suggestion;

ACM Classification Keywords

H.5.5. Information Interfaces and Presentation (e.g. HCI): Modeling

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 [10]. User model is built of these characteristics and traditionally it has been a model of a typical user [10].

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.

Information retrieval tasks the users do are different, and it has been shown that there are different types of searching []. We dive into the exploratory search which has its properties that distinguish it from other searching types. We adopt the description of exploratory search by Marchionini:

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 [9].

An exploratory search task is not a well-defined task that the user stops after finding an answer to a question. Instead, as the searcher is not after a specific fact, the search process is iterative and the user might return to the task several times and in several sessions. The time gap between the sessions means the user is easily distracted or lead to different paths, since they have lost the 'flow' they had in the previous search session []. Having lost the flow, their (motivation is weak, easily distracted, open to new ideas) and the overall search goal may get redefined. The nature of an exploratory search is (hard-to-define). Supporting such a process is challenging, because the tactics and preferences vary. Although the evaluation of exploratory search user interfaces has been proven to be difficult [], some design guidelines have been proposed and some solutions have been found to help users in their exploratory search tasks.

To this (literature review/lecture/essay) we chose three solutions that are seen in search systems in general, but these solutions can be enhanced further by (taking advantage of) user modeling. The chosen solutions were (chosen), because 1) they take place in different stages of the search process, 2) their use of user modeling differs, and 3) the reward the user (gets is different). While presenting the solutions, these (differences) are explained. In addition, we describe how an exploratory search process benefits from use of user modeling in these three solutions.

(move this?) [14]: 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) [10], [4]

Exploratory Search

(Ilkka)

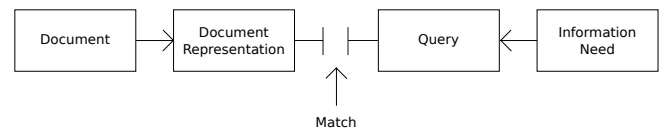
The search for information and (growing one's knowledge) is a built-in (property) in humans. Last few decades have transformed search and made it an everyday function. It wasn't until the late 1990's until web search services began offering quick access to information and traditional (thinking) about information retrieval had to be updated. Before the Web, information search services were used mainly by academics and information professionals. Small and motivated user base (prevented) search system manufacturers to focus on the user experience and the motivation to improve the user interface was (small). Traditional search systems were built on (query-results thinking) and as such, the query formulation was an essential stage of the search task [1]. Some studies have even evaluated systems, where the query results take days to return [1]. In such cases, the (user learns the system very slowly), since becoming a better user of such a system requires interaction with system and slow (query return) means slow learning.

Search system must be quick for other reasons too. Cognitive abilities restrict the searcher from running other tasks and starting sub-tasks in parallel with the original task [1]. That means the system should be simple and support user's (state of mind), not adding to the cognitive load already affecting the user. Whenever the user needs to (think on a higher level) it's possible the flow' of the search task is lost. This so-called flow is a state of mind that the searcher has and having to set that aside might result in losing it.

The flow however, is not so important when the search task is simple. Let's think of a simple search task: a geography teacher has assigned a group of students to draw a poster about a European country and the group have chosen Denmark. They want to display some facts about the country in their poster, for example the population, land area and internet top-level domain. To retrieve these facts, a simple search with keyword Denmark will suffice. The facts are copied from Wikipedia or the official website of Denmark to the poster. This kind of lookup-based information retrieval model is described in Figure 1. After finishing the poster, the group is given a new assignment. Now their assignment is to write a play of a historical event that was of special importance for Denmark and then perform the play for the class. This time,

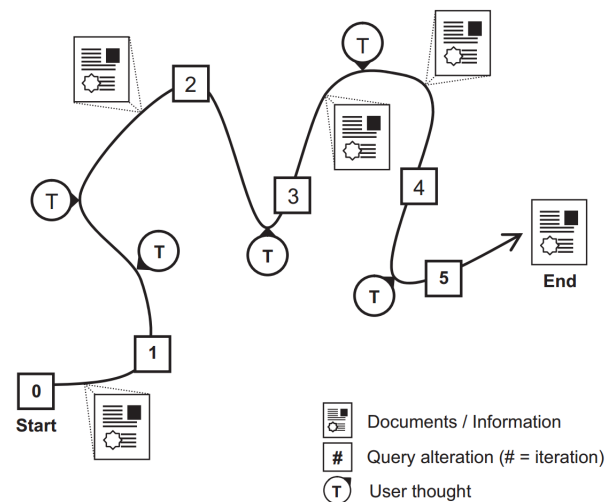
the search task demands more time and is not as (easily definable) as in the previous assignment. The students might start by reading the history of Denmark in the Wikipedia. Along their exploration, they come upon different events and characters that have had an effect in Denmark's history. With their now improved knowledge of Danish history, they need to decide which event they are to (display) in their play. This requires collaboration within the group. After decision on the event, the group needs to gather information on the people involved in the event. And if possible, they investigate what kind of clothes were used in the era the event took place in. The tasks described (fit well in the strategy of) berry-picking. See Figure 2.

Figure 1. The classic information retrieval model. Based on [?].



This text is between figures in .tex file.

Figure 2. . Based on [?]. From [9].



This text is between figures in .tex file.

Introduction to exploratory search. See Figure 3. [9], [15], [13]

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. [8]

Exploratory search and iterative search differ. See Figure 4.

Exploratory search tasks can be characterized as either learning oriented or investigative and they have common aspects

Figure 3. Different search tasks are split into three overlapping search activities: *lookup*, *learn* and *investigate*. Based on [9].

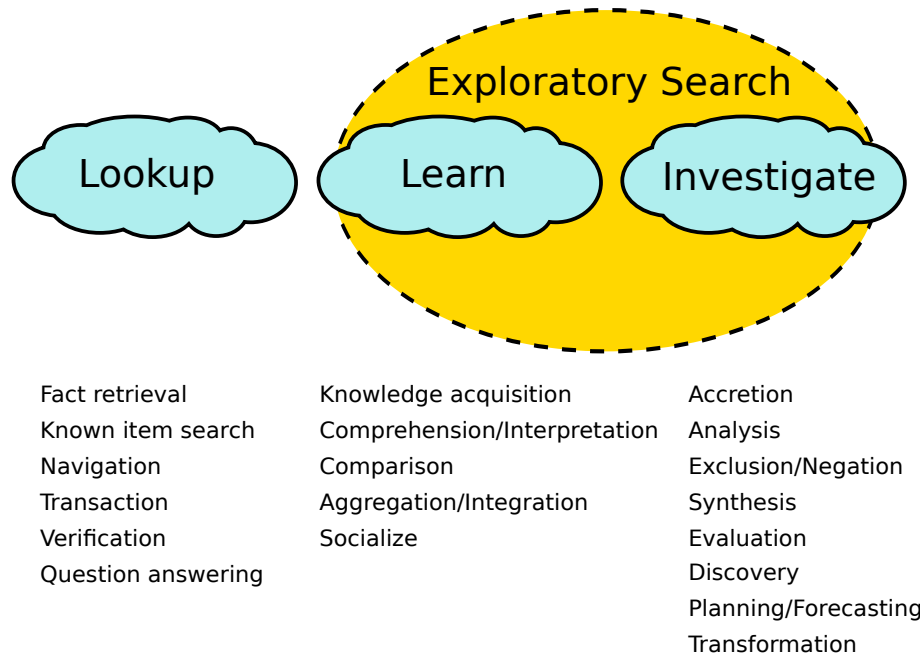
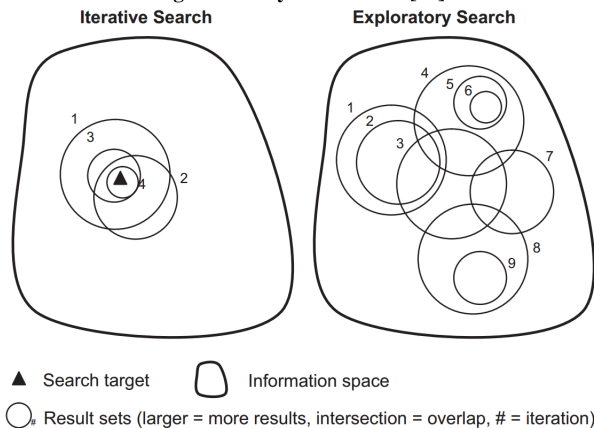


Figure 4. They differ. From [15].



like uncertainty, ambiguity and discovery distinguishing them from look-up oriented tasks [7].

User Modeling (Tuire)

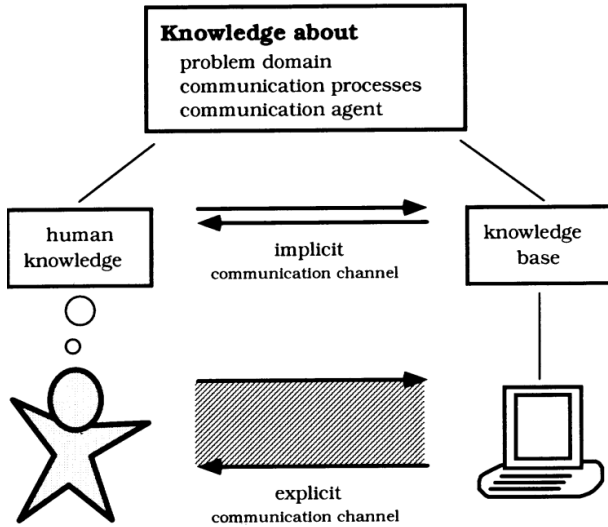
In the first ten years of Human-Computer interaction history the focus of research was in user interfaces and particularly the possibilities and design criteria of graphical user interfaces [4]. Later the focus has moved toward more broadly improving the way people use computers for example to work, think, communicate and learn [4]. 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 [10]. 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" [4].

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 [4]. 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 5).

In order to accommodate to differing needs of users or user-groups over time, a system may use one of three basic approaches [14]. System is called *adaptive* if it alters its 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 [14]. 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

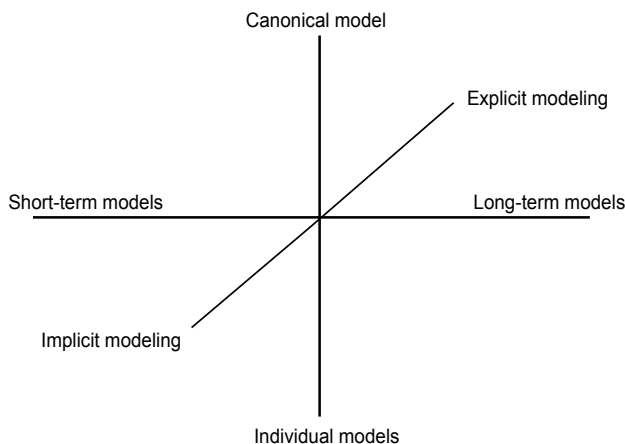
Figure 5. Knowledge-based HCI [4].



user modeling and the use time becomes design time [4].

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 [10]. The user model classification map is depicted in Figure 6. and the three dichotomies are explained further in this chapter.

Figure 6. The dimensions of user modeling.



The *single model vs. collection of models* polarity [10] 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 [10]. 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 [10]. 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 heterogeneous for it to be useful enough [10]. 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* [10]. 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* [10]. 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 [10].

(oli: Personalization) Individualization of user models, Adaptive/Adaptable User Interfaces, intelligent user interfaces [2], [3], [1]

[14]: 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 [5]. 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 unpredictable [5]. 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.

[7]: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, breadcrumbs 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 [7]. 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 [7]. 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 [7].

[12]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 doesn't 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 user's 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 user's 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 [11]. This, however, requires extra effort from the users and users are reluctant to make extra effort [6]. As [12] 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 7. 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 8) to include a text field to further explain the feedback. Similar examples of relevance feedback have been reported by (viitteet).

Figure 7. Relevance feedback

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 8. Relevance feedback

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]

Figure 9. 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. .

Some dynamic term suggestion systems show only query suggestions whose prefix matches what has been typed so far. Figure 9 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 10 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 11. The search system does not suggest any query terms and a user's misspelled query fails. However, as we see on the right part of the figure, just 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 11. Initial use of misspelled query term in Ipswich Borough Council website search system results in a page without suggestions (left). Just seconds following the initial use of misspelled "morgage" query term, we resubmit the query as before and as we see on the right, the system offers "mortgage" as a suggestion.

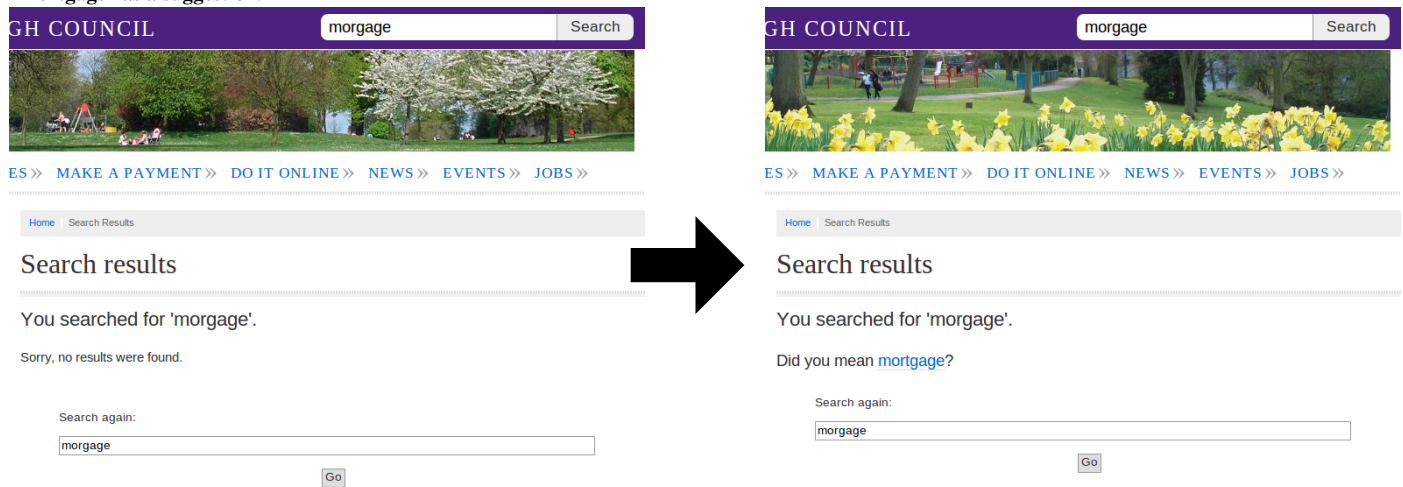
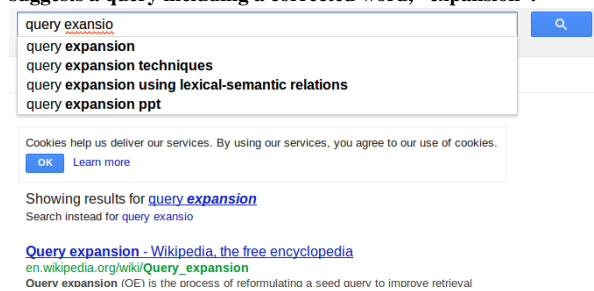


Figure 10. Another view of dynamic query term suggestion in Google where the system detects an upcoming spelling error "exansio" and suggests a query including a corrected word, "expansion".



DISCUSSION

In the modern day people use online search applications widely. Search applications are actually the second most frequently used online applications. Online environment seems to be perfect for both searching quickly for facts and doing more sophisticated searching with goals like investigating, forming overall understanding of a given subject or trying to learn about previously unknown topics. People's online lives may include anything from finding the perfect companion or watching Psy's next huge viral hit video to learning the basics of physics. Built on the idea of hyperlinking, the internet is practically made for exploring. Exploratory search tasks have become part of everyday life, maybe because the internet has applications that support it better than the traditional information sources like libraries, books and human-to-human conversations.

Exploratory searchers can be supported in their tasks by means of user modeling, the goal being to accurately model the users' information needs. This task is unfortunately a very difficult one. Indeed, it is hard for users themselves to precisely describe what their information need eventually even is.

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. The solutions that aid exploratory search usually include methods to recommend alternative search paths or methods to suggest sources of related information.

Faceted search has been found to be a useful means of supporting exploratory search. It supports the refining of the query and enables browsing through the formed categories thus enriching the understanding of the knowledge domain. Additionally, the faceted search helps the users to refine or re-evaluate their information needs. Still, even though faceted search interfaces make the search more efficient and the users explore the results more broadly than without facets, they don't always prefer it.

Another supporting technique that we came upon was query term suggestion. It saves the user from typing as the system suggests combinations of query terms based on the characters or words the user has already typed in the search form. A study suggests that as much as 15 percent of search queries return no results because of misspelled query terms. By clicking a suggested search query, the search gets underway with correctly spelled words and with a certainty that some results will be returned, since the suggested query has successfully been used before.

Most search systems rely only on the query the user has submitted when constructing a user model, but more direct ways to gather information exist, too. Asking user to rate the result based on its usefulness provides the system with direct and precise evidence of the relevance of the search result. Relevance feedback has been found effective in enhancing search accuracy, but this method requires a system to analyse given feedback and apply it into the ranking of the search results.

In practise, the extra effort the user needs to make seems to be too much for the users.

In our opinion using user models in supporting exploratory search has some payoffs. The user might feel insecure if the application forms search results for reasons the user does not understand, depriving the user's control of the system. User modeling also has privacy implications. Who stores the information that has been gathered from the user's interactions? How securely it is stored and who controls the purposes it is used for? For example, Google collects information about the user's actions and uses them to construct user model for marketing purposes. The privacy of using the data is questionable and they provide no means of preventing them from collecting one's interaction data.

The methods we encountered during our literature review were familiar to us as major search engines have incorporated them in their user interfaces, but before writing this paper we didn't know their names, let alone their theoretical background. It is obvious to us that the solutions described in our paper are useful to many exploratory searchers although not all of the implementations we tried were adequately intuitive or mature enough in their visualization.

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. 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.
4. Fischer, G. User modeling in human–computer interaction. *User modeling and user-adapted interaction* 11, 1-2 (2001), 65–86.
5. Hearst, M. A. Clustering versus faceted categories for information exploration. *Communications of the ACM* 49, 4 (2006), 59–61.
6. Kelly, D., and Teevan, J. Implicit feedback for inferring user preference: A bibliography. *SIGIR Forum* 37, 2 (2003), 18–28.
7. 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.
8. 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.
9. Marchionini, G. Exploratory search: From finding to understanding. vol. 49 (2006), 41–46.
10. Rich, E. Users are individuals: individualizing user models. *International Journal of Human-Computer Studies* 51, 2 (8 1999), 323–338.
11. Salton, G., and Buckley, C. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science* 41, 4 (1990), 288–297.
12. Shen, X., Tan, B., and Zhai, C. Implicit user modeling for personalized search (2005). 824–831.
13. Tvarožek, M. Exploratory search in the adaptive social semantic web. *Information Sciences and Technologies Bulletin of the ACM Slovakia* 3, 1 (2011), 42–51.
14. 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.
15. 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.