

Mika Käki

Enhancing Web Search Result Access with Automatic Categorization

ACADEMIC DISSERTATION

To be presented with the permission of the Faculty of Information Sciences of the
University of Tampere, for public discussion in the Pinni auditorium B1097 on
December 2nd, 2005, at 13:00 o'clock.

Department of Computer Sciences
University of Tampere

Dissertations in Interactive Technology, Number 2
Tampere 2005

ACADEMIC DISSERTATION IN INTERACTIVE TECHNOLOGY

Supervisor: Professor Kari-Jouko Rähkä, PhD,
Department of Computer Sciences,
University of Tampere,
Finland

Opponent: Dr. Polle T. Zellweger
Bellavue, Washington,
United States of America

Reviewers: Dr. Steve Jones
Department of Computer Science,
University of Waikato,
New Zealand

Professor Samuel Kaski, PhD,
Laboratory of Computer and Information Science,
Helsinki University of Technology,
Finland

Electronic dissertation
Acta Electronica Universitatis Tamperensis 496
ISBN 951-44-6490-7
ISSN 1456-954X
<http://acta.uta.fi>

Dissertations in Interactive Technology, Number 2

Department of Computer Sciences
FIN-33014 University of Tampere
FINLAND

ISBN 951-44-6450-8
ISSN 1795-9489

Tampereen yliopistopaino Oy
Tampere 2005

Abstract

Information in the Web is typically found with the help of a Web search engine. For instance, Google has been reported to index over eight billion Web pages and to process over 200 million queries a day. Information is available, but users express their information need with very few query words, typically with one or two. The task of finding relevant information from a set of 8 billion documents with a cue of just two words is a tremendous challenge. Search engines perform incredibly well with sophisticated result ranking methods, but there are cases when the result ranking is not appropriate. For example, undirected informational searches where a broad understanding about a topic is sought or queries with ambiguous terms are such cases.

Our approach is to enhance users' result access process with automatically formed filtering categories. Categories provide an understandable overview of the results and make accessing of relevant results easy. The concept is implemented in a search user interface called Findex. Two different categorization methods have been developed promoting simplicity to make the functionality understandable to the users.

We evaluated our approach in controlled experiments, in a longitudinal study, and with a theoretical test. In the experiments we tested the usefulness of the proposed user interface and the categorization schemes with 20 and 36 participants. The results showed that finding relevant results is about 30–40% faster with the proposed user interface compared to the *de facto* standard, the ranked results user interface. The attitudes favor the new user interface. In an experiment with 27 participants we found that it is better to show only a small number of categories (around 10–15) instead of maximizing the result coverage by displaying more categories.

The results of the experiments were complemented with a longitudinal (two months) study in real use situation with 16 participants. The results indicated that the categorization user interface becomes a part of the users' search habits and is beneficial. However, the benefit is not as clear as the experiments indicated. In a real situation, the categories are needed and used in about every fourth search. The usage patterns indicate that categories help when result ranking does not bring relevant results to the top of the result list.

Acknowledgements

The most important support for my work has become from my research group. My closest colleague (literally, sitting in the same room), Anne Aula, had a major role in this work producing ideas and contributing to the design of the system. Without her, the system would not be what it is today. She also had a major role in carrying out pilot tests when we explored the experimental settings for the studies as well as in everyday decisions concerning the research. Working together made the whole process a lot of fun – I cannot image a better way of making a PhD.

In addition to Anne, other members of the research group have been of great importance in this work. Tomi Heimonen, Harri Siirtola and Natalie Jhaveri have all commented, discussed and helped me with problems, writing papers and evaluating ideas. Their role in the final product is considerable and I am grateful for the collaboration. Johanna Höysniemi is also worth mentioning although not an official member of the group.

Dr. Scott MacKenzie's course on Methods, Models, and Measures was revolutionary for my research. The sound methodology presented and description of the research process sharpened my understanding of the matter to an extent that made the implementation of the studies clear and strict. In fact, these insights shaped the form of the whole research.

Kari-Jouko Rähkä has been the enabling factor in the whole process. Experience from short term projects gives me a perspective according to which this work would not have been possible without proper funding from the graduate school (UCIT). It enabled the persistent and systematic work towards the goal. Kari-Jouko has been a founding member of the graduate school thus essentially making this all possible.

Stina Boedeker has provided much inspiration and confidence for my work. Poika Isokoski's attitude and dedication to the research has provided me with a model that definitely had an effect that was needed for accomplishing the thesis.

I would also like to thank Mari, who has pushed me to try harder on our discussions about the research and ways of doing it. She had also a key role in starting my final writing process.

Tampere, 1st of November, 2005

Mika Käki

Contents

1	INTRODUCTION	1
1.1	Objective.....	1
1.2	Context.....	2
1.3	Previous Work.....	3
1.4	Method.....	4
1.5	Results.....	4
1.6	Structure of the Thesis.....	5
2	ACCESSING WEB INFORMATION.....	7
2.1	Web Searching.....	7
2.2	Search Process.....	8
2.3	Information Foraging Theory.....	11
2.4	Web Search Engine User Interfaces.....	12
3	ENHANCING SEARCH RESULT ACCESS	17
3.1	Overview.....	17
3.2	Keyword-in-context index.....	18
3.3	Visualizing Search Results.....	19
3.4	Query Refinements.....	23
3.5	Categorizing Web Search User Interfaces.....	25
4	ENHANCING SEARCH RESULT ACCESS WITH CATEGORIZATION	29
4.1	Cluster Hypothesis.....	29
4.2	Categorization Techniques.....	30
4.3	Central Search Result Categorization Systems.....	36
4.4	Related Clustering Systems.....	40
4.5	The Findex System.....	42
5	METHODOLOGY	47
5.1	Constructive Approach.....	47
5.2	Measuring the Use of Search Interfaces.....	47
5.3	Contributed Measures.....	49
5.4	Experimental Design.....	50
5.5	Tasks.....	51
6	STUDIES.....	53
6.1	Overview of the Studies.....	53
6.2	Study I: Experiment of Statistical Categories.....	54
6.3	Study II: Search User Interface Evaluation Measures.....	55
6.4	Study III: The Effect of the Number of Categories.....	57
6.5	Study IV: Longitudinal Study of Findex.....	58
6.6	Study V: Experiment with Context Categories.....	60
6.7	Study VI: Evaluation of the Categorization Algorithms.....	61
6.8	Division of Labor.....	62
7	CONCLUSIONS	65
8	REFERENCES	69
	APPENDIX 1	81

List of publications

This thesis consists of a summary and the following original publications, reproduced here by permission of the publishers.

- I Mika Käki and Anne Aula (2005). Findex: improving search result use through automatic filtering categories. *Interacting with Computers*. Elsevier, Volume 17, Issue 2, pages 187–206. 85

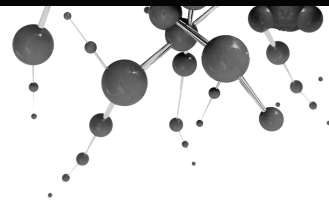
- II Mika Käki (2004). Proportional search interface usability measures. In *Proceedings of NordiCHI 2004 (Tampere, Finland)*, 23–27 October 2004. ACM Press, pages 365–372. 107

- III Mika Käki (2005). Optimizing the number of search result categories. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2005 (Portland, USA)*, April 2005. ACM Press, pages 1517–1520. 117

- IV Mika Käki (2005). Findex: search result categories help users when document ranking fails. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2005 (Portland, USA)*, April 2005. ACM Press, pages 131–140. 123

- V Mika Käki (forthcoming). fKWIC: frequency based keyword-in-context index for filtering web search results. Accepted for publication in *Journal of the American Society for Information Science and Technology*. Wiley. 135

- VI Mika Käki (2005). Findex: properties of two web search result categorizing algorithms. Accepted for publication in *Proceedings of the IADIS International Conference on World Wide Web/Internet (Lisbon, Portugal)*, October 2005. IADIS Press, pages 93–100. 153



1 Introduction

1.1 OBJECTIVE

The motivation for this study is to enhance the end users' opportunities to find meaningful results from the Web search engine results. We intend to achieve this by automatically categorizing the search results and by presenting an overview of the results to the user. Two alternative systems have been implemented and discovered to enhance the users' ability to access search results.

Web searching is a ubiquitous, basic way of finding information from the ever expanding World Wide Web (WWW). Jakob Nielsen (2004) has stated that 88% of the navigation sessions are initiated by the use of a search engine. Google (www.google.com) is currently the most popular search engine indexing over 8 billion Web documents and handling over 200 million queries a day (Google Timeline, 2005). The user interface of Google, and of other popular search engines, still resembles the solutions first introduced in the beginning of the 1990s when the Web was young and much smaller (e.g., first release of Lycos in 1994 indexed about 54 000 documents (Mauldin, 1997)).

Although the size of the Web and Web search engine databases is rapidly increasing, the skills of the users have not altered all that much. Extensive studies of the Web searchers' behavior with search engines show that the topics of the searches have changed with the evolution of the Web, but the query formulation skills of the users have not (Spink et al., 2002; Jansen & Spink, 2006). In particular, users routinely submit short queries containing just a few words (on average about 2.5 words).

When we combine these two facts, the motivation for our study becomes evident. Although the search engines use sophisticated techniques for

ranking the search results, it is virtually impossible to return the most relevant document to the users out of 8 billion if the cue consists only of two words. This is especially true in situations where the query words are ambiguous or when the user wants to learn different sides of a certain topic.

Inspired by this, we ask the following research question: **can we enhance the users' search performance by new user interface solutions?** If so, **how can they be achieved and how important are such advances?**

In accordance with our research question, we designed user interface prototypes based on the idea of categorizing search results. The solutions were tested mostly in human-computer interaction (HCI) driven studies.

1.2 CONTEXT

In addition to the obvious context of Web searching and current Web search engine user interfaces, this research is connected to various research fields, discussed briefly below.

The primary field of research relevant to the current thesis is human-computer interaction. Human-computer interaction emphasizes the end user's role in the success of a system. Methods from HCI research can be applied in many domains, but software user interfaces were the original focus. In HCI, a solution is considered successful if we can observe measurable improvements in the end user's performance with the system in a particular task. This study is strongly HCI driven and aims to contribute to the knowledge in HCI.

The second important field is information retrieval (IR) studies or, more broadly, information studies. The roots of IR are in the early days of storing textual data in computer systems. Textual data, as opposed to structured data in databases, poses particular challenges in retrieving the information from the storage. Exact matching is not desirable in the same sense as with structured data. It would be frustrating trying to find a book if one had to type in the title in the exactly correct form. The example illustrates the different user needs that are associated with IR systems. The main results from IR studies are in the core of every (Web) search engine today. As the availability and importance of electronically stored information has increased, the information retrieval community has focused on a wider context of using information. For example, information seeking (IS) considers the wider context of information use including the retrieval of the information from databases. This study resembles many IR and IS studies and can provide a contribution for these communities.

The third partially related field is data mining or knowledge discovery. This is a field that studies ways of automatically extracting useful

information from large databases. In our view, the size of the database is related to the user's task and resources. If the utilization of information is difficult or too time consuming for the user in a given task, we consider the database large. This implies that we can utilize automatic knowledge discovery or summarization methods to help the user in understanding the information. Data mining and knowledge discovery are fields where techniques similar to our automatic categorization are developed and studied. However, we do not aim to contribute to data mining technologies in our study.

Fourthly, natural language processing (NLP) is a field of research that we use in our work. NLP can be regarded as a set of computing techniques that aim, in extreme, to achieve human-like language understanding with computer software. Such techniques include, for example, word stemming and part-of-speech analysis. When we compute categories for textual data, it is common to utilize some techniques used or developed in the NLP field.

1.3 PREVIOUS WORK

An early pioneer of automatic overviews for accessing information was the SuperBook prototype (Remde et al., 1987). It was among the first systems where the (meta)data contained in the document was used to automatically create a meaningful overview of it. Although the text was nicely structured (it had headings and sub-headings clearly marked), the idea of automatically producing an overview of the text was important, especially since it was found to be beneficial in a subsequent user study (Egan et al., 1989).

Scatter/Gather (Cutting et al., 1992; Cutting et al., 1993), developed in Xerox PARC, took the idea a step further. The tool enabled browsing a large document collection without explicit search functionality. The solution was based on an organization created by automatic clustering. One of the contributions was a new clustering technique that made it feasible to handle substantial numbers of documents, which had not been possible previously. Later on, Scatter/Gather was used for accessing search results with the same clustering idea (Hearst et al., 1995; Hearst & Pedersen, 1996).

Another pioneering system was presented by Allen, Obry, and Littman (1993), who focused on introducing structure to the result documents of a search. The user interface was built around an interactive dendrogram (a type of tree structure) built by a hierarchical clustering algorithm. The user could select branches from the tree and see the corresponding article titles (search results) in the user interface.

One of the most influential systems in organizing Web search results is Grouper (Zamir & Etzioni, 1999). Grouper uses a clustering algorithm specifically built for organizing Web search results. The system was successful and it is extensively cited in the literature. Our approach is close to Grouper, but the categorization algorithm and the user interface are different. In addition, we evaluated our solution in laboratory settings. This gives us more information about the use of such systems.

Whereas the above-mentioned systems use clustering techniques where similar documents are brought together to form groups, there are also systems that use classification methods. DynaCat (Pratt & Fagan, 2000) applied Medical Subject Headings (MeSH) classification in retrieving medical documents. Chen and Dumais (2000) used a similar technique in the classification of Web search results. Their evaluation method is partly used in our studies.

1.4 METHOD

This work is based on constructive and empirical research. We have constructed a software program that implements a user interface with automatic categorization facilities. Two different categorization techniques are available through the same user interface.

The evaluation part is based on 1) laboratory experiments and 2) a longitudinal study. We conducted three controlled experiments with 20–36 participants for testing the effects of the proposed user interface on the user performance. The controlled environment enabled us to measure accurately the users' interaction with the system.

A longitudinal study was used to compensate the limitations of the controlled experiments. The usefulness of the system in real situation cannot be fully understood by only relying on the laboratory tests. Thus, information on real use was collected in a longitudinal study with 16 participants over a period of two months.

Mathematical measures were also employed to characterize the properties of the categorization algorithms. Specifically, the last study followed the example of many information retrieval studies on systems similar to ours. The system was empirically tested, but without human participants.

1.5 RESULTS

The research produced five main results that address different sides of our research questions and confirm findings in previous research:

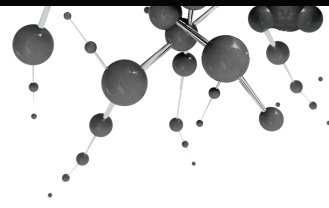
1. Automatic categorization can be used to enhance user performance in search tasks. We describe two methods for categorizing Web search results and a filtering user interface concept for enhancing the users' task of evaluating the search results.
2. User performance is significantly improved by our techniques. The benefit is about 30–40% faster speed in finding relevant answers while the number of irrelevant answers is reduced. In addition to these, we found evidence that users prefer the suggested search user interfaces. These results are based on observations in laboratory studies conducted with both categorization systems.
3. New user interface techniques can have an impact on users' ways of searching. In the longitudinal study, users adopted the new categorizing technology as a part of their search habits and they reported benefiting from it. They also reported having changed the way they formulate search queries. Log files showed that the use of categories stayed at a constant level being used in roughly every fourth search.
4. The two presented categorization algorithms work acceptably and complement each other. In a theoretical test without user participation one method produced higher coverage and overlap results whereas the other produced higher quality category names. The computational performance of the algorithms was on an acceptable level although they are not optimized for top performance.
5. The cluster hypothesis is confirmed. Jardine and van Rijsbergen (1971) stated in their so-called *cluster hypothesis* that relevant documents for a query tend to be similar. According to our results, this seems to hold in the context of clustering Web search results.

1.6 STRUCTURE OF THE THESIS

This thesis consists of a summary and six original articles published in international conferences and journals. The summary will first introduce the reader to the phenomenon of finding information in the Web. This is followed by discussion and a review of the related work on helping users find relevant information in the Web environment. As we have seen various methods for enhancing the access to Web search results, we will focus on methods that are based on result categorization. Our approach is based on categorization and thus this chapter contains the closest references. At the end of the chapter, we introduce our own Findex tool.

Whereas the beginning of the thesis mostly concerns the related work, the latter part of the summary (from the methodology chapter on) describes our approach. The methodology chapter discusses our research approach

and the options we had in selecting the evaluation methods. In the 'Studies' chapter we connect each of the separate publications to the context of this thesis and explain their meaning for it. The thesis ends with conclusions drawn from the results. But before going to the conclusions, let us first begin with the basic information about searching information from the Web.



2 Accessing Web Information

2.1 WEB SEARCHING

The World Wide Web has become a widespread and extensive source of information. The actual number of documents available on the Web is impossible to count due to the distributed nature of the Web, but Google reported indexing over 8 billion pages by the end of 2004 (Google Timeline, 2005). Clearly the amount of information is well beyond the comprehension of any information user.

Because the number of Web documents and Web sites is so extensive, the use of Web search engines is one of the basic activities while using the World Wide Web. Almost 90% of Web navigation sessions start with the aid of a search engine (Nielsen, 2004). Web search engines have put information retrieval systems into the every day lives of millions of people.

However, Web searching differs from the use of conventional information retrieval systems. The most prevalent difference is the user population. According to a study by Jansen and Pooch (2000), the users of traditional information retrieval systems enter fairly sophisticated and long search queries. Such systems are typically used by professionals (like librarians) who are formally educated and who understand Boolean logic. In contrast, with OPACs (online public access catalogs) and Web search engines, the use of advanced query features or comprehensive terminology is rare. Users make short queries, typically consisting only of one or two words.

Both of these latter systems (OPACs and Web search engines) are used by laypersons with varying backgrounds. The users are not experts in information retrieval and may not even know much about the topic they are exploring. Formal categorizations or classification terms are not known to such users and thus, the selection of the query words may be

suboptimal (especially with OPACs). With the Web search engines, the user population is even more varied than with OPAC systems. Practically every Web user is a search engine user and the variety in the skills, background knowledge, and interests is enormous.

Spink, Jansen, and their colleagues (Jansen et al., 1998; Jansen et al., 2000; Spink et al., 2001; Spink et al., 2002; Jansen & Spink, 2006) have conducted comprehensive studies about the use of Web search engines by analyzing the query logs of the Lycos Web search engine. The data collections are impressive, covering over one million search sessions in multiple samples over the years. The data covers the search behavior of about 200 000 searchers.

The main results of the log analysis are clear: Web users formulate short queries consisting only of a couple of terms, typically one or two. Advanced operators, Boolean operators in particular, are seldom used (on average in less than 10% of the queries). Among the operators, phrase search is the most popular. The topics that are being searched have changed over time, but the query formulation skills and habits remain the same. This is notable, as at the same time the amount of available information has exploded.

The user's next task after formulating and submitting a query is to evaluate and exploit the results. Evaluation refers here to the process of scanning the result listing and deciding on the relevance of the individual results. If a result seems relevant for the user, it is opened for closer inspection. The Lycos search engine logs show that users evaluate results contained in the first two result list pages, meaning that users evaluate 10–20 results. From these result list pages only a few actual result documents are opened. One query typically ends in opening one or two result documents for further consideration.

These observations have a major impact on the requirements of the search engine functionality and the user interfaces. This means that the ranking of the results is a crucial feature. A query may result in hundreds of thousands or even millions of result documents while the user is willing to evaluate only the first ten or twenty of them. In these circumstances, it seems evident that the ranking method will fail from time to time. We can also see the meaning of the observations differently. It means that the user interfaces of the search engines face a serious challenge in delivering the relevant results to the users.

2.2 SEARCH PROCESS

The every day experience of Web searching may seem confusing and somewhat chaotic. The actions the user takes may vary from time to time

and depend on the situation. However, we need a model of the search process at an appropriate level in order to be able to understand the meaning of different actions and artifacts involved in the process. Because Web searching is a special case of general information searching (retrieval), the obvious source for such a model could be information retrieval (IR) studies.

At the core of traditional IR studies is the process of matching a query against the documents. For example, according to Robertson (1977), the *Swets* model identifies two steps in the matching process. In the first step, the value of a matching function is computed between the query and each document in the data set. The second step selects the documents with the highest values from the first step. This model emphasizes the role of the information retrieval *system* in the search process. The matching problem is difficult and thus an understandable and important focus area. However, being so focused on the matching process, the user is almost completely eliminated from this view. Our research question involves the behavior of the user and thus, this model is not appropriate for our purposes.

During the 1980s more and more studies started to shift towards the users and the user interaction with the information retrieval systems (Saracevic et al., 1988). Because the user of the system is focused on, the searching model also changed. A general model (Saracevic et al., 1988) of information seeking and retrieving identifies seven phases in the search process. The phases are characterized by the following events: 1) user has a problem to be solved, 2) user seeks to resolve the problem by formulating a question, 3) presearch interaction with a searcher, 4) search formulation, 5) search activity and interaction, 6) delivery of the results, and 7) evaluation of the results. This model brings the user into the process and emphasizes the actions taken before the actual use of the information retrieval system. Vakkari (1999), on the other hand, has stressed that information seeking happens in a context where information is needed for a certain purpose (a specific task).

The information seeking approach has also been applied to Web search studies. Choo, Detlor, and Turnbull (2000) developed a new model for Web information seeking that describes different types of search behaviors in terms of scanning modes and search moves. This work does not provide us with more detailed information about the focus of this study: the nature of interaction with the search engine.

Sutcliffe and Ennis (1998) proposed a cognitive model of users' information searching behavior. According to the model, the four activities in the information searching process are: 1) problem identification, 2) need articulation, 3) query formulation, and 4) results evaluation. The model includes iteration so that unsuccessful queries can

lead to a new problem identification. This is an important point in the Web environment as about half of the query sessions have been shown to contain more than one query (Spink et al., 2002). However, the emphasis on cognitive processes in the model reduces its utility in our case where the focus is in the system-user interaction.

Marcia Bates (1989) has proposed a model for online search interfaces that contains and actually emphasizes the iterative or progressive nature of the search process. Her notions of browsing and berrypicking reflect the fact that the information need is rarely constant even over one search session. The query results gathered may affect users' search strategies, query formulations and even the information need. Her model describes information search as an evolving process where the information need is continually sifting and it is fulfilled not by one query but by a set of queries produced in this process.

Shneiderman, Byrd, and Croft (1997; 1998) have proposed a search user interface framework that is based on a four-phase model of the search process. This model emphasizes the interaction between the user and the system and is thus consistent with our research question. This is why we have used this model in our study. According to the model, the search process consists of the following phases:

1. **Query formulation:** the initial phase where the user formulates the information need in terms of a query. The phase also includes the decisions about the information source and the fields of the documents to search.
2. **Action:** starting the actual system-performed search operation. This may happen implicitly or explicitly depending on the search system. For instance, Microsoft Windows help indices start the search implicitly as the user types in text while a typical Web search engine requires users to press enter or click a button in order to start the search.
3. **Result evaluation:** when the search is performed, the results are presented to the user and the user needs to evaluate them in order to find the relevant documents. Typically result evaluation is facilitated in current search engines by presenting the results as a ranked list with short document summaries.
4. **Query refinement:** search is typically an iterative process where the results of one query affect the next queries. In this respect, it is important to make it possible for the users to easily edit and modify the current query.

Our contribution is targeted at facilitating the user performance in the result evaluation phase while the other phases are more or less excluded

from the studies. However, all the phases are certainly interconnected and our research in large (within our research group) has contributed to the other phases as well. For example, we have designed aids for facilitating query refining and studied the users' query formulation skills. These studies are nevertheless beyond the scope of this work.

2.3 INFORMATION FORAGING THEORY

The above search process models evolved from a strictly system oriented model towards models where the human operator had a bigger role. We can still take a step and look into models of human behavior in the search process. What can be said about the searcher? Which factors and motivations guide the actions in an information searching task?

Information foraging theory (Pirolli & Card, 1995, 1999) answers these questions by analyzing the human activities associated with the search process. As the name suggests, the information searching process is compared to food foraging and the analogies are used widely in the theory.

Information foraging theory consists of three models. The *information patch model* describes how the information is scattered around the environment (physical or virtual) and how information seekers allocate time and effort in order to find relevant information. The *information scent model* is concerned with the process of identifying valuable information from cues that are available in the environment or information space. Lastly, the *information diet model* addresses the selection of the actual information items.

For the current work, the most interesting part of information foraging theory is the patch model. The central idea is that the information seeking process starts by first locating a patch of information, an area in the (physical or virtual) space that has high information concentration. Next, the information is gathered within this patch as long as it appears to be more efficient than locating a new patch. Note that the theory implies modification of the strategy employed in order to maximize the rate of gaining valuable information. In other words, the information searching process is a constant calculation of cost and benefit. Certainly, this calculation may happen without conscious effort, but it affects the behavior.

Let us consider a simple Web search session as an example. The query formulation can be seen as an initial effort to find the first information patch. As we come across the patch (get the query results), we start to evaluate it. We look for relevant pieces of information and use all available cues to find it. For example, keywords in bold face and the context in

which they appear in the result summary can provide the user with a scent of relevant information.

If the patch, in our case the list of results, appears to be too sparse in relation to our information need, we start to look for a new information patch. In our example this means formulating a new query. The decision on when this happens is affected by the personal characteristics of the searcher. Some searchers find it easy to look for new patches (formulate queries) and some do not. The available tools also affect the decision. If the search space can provide the searcher with easy and efficient tools for finding a new patch or for finding new relevant results within the current one, it affects the decision on when to switch to a new patch.

In terms of information foraging, our research aims at providing the users with new means of finding new information patches. The patch of information in our context gets a somewhat different meaning than in the previous example. Because the result set is divided into easily accessible clusters, one such cluster can be seen as an information patch. In effect, the result categorization approach provides a user with multiple easily accessible information patches with one query. This is likely to reduce the cost of changing patches and thus the users become more demanding in the evaluation of one patch. Because the patch switching is easier, the searchers can concentrate on patches whose information density is high.

The authors of the information foraging theory have proposed similar categorization based methods for enhancing (Web) information access. In the beginning, this was based on automatic clustering in the Scatter/Gather (Pirolli & Card, 1995) system and later on automatically extracted structures from Web link graphs and users' navigation actions (Pirolli, Pitkow, et al., 1996; Chi et al., 2001; Chen, LaPaugh & Singh, 2002).

2.4 WEB SEARCH ENGINE USER INTERFACES

So far, we have seen that Web searching is a frequent and important part of modern information management. The search process models agree that there is an information need that is transformed into a query string. The information retrieval system compares the query to the documents and presents the best matches to the user. This is the high level principle of how the present search engines work.

Let us now take a look at the state of the art of commercial Web search user interfaces and the tools they provide to the users in order to overcome the challenges of Web searching. We include in our review three major Web search engine user interface types: 1) ranked result list (e.g., Google Search Engine), 2) directory service enhanced result list (e.g., Yahoo! Search Engine), and 3) query refinement suggestions (e.g., Teoma

Search Engine). The list could be accompanied with search result visualizations and automatically categorizing search engines. We consider these user interface solutions as emerging and thus they are covered in the next chapter.

Ranked List User Interface

Perhaps the most ubiquitous and most popular type of search engine user interface is a ranked list user interface. It is simple and easy to understand, which probably explains its popularity. Google search engine is a well known example of such a user interface.



Figure 1. Google search engine user interface.

Google (Figure 1) is fairly conservative in its user interface design. It relies mostly on its unique PageRank mechanism to bring the most relevant results to the top of the result list and works remarkably well. However, it cannot deliver alternative results in queries where search terms have multiple meanings.

Another useful feature in Google is the spell checking of the queries that brings up suggestions for alternative ways of spelling the query words. This reduces errors in the query formulation, but does not solve the problem of ambiguous queries. Another exception to the simple basic user interface is the sponsored links (on the right in Figure 1), which bring up advertisements related to the user's query.

Directory Enhanced User Interface

Yahoo! is a good example of user interfaces utilizing human moderated directory (Figure 2). The Yahoo! directory consists of an hierarchical categorization of terms and topics and of Web sites and pages assigned to those categories. Because the directory was created by humans, the quality of the categorization is good. One problem with such categories is that the categorization may be unfamiliar to the user and thus, browsing may be difficult at first. The search functionality helps to get started and the categories associated with the relevant results allow the users to find other relevant documents as well.

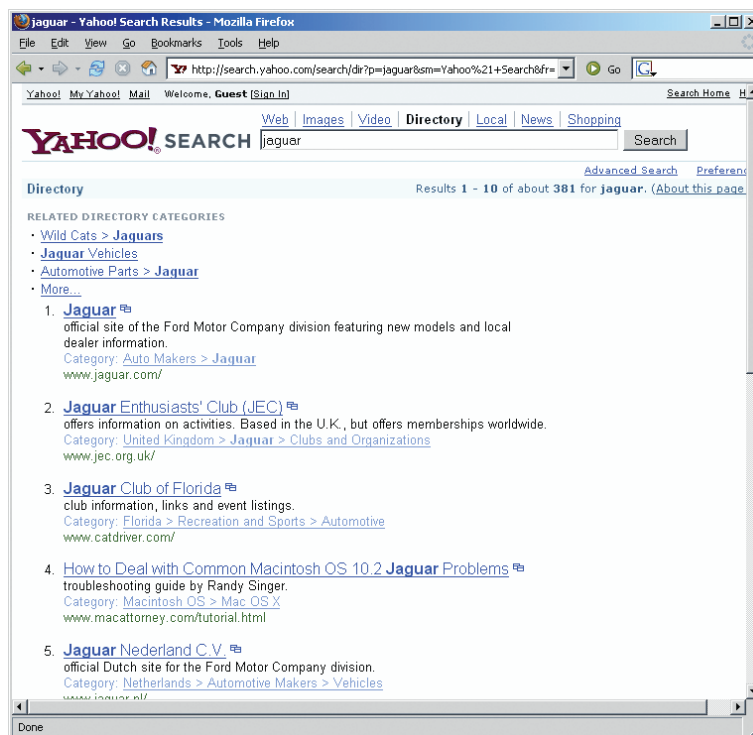


Figure 2. Yahoo search engine user interface.

Yahoo! categories are utilized in the user interface to give further contextual information about the results. First, a few relevant categories are listed, through which the user can start browsing their contents. Second, each individual result item is accompanied by a category link that describes the category to which the result belongs.

User Interfaces with Query Refinement Aids

Query refinement aids aim to help users to express their information need in a more precise form. Typically, the initial query must be entered without assistance, but the subsequent queries can be affected by the refinement aids.

Teoma is a good example of a user interface with query refinement aids (Figure 3). Teoma puts the query refinement suggestions into a significant role in the user interface.



Figure 3. Teoma search engine user interface.

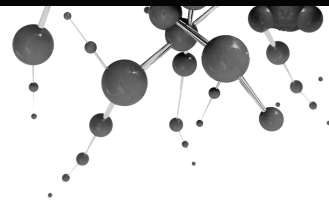
Teoma clusters Web pages according to so-called communities, meaning Web pages that “are about or are closely related to the same subject” (<http://sp.teoma.com/docs/teoma/about/searchwithauthority.html>). The user interface presents these communities under Refine functionality that allows users to rephrase their query with a particular topic covered by a community. For the user, the selection of such a refinement appears as a new search with more focused results. In addition to the refinements, communities are used to look for experts in the area. These results are presented to the user in a separate area in the user interface, giving it a central role in the user experience.

Yahoo! is another example of this kind of functionality. In regular searches, Yahoo! has a so-called “Also try” feature. This feature presents a set of queries entered by other searchers that are similar to the current query. Typically this feature can help users to narrow down the search queries and become aware of the other meanings of the query words (in ambiguous queries). When the user selects one of the proposed query formulations, the current query is replaced with the new one and the new query is executed.

To conclude the present state of Web information access, we can see that Web searching is a frequent but challenging activity for a huge user population and thus, well worth studying. Multiple user interface solutions have been proposed for the application domain, most of which are decidedly simple. The solutions work relatively well, given the vast variation in the user population. The sound, simple and well tested

solutions of current user interfaces are something that we wish to utilize in our solution as well. Users are accustomed to the simple ranked result listings and thus, we must consider the benefits of them thoroughly in new solutions.

The theoretical work on searching sets a framework for our new solutions. The search process models imply a separate result evaluation phase that will be in the focus of the current research. The model has also influenced the test setups used in our studies. In addition, the information foraging theory provides us with factors affecting the actions of the searchers. These views have inspired us to look for ways of helping the users to find new and meaningful information patches.



3 Enhancing Search Result Access

3.1 OVERVIEW

There is a large number of user interface related techniques that can be used to enhance users' access to search results. Initial query formulation is difficult, but refining the queries and evaluating query results have both received a lot of attention. The actual techniques to enhance users' performance vary widely, ranging from simple text layouting to complex visualizations.

The work for enhancing search result access has started already in the 1950s. Back then, the output devices were rather limited (printers and simple character-based displays), which set strict limitations to the solutions available. With the development of the display technologies, the original problems have changed, but the fundamental questions of how the search results should be presented and what kind of tools the users need in order to access them remains.

In the following, we will take a look at early work in keyword-in-context (KWIC) indices, newer work on result visualization techniques, query refinement suggestions and current categorizing Web search user interfaces. All of these techniques are related to our solution and have inspired our work. A more thorough survey of the techniques and different approaches can be found in the Marti Hearst's (1999) chapter 'User Interfaces and Visualization' in the book 'Modern Information Retrieval'.

3.2 KEYWORD-IN-CONTEXT INDEX

The keyword-in-context (KWIC) index is a type of concordance (word index) designed for presenting the results of a search query. Keyword-in-context index dates back to 1959, when Hans Peter Luhn published an article about it. This discussion is based on Salton's description of the technique (Salton, 1989, pp. 384–386).

The idea in KWIC is to provide users with a meaningful and easy-to-scan query result listing in a limited environment such as those available in the late 1950s. The system is character based and it displays one result per one line of text, typically representing the result by its title (such as a book title in a library system).

The keyword (query term) used in the query is central in displaying the results. The titles are printed in the KWIC index so that the instances of the keyword are aligned (see Figure 4). The scanning of the list of hits is assumed to be fast and easy as the user can easily see the context in which the keyword appears in each of the result items.

graphic scheme based on	abstract and index cards
tic information using	abstract and index publications
publishing modern	abstract archive of alcohol literatu
company pharmaceutical	abstract bulletins
a punched card	abstract bulletin
the	abstract file on solid state and tra
relation of an	abstract of the technical report
from journal article to	abstract to its original
	abstract

Figure 4. An example of keyword-in-context (KWIC) index (picture after Salton (1989)).

The typical way of presenting results in modern Web search engines can be seen as one type of KWIC. The Web search results are typically represented with a short summary text that contains the title of the document and a so-called query-biased text summary. This kind of summary is built so that short parts of the text (*snippets*) containing the query keyword are selected from the document (see Figure 5). The approach has been shown to be advantageous for the searchers (Tombros & Sanderson, 1998; White et al., 2001).


[Advantages of query biased summaries in information retrieval](#) 
Advantages of **query biased** summaries in information retrieval Advantages of **query biased** summaries in information retrieval Anastasios Tombros Mark Sanderson ...
portal.acm.org/citation.cfm?id=290947&coll=portal&dl=ACM - [More from this site](#)

Figure 5. An example of a query biased search result summary from Yahoo! search engine. The query was 'query biased'.

Typically the keywords are highlighted (e.g., with bold type face) in these query-biased result listings. The purpose and the effect of bolding is comparable to the term alignment in KWIC. Bolded keywords direct the visual scanning process so that keywords are easily found and thus, the user is able to determine the context in which they appear in the results.

3.3 VISUALIZING SEARCH RESULTS

One possible way of improving the user interfaces of the search engines is to visualize the results. Visualization of the extensive amount of information sounds appealing. Visualizations are assumed to have the power of delivering a clear insight of a problem.

However, the actual status of search result visualization is different. Unfortunately, visualizing unstructured textual information is all but a clear case. We can find a considerable set of result visualization techniques in the literature, but none of them have met the great expectations people have of them. We can distinguish two major approaches to visualizing search result (Zamir, 1998):

1. visualizations based on properties of individual documents, and
2. visualizations of the inter-document relationships.

When the document properties are visualized, the options are to utilize the query terms and visualize their distribution or to use known attributes of the documents such as publication date, author, and type of document (e.g., book, article, magazine).

Envision (Fox et al., 1993; Heath et al., 1995; Nowell et al., 1996) is a user interface for a library system that employs the latter approach. In Envision, the results for a query are displayed by icons in a matrix (Figure 6). The user can control the attributes visualized by different visual variables, for example, the year of the publication can determine the position of the icon on the X-axis and the icon can represent the type of the publication. Note that Envision relies heavily on structured data that is available in library systems. A similar approach is much harder to implement in the Web environment. The GRIDL prototype employs a similar approach, but adds so-called *hieraxes*, categorical and hierarchical axes to the visualization (Shneiderman et al., 1999).

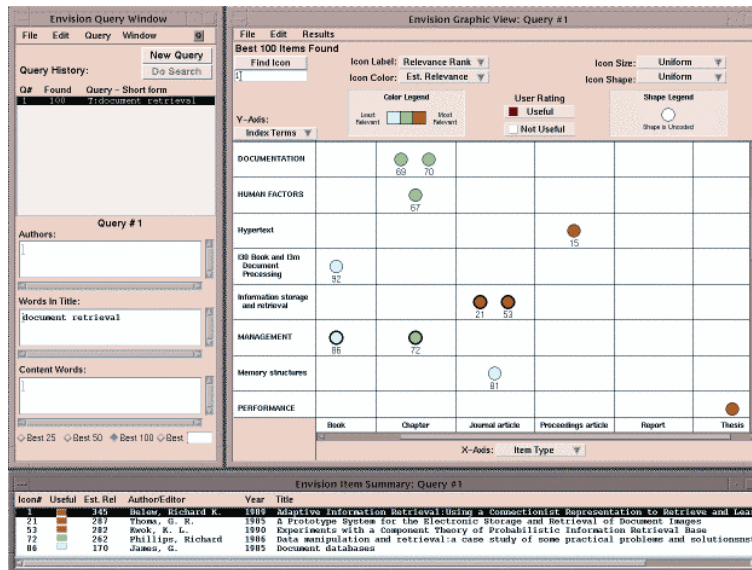


Figure 6. Envision user interface for search results.

Perhaps the most widely known example of visualizations based on query term distribution is TileBars by Marti Hearst (1995). TileBars represent the document as a rectangle whose length is proportional to the length of the document (Figure 7). The rectangle is broken into a number of bins whose darkness represents the density of particular query facet occurrences within the corresponding section in the document. The rectangle is divided into rows that stand for each of the query facets.

Veerasamy and Belkin (1996) proposed a system where documents are represented by vertical columns and query terms are located in rows. In the intersections, there are bars whose length visualizes the weight of that term in the corresponding document. In a user study, the system was found to be beneficial, but no convincing evidence for its utility and for the visualization approach in general was found.

InfoCrystal (Spoerri 1994a, 1994b) is a way of visualizing the query term distribution between documents rather than within them (Figure 8). The idea in InfoCrystal is to take the query terms and display the number of

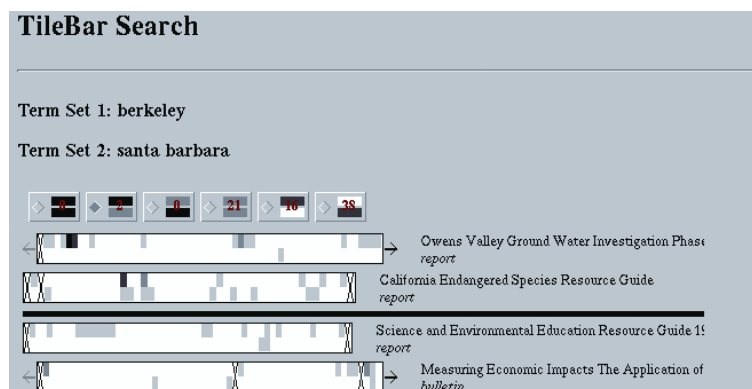


Figure 7. A screenshot of TileBar User interface.

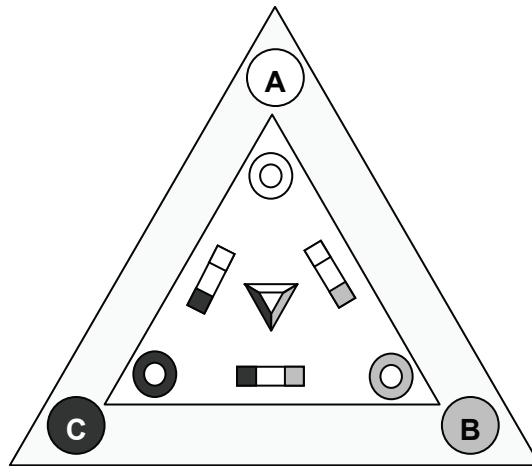


Figure 8. Original form of InfoCrystal. The icons have the following meanings: solid circles = query terms, hollow circles = instances of terms alone, rectangles = instances of two term intersections, hollow triangle = instance of three term intersections (picture after Spoerri 1994a).

matches that each Boolean (e.g., 'and') combination of them corresponds to. The matches are represented with icons whose form and position indicates what kind of combination is under consideration. For example, a rectangle contains two ends and it thus indicates the number of matches that the two facing concepts have given the Boolean operator. InfoCrystal aims to provide an understanding of what parts of the query are potentially too restrictive or not discriminatory enough. However, the resulting visualizations seem to get rather complicated and hard to understand.

The visualization idea of InfoCrystal was applied to meta searching in the MetaCrystal prototype (Spoerri 2004a, 2004b). MetaCrystal visualizes the overlap of the search results from multiple search engines. There the different icons in the crystal visualization represent results from different search engines. The user is able to focus on documents found by certain search engines.

Visualizing the whole search result collection and the relationships between individual items has been another popular approach. Kartoo (Kartoo Search Engine) is a publicly available tool that makes Web searches and visualizes the results as a concept map (Figure 9). The idea of seeing the whole result set at one sight is appealing, but in practice, the map is difficult to interpret.

Cat-a-Cone (Hearst & Karadi, 1997) is a research prototype that visualizes the search results in a three-dimensional cone tree. The structure for the tree comes from a predefined classification system (MeSH) that is also used to classify the result documents. The tree shows all the topics in

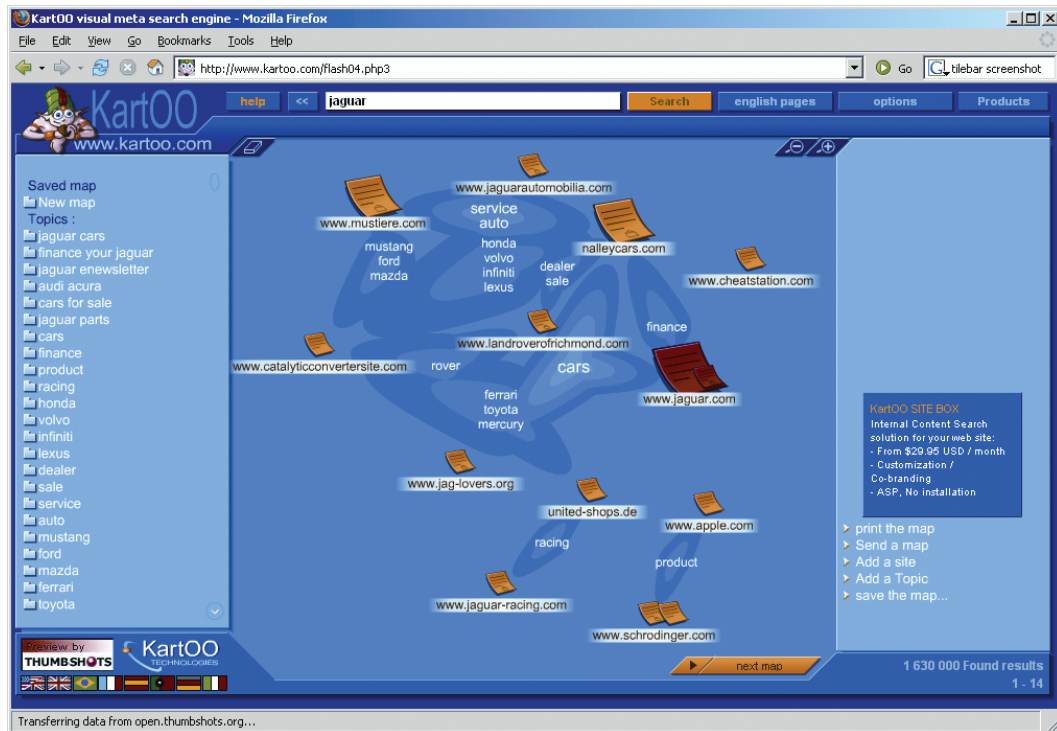


Figure 9. Kartoo search engine user interface displaying the results for a query 'jaguar'.

which the results belong and aims to provide an easy access to them. The essential feature in the Cat-a-Cone system is its ability to display multiple selected categories in the category hierarchy simultaneously. The three-dimensional representation of the category tree makes this possible.

Self-Organizing Maps (SOM) is a general technique for mapping any multidimensional data into a two-dimensional space. Self-Organizing Maps are implemented using artificial neural networks and the map generation phase can be seen as a teaching phase of the network. The resulting map places closely related data items (such as Web documents or search results) next to each other.

Lin, Soergel, and Marchionini (1991) were among the first to apply SOMs in information retrieval and for handling document sets. They proposed a user interface where a document collection was presented as a map with the most dominant concepts visible and with borders between the major regions. Later, the applicability of SOMs was demonstrated for large databases with the WEBSOM (Figure 10) system (Kohonen, 1997; Kaski et al., 1998). WEBSOM was intended to enable interaction with and comprehending a large document collection (such as static Web pages or Usenet news articles). The approach is expected to reduce the problem in selecting appropriate query terms as the query formulation step is essentially eliminated from the search process. Zamir (1998) suggested that the approach could be applied to search results, in addition to static document collections.

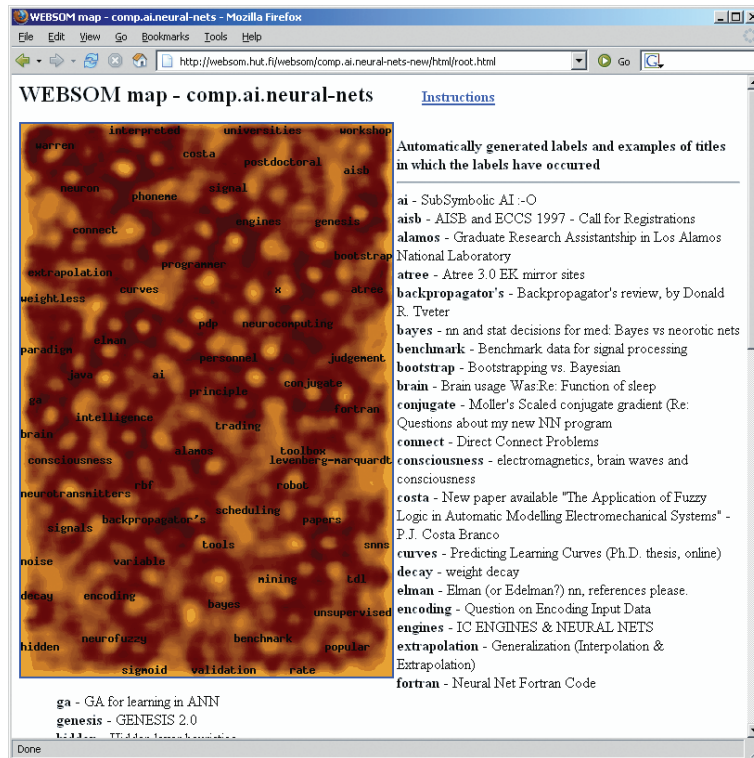


Figure 10. A screenshot of WEBSOM demonstration.

This brief overview of the (Web) search result visualization techniques is not intended to be comprehensive. Result visualization is a field of active research activities.

3.4 QUERY REFINEMENTS

The goal of result visualization is to help users to understand the retrieved result set. The approach may not help in a case where the result set does not contain any relevant documents, although such an understanding is an important step. Aids for refining the query formulation can be helpful in such a situation.

In the information retrieval community, most work relating to query refinements has been done in automatic query expansion. Automatic query expansion refers to a process where the search system automatically adds synonyms or other closely related terms to the query in order to improve the recall or the precision of the query. Our approach emphasizes the active role of the user and thus the automatic query refinement *suggestions* are of greater interest here.

Vélez, Weiss, Sheldon, and Gifford (1997) proposed a fast query refinement algorithm. The system, called RMAP, uses a pre-computed corpus in order to speed up the process of computing the refinement suggestions. This is important in Internet searches, where the number of searches is vast. In the evaluation of the system, it was found to be an

attractive approach especially when processing time is critical. However, the evaluation did not include end users, and thus, it is not known whether the approach increases users' search effectiveness.

Belkin and his colleagues (Belkin et al., 1999) followed a different path in the evaluation by presenting two-term suggestion systems to 36 volunteer searchers. The first system (RF, relevance feedback) was based on explicit feedback from the users. The second system (LCA, local context analysis) produced term suggestions automatically. The comparison produced the first statistically significant difference measured in the TREC interactive track as they discovered that the LCA system was considered to demand less user effort while using the system.

Both of these first two systems can be considered to be traditional information retrieval systems that may have a somewhat limited user population. However, a similar automatic term suggestion approach has been employed in the Web environment as well. Bruza, McArthur, and Dennis (Dennis et al., 1998; Bruza et al., 2000) have proposed a query refinement system called Hyper-index (Bruza & Dennis, 1997) and a Web search user interface called Hyper-index Browser (HiB). Hyper-index Browser requires the users to always refine their queries before they are presented with the search results. In practice, the users are presented with a list of possible query refinements after query submission rather than a list of results. When one of the refinement suggestions is selected, the query is actually executed and the results are presented to the users.

The Hyper-index Browser was evaluated in user studies where it was compared first to Excite and then to Yahoo! and Google. The studies produced evidence that the approach is beneficial in ambiguous queries. It was also noted that with HiB, the users spent the least amount of time (relatively) in evaluating the actual result documents. This indicates that the time spent in refining the query (making a selection from the lists of suggestions) can be gained back in the result evaluation phase.

Query term suggestions are also used in commercial search engines and Peter Anick has studied their actual use. The research contains a system prototype, Paraphrase Search Assistant (Anick & Tipirneni, 1999), and a log based study of the use of the AltaVista query term suggestion system called Prisma. This fairly extensive (over 15,000 search sessions) study concluded that the query suggestions were as effective as manually reformulated queries when they were used. However, the vast majority of the query reformulations were made manually (Anick, 2003).

3.5 CATEGORIZING WEB SEARCH USER INTERFACES

Another group of search engines utilizes clustering methods. These search engines include Vivísimo (Figure 11, Vivísimo Search Engine), WiseNut (Figure 12, WiseNut Search Engine), and iBoogie (Figure 13, iBoogie Search Engine). All of these engines take a set of results and compute categories for them online.

The resulting categorization is hierarchical in all these search engines. The categories are presented in the user interface as a list (either at the top of the page or to the left of the results). The exact details of the categorization algorithms have not been published, but their functionality is similar to the users. The user selects one of the categories and the user interface displays the documents that belong to that category. The major difference from the previous query refinement aids is the fact that selecting a category does not execute a new search, but alters the way the result listing is displayed.

From the research perspective, there are a few problems with these commercial systems. First, the actual categorization algorithm is not public, and thus, the understanding cannot be shared in the research community. Second, there are no published evaluations of the usefulness of the categorizations. Addressing this shortcoming is one of our main contributions.

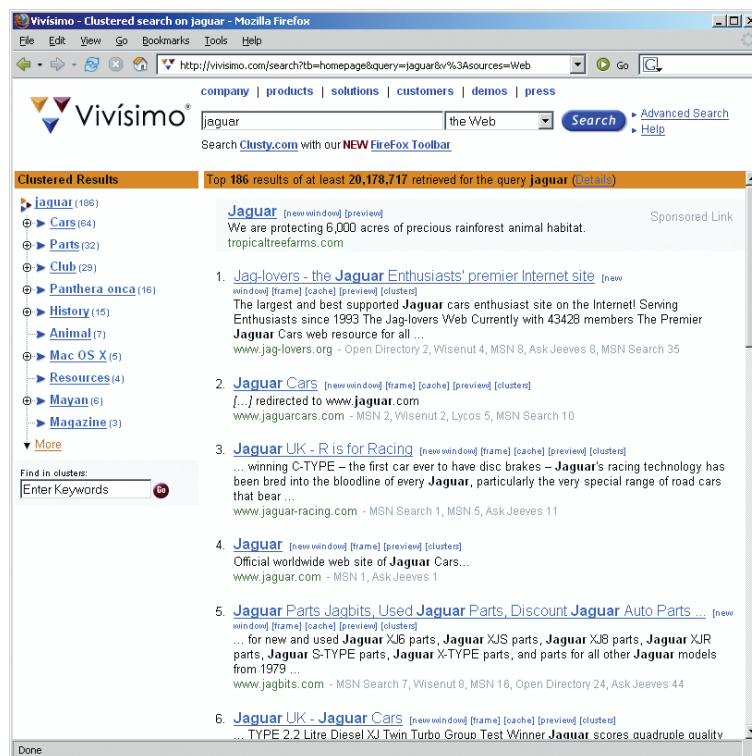


Figure 11. Vivísimo search engine user interface with the categories for query 'jaguar'.



Figure 12. WiseNut search engine user interface with the categories on the top for query 'jaguar'.

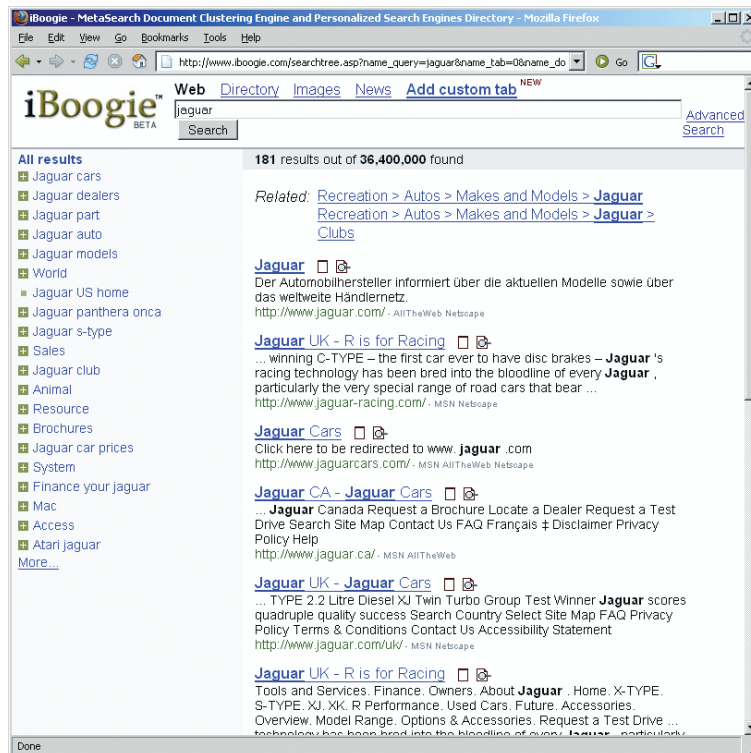
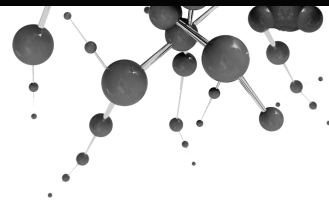


Figure 13. iBoogie search engine user interface with the categories for query 'jaguar'.

All of these commercial systems employ a hierarchical categorization, although WiseNut uses it only in certain categories. We acknowledge the benefits of the hierarchical categorization schemes and appreciate them in certain situations. However, we assume that hierarchical categorizations could be too elaborate in everyday searching, where the search topics and thus, the concept hierarchy could be hard to understand or require too much attention. Search result evaluation is a tedious task and we believe that users may be impatient in going through the results. In that case, it could be better not to present them with a comprehensive hierarchy but rather, with a more compact overview. Based on these assumptions, we have selected a flat categorization approach for our prototype.



4 Enhancing Search Result Access with Categorization

In the previous chapter, we presented multiple ways to improve the search result access, including ways to enhance result summaries, result visualizations, query reformulation aids, and result categorization. In this work, we have chosen the categorization approach. Next, we will discuss the relevant theories and the previous research in the field.

We will first discuss the cluster hypothesis, which is a theory describing the rationale behind the categorization approach. It will be followed by a brief historical review of how the technology has evolved, what the most prevalent categorization technologies are, and how they work. Next, we will go through various research prototypes that have employed categorization techniques in accessing search results. Finally, our approach and our research prototype, Findex, are presented.

4.1 CLUSTER HYPOTHESIS

In 1971 Jardine and van Rijsbergen published the so-called *cluster hypothesis* in the context of information retrieval. The cluster hypothesis states that *relevant documents for a query tend to be similar to each other*. This is an important discovery for our approach, as it is the underlying motivation for clustering the results.

In the 1970s the clustering approach was not typically employed for accessing the query results, but rather, in the search process itself. If the document database can be divided into similar clusters, the searching efficiency can be improved and thus more elaborate methods could be used in the process. In practice, this means that the search process would

consist of two phases. The first phase would locate the relevant clusters and the second would retrieve the actual documents. The second phase needs to consider only the documents in the clusters identified and thus the number of documents is smaller than in the whole database. This makes the use of computationally intensive retrieval (or matching) algorithms feasible.

Voorhees (1985) proposed a new method for testing the cluster hypothesis on a given document collection and noted that cluster based searches performed better than sequential searches in smaller collections. This is a good reason for clustering the search result lists, which are typically clearly smaller than the document collections. This is supported by the results from a study by Hearst and Pedersen (1996) where the hypothesis was tested with the Scatter/Gather system. They added a new assumption that clustering should honor the different retrieval contexts and thus the clustering should be done in context of a query. Both of the conditions (small collection and query-dependent clustering) can be met in result set clustering.

When focusing on the performance of the end user in evaluating the search results, the basic idea of the cluster hypothesis still holds. When similar search results are clustered together in the user interface, the workload of the *user* is reduced. Ideally, the user needs to first locate the relevant clusters and then evaluate the documents within those clusters. Again, the number of documents to be evaluated is reduced as the number of documents in the relevant cluster is smaller than in the whole result set.

4.2 CATEGORIZATION TECHNIQUES

The literature describes a vast number of technologies used in making meaningful categories of textual documents. Before we present an overview of the techniques, we define the terminology used in the discussion. The terminology used in the literature is not always consistent.

Terminology

The terminology used in discussing document or text categorization varies considerably from source to source. Many terms, such as categorization, clustering, grouping, and classifying, can all be used to refer to the same high level concept of organizing a set of textual documents into a number of smaller groups. In this work, the term *categorization* refers to both the process of making such an organization and the outcome of the process.

There are multiple automatic techniques for computing categories. The techniques can be divided into two groups, and we use the following terms for them:

- a) *clustering* techniques bring similar documents together, and

b) *classification* techniques assign documents to predefined classes. Multiple implementation techniques have been proposed for both approaches.

In addition to clustering and classification, *keyword extraction* is a technology that can be used in document categorization. Keyword extraction is a process where descriptive words or phrases are found in a textual document.

When a categorization has been built with a given technique, the resulting categorization is said to contain *categories*. The terms *class* and *cluster* refer to categories in the context of classification and clustering techniques, respectively. *Group* can also be used to refer to categories.

Document Representation and Similarity Measures

Many categorization techniques, especially clustering techniques, are based on similarity measures between the documents. Document similarity can be measured in a number of ways, but the most common ones include document representation in weighted vector format, so-called vector space model (Salton, 1989). In the vector space model, each document is represented by a vector of words where the importance of each word is represented by a number, e.g., its frequency.

As some words like 'a', 'an', 'the', or 'and' are frequent, such words are often removed from the vector. This can be accomplished by using so-called *stopword* lists that enumerate words to be discarded.

A more sophisticated method is to use more elaborate weight measures than simple word frequency. TFIDF (Term Frequency Inverse Document Frequency) is a statistical technique widely used in achieving this (e.g., Salton, 1989, pp. 280). There are multiple formulas for computing the measure, but the basic principle is the same. The measure compares the frequency of a term in the current document (*tf*) to the number of documents containing the term in the whole collection (*df*). Because the document collection frequency is used as an inverted multiplier, the terms that are common receive a small weight indifferent of their frequency in the current document. In contrast, words that appear frequently in the current document, but are rare in the collection, receive a higher score and are considered descriptive.

As the documents are represented by weighted vectors, their similarity (or the distance between them) can be computed using methods from vector algebra. Again, there are multiple ways of applying them, but the so-called *cosine measure* is perhaps the most widely used. The cosine measure is defined as the dot product of two vectors divided by the product of the length of the vectors (d_1, d_2):

$$\text{cosine}(d_1, d_2) = (d_1 \cdot d_2) / |d_1| |d_2|.$$

The length of the document vectors is typically normalized to be 1 (unit vector) and thus, the cosine measure is simply the dot product of the vectors.

Clustering Techniques

An overwhelming selection of clustering methods can be found from the literature. Not all of them are relevant for clustering textual documents, but even those that are, are too numerous to present here. Berkhin (2002) has presented a comprehensive survey of the clustering methods. We will concentrate here on the basic clustering techniques that are most closely related to the problem of categorizing search results.

Clustering techniques can be divided into those that produce a hierarchical clustering (hierarchical methods) and those that produce an un-nested partitioning of the documents (partitioning methods). There are several techniques for implementing both. The following partial classification (after Berkhin, 2002) illustrates the relationships of the methods we will describe. Our description of the methods follows the order of the classification.

Partial classification of clustering methods

- Hierarchical methods
 - Agglomerative clustering
 - Divisive clustering
- Partitioning methods
 - K-means
 - ...

Agglomerative clustering is an iterative bottom-up process where the two most similar clusters are merged together at each step. The process starts with individual data points as clusters and in the end the whole collection is contained in one root cluster. The resulting data structure is a tree called a dendrogram. As the building process suggests, each node in the dendrogram branches into two until the leaf nodes are reached. The quality of the clusters is dependent on the similarity measures between the documents (and clusters) and several variations of the method described above have been employed.

Divisive clustering is the other way of building hierarchical clusters. It uses a top-down process where the document collection is first considered as one cluster that is then divided into smaller sub-clusters until they contain only individual data points. The factors that affect the resulting clusters include algorithms to decide which cluster to split and the criterion for assigning the documents to the new clusters. In the process, document similarity measures are needed, for example, in determining which cluster has the most variation in it. Agglomerative algorithms are more common of the two.

There are also a number of clustering techniques that produce a flat categorization. We will describe a well-known and most widely used technique in document clustering. In the *K-means* algorithm, the basic solution is to first select target number K cluster centroids (central points in the document space) around which the documents are then clustered. The selection of a fixed number of centroids is also the source for the name of the technique. In principle, each document is assigned to the closest cluster (represented by a centroid).

The first interesting issue in the algorithm is the selection of the centroids. Multiple approaches have been tried, including random selection. The second issue is the selection of the documents to be associated with a given centroid. This reduces back to calculating distances between vectors, because both the centroids and the documents are typically represented as vectors. As stated, documents are usually associated with the closest centroid.

As the initial clustering is achieved with the previous procedure, the result can be optimized in an iterative process. This can be done by recalculating the centroid based on the documents contained in the cluster and then reassigning the documents again to the new centroids. Another optimization option is to find an optional centroid and to analyze the effects it would have on the clustering. Note that the K -means algorithm can be used to build a hierarchical clustering by applying the algorithm recursively to the clusters once computed.

All the techniques discussed above may produce good quality clusters if all the parameters and factors are adjusted successfully. However, naming the clusters is a major problem. As the clusters are generated on the fly from a set of documents, the automatic description of the clusters has proved to be extremely difficult (Popescul and Ungar, 2000). Typical solutions employ representative words (frequently occurring or strongly weighted). However, the resulting word lists are typically hard to understand. This naming problem is a major shortcoming of these classical clustering methods, which is one of the motivations for our approach.

Classification Techniques

The main difference between clustering and classification methods is the source of the resulting structure. Where clustering creates the structure in the process, classification methods rely on predefined, typically man-made topic structures, which typically are hierarchies (such as MeSH for medical documents or Yahoo! directory for Web content).

The classification process is based on a set of target classes that are characterized by a set of features. The classified items are also characterized by similar features and the classification algorithm must make a decision on which class a data point belong to. We can find a

number of techniques used for this purpose from the literature ranging from simple nearest neighbor and multivariate regression models to various Bayesian models and neural networks. The bottom line is that the algorithm must place the data item in one of the available classes.

Machine learning techniques are often applied in classification. Learning algorithms are used to optimize the classification process by teaching the system with a correctly classified training set. Such a set could be, for example, from a Web directory service (such as Yahoo!). Because the correct classification of each data point is known in the training set, the parameters affecting the classification can be adjusted in the process.

One typical feature of the classification methods is that they do not directly support hierarchical classifications although the target classification scheme is often hierarchical. If nothing is done, the structure is flattened because of the classification technology. There are, however, systems where the complete hierarchy of the target classification scheme is employed. One can build a hierarchy of classifiers so that in the first level, a coarse decision needs to be made (e.g., distinguish computer articles from automobile articles). The next classifier depends on the decision made by the previous one as the process proceeds from one level to the next. Such an approach is used, for example, by Dumais and Chen (2000) and by Koller and Sahami (1997).

For the end user, the most notable consequence of using a classification technique is the quality of the category names. Because the documents are classified to an existing taxonomy, the class names are also predefined and can be carefully selected to optimally convey the intended meaning. Thus the naming problem associated with the clustering methods is avoided. However, the classification scheme may be too rough for the given data set resulting in a categorization where all data items are placed in one or two classes. Such a categorization does not reduce the number of evaluated documents enough to realize the promise of the cluster hypothesis.

Keyword Extraction

Keyword extraction is an area of research that is closely related to clustering. It is especially important for our categorization method that is based on word and phrase frequencies. In contrast to our solution, keyword extraction typically aims at automatically extracting keyphrases for describing the contents of a document. Such keywords or keyphrases are often required by academic publications and their automatic extraction would be useful in many ways. Another popular application area of keyword extraction is in the query refinement suggestion systems. There the extracted keywords are used to give the user more options in reformulating the query.

According to Jones and Paynter (2002), Turney (2000) was the first to apply learning methods to the keyword extraction task. Barker and Cornacchia (2000) developed a system that utilized the extractor component developed by Turney, but added a new way of selecting the final keywords from the extracted candidates. The selection was based on noun phrases and their frequency. The extraction process scanned the text word by word and looked for sequences of nouns and adjectives ending with a noun. For identifying the part-of-speech, an online dictionary was used rather than a part-of-speech tagger.

The digital library project in New Zealand has produced a keyword extraction algorithm called KEA (Jones & Paynter, 2002). In addition to simply extracting descriptive words for documents, it was also applied to facilitate search result access in a Web-based library system. Another example of its application is a document clustering system with a special stress on naming the clusters. In both tasks, the automatically extracted keywords were shown to be effective (Jones & Mahoui, 2000). Thus, KEA is closely related to our research.

KEA is based on a supervised learning approach. The system is trained with a set of documents whose accurate keywords are known (e.g., author provided). Each document is transformed into text and stemmed candidate phrases with a length of one to four words are formed from the text. For each candidate, two measures are computed: 1) document distance, which describes how far the first instance of the phrase is in the document, and 2) the TFIDF measure of its frequency. Based on these measures, a Naïve Bayes classifier is constructed. When the classifier is ready, keyword candidates and their measures are computed in the same way and the classifier is used to select the most promising candidates.

KEA is used in multiple user interfaces in different roles. In the simplest case, it can be used in library search to describe the retrieved documents with automatically extracted keywords if author-provided keywords are not available. In KeyPhind (Gutwin et al., 1999) keywords, or keyphrases, are used for refining a query. When the user enters a query, keyphrases that contain the query term(s) are listed. Upon selection of a keyphrase, the related keyphrases and the documents containing the keyphrase are displayed in the user interface.

Phrasier (Jones, 1999) utilizes the automatic keyphrases in a complete browsing, querying, and reading environment for a digital library. Keyphrases are the basis for automatically creating hyperlinks between the documents that share the keyphrase. The user interface also displays the related documents. Kniles (Jones & Paynter, 1999) is a simpler version of Phrasier for the Web environment having basically the same features.

In addition to the digital library environment, KEA has also been tested in the Web environment. Jones, Jones and Deo (2004) presented a system for PDA devices that used KEA produced keyphrases as search result surrogates on a small screen. The solution was compared to displaying document titles, but no performance differences were observed in the study.

4.3 CENTRAL SEARCH RESULT CATEGORIZATION SYSTEMS

Now that we know the basics of clustering textual documents, we can direct our attention to actual systems where the techniques are utilized. We will first take a look at the search result categorizing systems that have had a notable impact on the research in the field. Later, we will make a more extensive survey of the related systems.

Systems where categorization was an explicit part of the end user's experience started to emerge at the beginning of the 1990s. Scatter/Gather (Cutting et al., 1992) was one of the first systems where automatic clustering was tightly integrated to the user interface. In the late 1990s, Grouper (Zamir & Etzioni, 1999) was introduced with a clear focus on categorizing search results. Around the same time, classification based systems were also introduced. These include the SWISH prototype by Chen and Dumais (2000) and the DynaCat system by Pratt and Fagan (2000).

These four systems have been discussed widely in the HCI community and are closely related to our work. In addition, they exemplify the two major approaches: clustering and classification. The prototypes use two types of data sources: digital library (with structured data and complete documents) and Web searches (data limited to summary texts only). This separation is important because the data type affects the techniques used. Figure 14 summarizes the technical approaches and the data sources of the four central systems.

	Digital Library	Web Search Results
Clustering	Scatter/Gather	Grouper
Classification	DynaCat	SWISH

Figure 14. Techniques and data sources of the central prototypes.

Scatter/Gather

The Scatter/Gather user interface is based on an interactively and iteratively built document structure. In the beginning, the whole database

would be divided into a number of clusters (scatter). These clusters were presented to the user by showing a list of representative words and a short list of sample document titles contained in the cluster. The user then selects one or more of these clusters for focusing on the interesting topics. The selected clusters from a new base set (gather), which is then divided (scatter) into clusters again. The user controls the clustering process by selecting the clusters of interest and forms a tailored hierarchical categorization of the document set.

The original idea of Scatter/Gather was to function as a browsing tool for large document collections, but quite soon the idea was employed in accessing search results. From the user's perspective, this does not change the situation much. The difference is that the initial document collection is formed by a search query (the result set), but from there on, the interaction with the system is the same. However, the performance issue is reduced as the initial document collection is considerably smaller than in the original case.

In addition to being among the first systems to actually try and demonstrate search result clustering in practice, research on Scatter/Gather also presented one of the first user studies on such systems (Pirolli, Schank, et al., 1996). As with many experimental technologies, the initial results from the user studies were not a major success. In fact, Scatter/Gather appeared to be both slower and less accurate when compared to the standard information retrieval system based on similarity search. Despite the slightly disappointing results in simple document retrieval, Scatter/Gather was seen to effectively communicate the topical structure of the document collection.

A follow-up study focused on the usefulness of the clustering approach (Hearst & Pedersen, 1996). The overall performance of the system would not be as important as the success of using the categories for a given task. This approach produced results. The study concluded that the users found and selected the most relevant clusters. This is crucial for the cluster hypothesis and the results provided confirmation that the clustering approach may be beneficial in search result access.

Grouper

After Scatter/Gather, the idea of categorizing search results seemed to fade and it was not a popular research topic, but the area was rediscovered in the late 1990s. At that time, the Web had grown to vast proportions and finding information in it became harder. Web searching was an important motivation for the beginning of the next wave of search result categorization systems.

Zamir and Etzioni (1998) were the first to demonstrate the feasibility of this approach in the Web environment. They compared multiple

clustering techniques (Zamir et al., 1997) and finally presented their own clustering method, Suffix Tree Clustering (STC). STC is based on shared words and phrases in the documents and the technique was especially designed for Web searching. The authors call the method clustering, but it is also close to term extraction. For example, the algorithm does not use the classical document similarity measures that are distinctive for clustering algorithms.

Zamir and Etzioni (1999) developed a Web search engine user interface based on the idea. The clustering search engine user interface was called Grouper. Grouper presents the search results in five categories. Each category shows representative words and a few sample document titles much as the Scatter/Gather system did. The interaction, however, is simpler compared to Scatter/Gather. In Grouper, the user simply selects one of the categories and the system will display the documents, whereas in Scatter/Gather cluster selection presents the user with new clusters.

However, Grouper forces the user to make one category selection, because initially only the clusters are displayed. Thus, clusters are emphasized and the design introduces an additional interaction step to the search process, namely the selection of the category. This may not be necessary, because the top results in the search engine rank order could satisfy the user's need. The design may be good in evaluating the use of the categories, but it may decrease the user performance in real situations.

Grouper has not been formally tested in an experiment, but it was evaluated by a log study. The results showed that the users followed more documents in a session and that the time needed to access multiple documents was shorter than when using a conventional user interface. These are positive results and indicate that result clustering is worth exploring further.

SWISH

The SWISH prototype employs another approach, as Dumais and Chen (2000) implemented a hierarchical classifier based on Support Vector Machine (SVM). The classifier was taught with LookSmart Web Directory documents that are organized into a hierarchy of categories by professional human editors. After the teaching phase, the classifier will assign new documents to the best matching categories.

The original user interface of SWISH organized the list of documents by category title names using them as headings. The user could collapse or expand the categories and each document was presented by a one line title underneath the category. The short document summary was available as a hover text on demand. The document title was a link to the actual document and the user interface contained separate buttons for opening

subcategories and displaying more documents within a category (Chen & Dumais, 2000).

SWISH was evaluated with 18 users comparing it to the typical rank order list user interface (Chen & Dumais, 2000). The test setup was one of the sources of inspiration for our own studies as the authors used predefined queries. The results of the study concluded that the category approach is faster and that there are fewer give-up situations compared to the ranked list user interface. In addition, the users showed positive attitudes towards the proposed system.

In a later study (Dumais et al., 2001), seven user interface designs were compared. The conditions included three ranked list layouts and four automatic category based layouts. The user interfaces varied in showing the result summaries and category names. The results indicate that the category based user interfaces are faster than the list based and the best performance was achieved in the condition where the document titles were displayed in the context of the categories. This means that a proper context is needed in order to understand the meaning of a category.

DynaCat

DynaCat is a search system in the medical domain intended for patients searching for information about various medical issues (Pratt & Fagan, 2000). Like SWISH, DynaCat uses the classification approach, but it utilizes multiple models in the process. It models the user's query according to predefined query types and uses a large domain specific terminology model (Medical Subject Headings, MeSH) to classify the retrieved documents. Thus, the category selection is influenced by both the user defined query and the retrieved documents.

The user interface of DynaCat resembles our solution. It lists selectable categories on the left side of the user interface. In contrast to ours, the categorization is hierarchical. DynaCat was evaluated in a user study with 15 participants where it was compared to the ranked results user interface. The results of the study show that the participants found more answers in the given time and that they were more satisfied with the results when using DynaCat.

In summary, we can see similar results in the user studies of these four prototypes (Scatter/Gather, Grouper, SWISH, and DynaCat). All except Scatter/Gather demonstrated faster and more enjoyable user performance in search tasks. The test setups in the studies were similar: the proposed categorization system was compared to a ranked list of results, the *de facto* standard. These results are in line with the cluster hypothesis in the context of search result access. It means that the result categorization is able to bring together relevant documents and help the user in finding the needed information.

4.4 RELATED CLUSTERING SYSTEMS

In addition to the previously discussed research prototypes, there is a large number of systems that are closely related to the current topic. Table 1 lists a selection of such systems including the previously presented most influential systems. The list is not comprehensive, but it gives us an overview of the systems, involvement in the research, the technologies used, and the user interface solutions employed.

The ‘Technology’ column reveals the main technique that is used in the corresponding prototype to create the categorization. The most common techniques include variants of *clustering* and *classification* as well as *term extraction* methods. In addition to these, this sample contains a few systems that employ *Web link structure* analysis.

The ‘Type’ of the categorization or organization refers to the structure that is produced in the organization process and displayed to the user. We assume that the resulting structure may have an important role in the understandability and usefulness of the system for the end users. The structures are either *hierarchical (H)* or *flat (F)*.

The type of user interface (‘UI’) is of great interest to us, because it has such a central position in the end user experience. We speculate that the utility of the most brilliant categorization system may be damaged by a suboptimal user interface design. Categorization of user interfaces is not a simple task, but we try. Table 1 summarizes our categorization principles. We distinguish three types of user interfaces and two target environments (the Web and graphical user interfaces (GUI)). The actual combination of the user interface type and the target environment are represented with the listed combinations of letters.

Finally the ‘Data source’ column tells what kind of data source is used in the prototype. The most important ones include *search results* from a search engine, complete (full text) search result documents (*search docs*), and rich data from a digital library (*DL*).

Table 1. Legend of the user interface types used in Table 2.

UI Type	Description	Web	GUI
Overview+Detail UI	Displays simultaneously an overview of the data and details of the selected item.	O-W	O-G
Browsing UI	A structure is used to navigate in the data collection. The whole structure and/or the data items are not simultaneously visible to the user.	B-W	B-G
Visualizing UI	A visual representation (as opposed to textual) is used to select interesting data items.	V-W	V-G
Multiple techniques	A combination of two or more of the above.	M-W	M-G

Table 2. Research prototypes using categorization in accessing search results.

No	System name	Reference	Technology	Type	UI	Data source
Systems discussed above						
1.	Scatter/Gather	Cutting et al. 1992	clustering	H	OB-G	DL
2.	Groupier	Zamir and Etzioni 1999	clustering	F	B-W	search results
3.	SWISH	Chen and Dumais 2000	classification	H	B-W	search results
4.	DynaCat	Pratt and Fagan 2000	classification	H	O-G	DL (MEDLINE)
Closely related systems						
5.	Adaptive Search	Roussinov and Chen 2001	clustering	F	B-W	search results
6.	AMIT	Wittenburg and Sigman 1997	link structure	H	V-G	web walker
7.	Carrot	Weiss and Stefanowski 2003	clustering	F	B-W	search results
8.	Cat-a-Cone	Hearst and Karadi 1997	classification	H	V-G	DL (MEDLINE)
9.	(CGRU)	Chekuri et al. 1997	classification	F	B-W?	search docs
10.	Cha-Cha	Chen et al. 1999	link structure	H	O-W	intranet search
11.	CI / Meta Spider	Chau et al. 2001	extraction	F	M-G	search results
12.	Dart	Cho and Myaeng 2000	clustering	F	V-W	search results
13.	DisCover	Kummamuru et al. 2004	clustering	H	O-W	search results
14.	HighLight	Wu et al. 2003	extraction	H	O-W	search results
15.	HuddleSearch	Osdin et al. 2002	clustering	H	OB-W	search results
16.	Info Navigator	Carey et al. 2003	clustering + extraction	F/H	V-G	search docs
17.	Interactive Dendrogram	Allen et al. 1993	clustering	H	V-G	DL
18.	iSEARCH	Chen and Chue 2005	clustering + link structure	H	O-W	search docs
19.	J-Walker	Cui and Zaïaine 2001	classification	H	O-W	search results
20.	(KS)	Kules and Shneiderman 2005	classification + clustering	H	O-W	search results
21.	(LC)	Leouski and Croft 1996	clustering	H	B-G	search docs
22.	PHIND	Edgar et al. 2003	extraction	H	B-W	DL
23.	Retriever	Jiang et al. 2000	clustering	F	B-W	search results
24.	SONIA	Sahami et al. 1998	clustering + classification + extraction	F	N/A	DL / search results
25.	WebACE	Boley et al. 1998	clustering	H	O-W	browse history
26.	WebCutter	Maarek et al. 1997	link structure	H	V-G	Guru / Lotus Domino
27.	WebRat	Granizer et al. 2003	clustering	F	V-W	search results
28.	(ZHCMM)	Zeng et al. 2004	extraction	F	O-W	search results

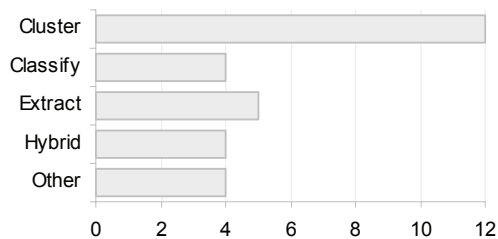


Figure 15. Distribution of the categorization techniques in Table 2.

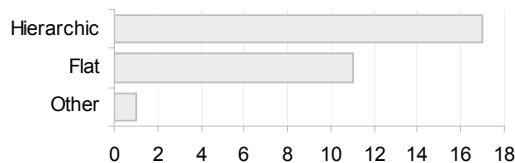


Figure 16. Distribution of the categorization types in Table 2.

To summarize Table 2, we can see that clustering (Figure 15) is the most popular technique in this sample (we assume that this gives a good picture of the overall situation). In addition, the structure is typically presented in a hierarchical structure (Figure 16).

4.5 THE FINDEX SYSTEM

To address our research question on how to enhance the search result access, we have implemented two categorization algorithms for Web search results and designed a filtering user interface for the task. The main idea is to present an overview of the results with automatically computed categories so that different topics contained in the results become visible and easily accessible. Result access is enhanced by the filtering user interface that allows users to select items in the category overview and see the results belonging to the selected category.

Categorization Methods

We have designed and implemented two categorization algorithms. The first, which we call the *statistical* method, aimed at simplicity while the second is a redesign aiming at better descriptiveness of the category names. The second design was inspired by the experiences gained from the first one and is called (*keyword*) *context categories* or *fKWIC* for short. Both categorization systems are based on the word and phrase frequencies found in the search results. In principle, the most frequent words and phrases are used as the categories.

The category computation is based on the textual data available in search engine result listings, i.e. result titles and summaries (snippets). The number of results used in the computation can be adjusted, but we have found about 150–200 results to be a good compromise between thoroughness, simplicity, and computational efficiency.

The categorization process starts with a computation of so-called category *candidates*. In statistical method, the candidates include all individual words and up to one sentence long multi-word phrases found in the result text. Each candidate is associated with a frequency figure, which describes

the number of results the candidate is found in, not the actual word or phrase count. Separately listed stopwords (e.g., articles, pronouns, and the like) are excluded in the candidate extraction process so that they do not appear as candidates or inside candidate phrases. In the candidate computation, the word order of the phrases is meaningful and only phrases with same word order are treated as equal.

The context categorization computes the candidates slightly differently. Context categories are required to contain at least one query term. Thus all the candidates are phrases (at least two words long). The requirement to contain one query term in the candidates reduces the number of valid candidates significantly. Other than this requirement, the candidate computation is similar to the statistical method.

In the early versions of the algorithms we employed a word stemmer for discarding the word endings that cause unwanted variation in words. In computation, simple inflections such as singular and plural forms of a word make them different (e.g., car and cars). However, the stemmer used (by Martin Porter) caused confusion for the end users in some cases and thus, we started to use a simpler non-exact string matching algorithm (for details see Paper VI). The effect of the algorithm is similar to stemming algorithms.

After the candidate extraction the actual categories are selected. This phase is important in contributing to the quality of the categories. The process includes merging the candidates that are considered to be the same and removing the candidates that are sub-phrases of one another. The selection process is slightly different in these two categorization methods and the details can be found in Paper IV. The main point in the selection process is to select highly descriptive (understandable for humans) categories while ensuring appropriate coverage of the results. In the end, n most frequent candidates are selected to be displayed to the user. Our study (Paper III) indicates that this number should be between 10 and 20. In our experiments, we used 15 categories.

The final categories are carefully selected words or phrases from the search results. These categories contain all the results where the word or phrase occurs. Due to merging of the candidates, the words in the phrase categories may appear in different order in the results or words may not be strictly sequential but may have stopwords in between. Other than such exceptions, the mapping between the category name and the associated results is straightforward. In fact, the categories can be seen as ready-made free text search queries for the result set.

User Interface

The user interface design follows the popular overview and details model (Card et al., 1999, pp. 285–286) and is divided into two panels. The left

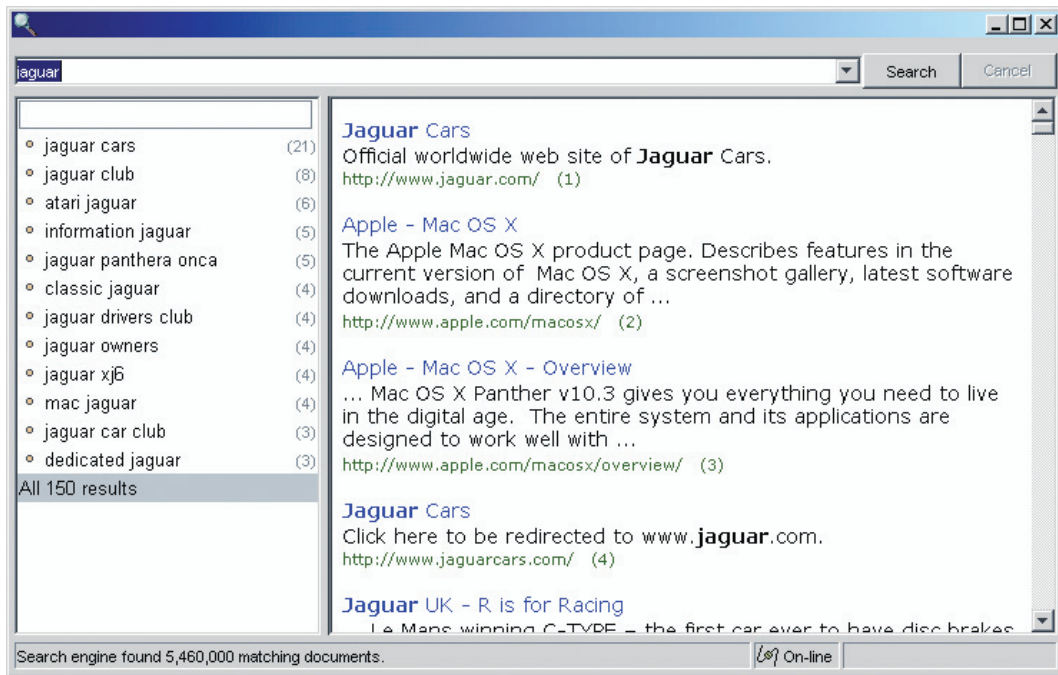


Figure 17. Findex standalone user interface with built-in 'All results' category selected.

contains the list of categories (overview) and the right shows the actual results. The user interface has been implemented both as a standalone graphical user interface application (Figure 17) and as a Web service (Figure 18). In both cases, the basic structure and functionality of the user interface are the same. The graphical application was used extensively in our experiments (it allows comprehensive logging) and the Web user interface was targeted for our longitudinal study to make the service easily accessible.

The selected user interface model was derived from our design approach where the aim is to provide new ways of accessing search results. This means that the new features are *added* to the current user interfaces so that the users can take advantage of their existing knowledge with them. This allows users also to ignore the new features when desired.

To enable this, our interface has a built-in 'All results' category. The user interface functions so that this category is automatically selected after each search. When the 'All results' category is selected, the conventional list of ranked results is displayed. This makes the user interface appear like any other Web search engine.

When a category is selected, the result listing is filtered to show only those results belonging to the category. Our categorization schemas are straightforward: a result belongs to a category if it contains the name of the category in its result summary text. This fact is illustrated to the user by highlighting the corresponding text in the result listing (Figure 18).

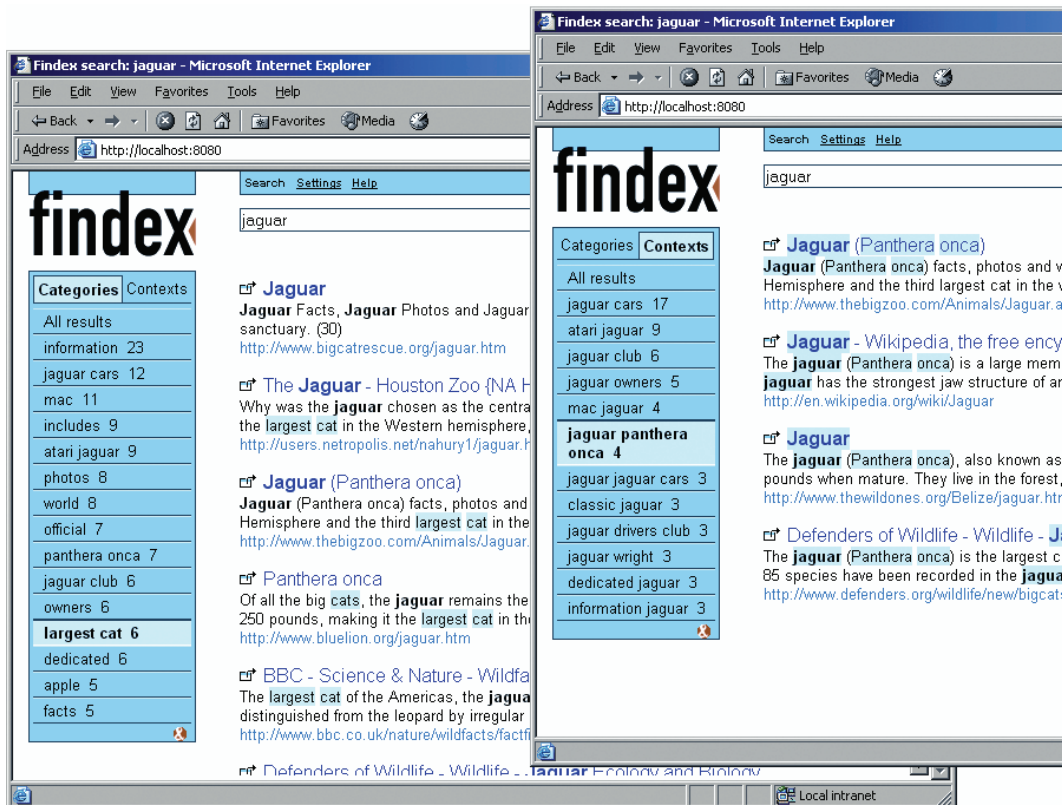


Figure 18. Findex web user interface. The larger image shows statistical categories, the smaller context categories. Highlighting shows the relationship between a result and the selected category.

In the latest Web user interface, the two categorization methods are visible and controllable by the user (this was not the case in our longitudinal study). On top of the category box on the left (Figure 18), there are two tabs labeled 'Categories' and 'Contexts' for statistical and context categories respectively. By selecting a tab, the user can control the type of categories displayed in the overview.

Differences from Related Systems

Because enhancing search result access by categorization has been under extensive research, the obvious question arises: what is the contribution of the present research?

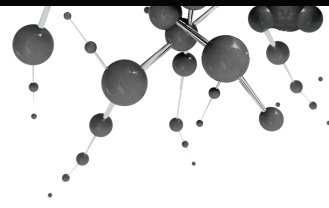
The differences and thus the contributions of this study are threefold. One aspect is the actual algorithms used to categorize the results, another is the combination of the algorithms and the user interface, and the third is the evaluation approach. The following summarizes our contribution in relation to the other systems:

1. The two categorizing algorithms we present are novel and designed especially for Web search engine results consisting of short text summaries. The algorithms are based on a similar term (phrase) extraction technique used in the STC algorithm by Zamir and Etzioni

(1998) and no document similarity measures are used. In contrast to Zamir and Etzioni, we do not merge clusters based on the documents they contain, but based on the similarity of the extracted phrases. This appears to produce understandable results.

2. The filtering user interface in combination with the type of categorizing algorithms we use is new. The Grouper user interface forced the users first to select a category and the results were displayed only on the next page. Our user interface treats categories as an added convenience that is provided in addition to the results. This allows the users take advantage of result ordering when it works, but gives them additional means of exploring the results when needed. DynaCat was similar in this respect, but the categorization method and data source were different.
3. The selection of the evaluation methods is unique, giving new insight about how search result categorization is used. Our approach combined experiments and longitudinal studies with the same system. In related research theoretical or mathematical evaluations are common, but our methods involve end users closely in the evaluation process.

In summary, our algorithms are unique, but not radically different from previous work. The user interface idea has also been presented by others, but the combination of them and the thorough evaluation with end users constitutes the contribution of this research.



5 Methodology

5.1 CONSTRUCTIVE APPROACH

Studying human-computer interaction often involves the construction of a software artifact that implements an interesting design idea. The artifact demonstrates the potential of the idea and makes its evaluation possible. The evaluation enables us to gather valuable information about the solution.

Building better ways of accessing Web search results is an activity where such a constructive research is valuable. It is impossible to evaluate the importance and the functionality of the design ideas without a working prototype. For example, it is easy to imagine a system with a perfect categorization system for intuitive representation of the information. However, it is difficult to build such a system, which is why we do not have them. A constructive approach makes the elimination of infeasible ideas clear and concrete.

The implementation of this study contained multiple stages of constructive work. The construction and the experiences from the evaluations taught us valuable lessons that are incorporated into the process in subsequent implementation phases. In our methodology, the major constructive phases were followed by an evaluation to enable such feedback.

5.2 MEASURING THE USE OF SEARCH INTERFACES

The selection of evaluation techniques for search interfaces is not a straightforward matter as one can follow at least the examples of HCI and IR research. The choice of methods depends on the research question. We

will now discuss the properties of the methods found in those fields and justify our selection.

The core measures in HCI are stated in the ISO 9241-11 (1998) standard. They are *effectiveness*, *efficiency*, and *subjective satisfaction*. Effectiveness measures the completeness and thoroughness of task completion. In the case of information search it means, for example, the number of found (relevant) documents and their coverage in relation to the given task. Efficiency, on the other hand, describes the value of the results achieved in relation to resources used (such as time or money). In information search tasks, this typically means the time used for accomplishing the task or the number of result documents opened for evaluation. Subjective satisfaction is usually evaluated with questionnaires eliciting users' opinions about the system.

The HCI measures are well suited in our situation where we are interested in the users' performance, but they are so general that they cannot be measured directly. There is a lot of room for interpretation as to what the measures actually mean. The evaluator must decide what the individual measures (effectiveness, efficiency and subjective satisfaction) mean in the application domain being studied.

The approach for evaluating search systems in the information retrieval community is different. The most fundamental measures are *recall* and *precision*. Recall describes the thoroughness of a search. It is presented with a number that states the proportion of the relevant documents retrieved to all the relevant documents in the collection. Precision, on the other hand, denotes the number of relevant documents within the result set. The greater the precision, the fewer irrelevant documents there are to distract the user in the result evaluation. Both these measures are stated as percentages.

The above-mentioned measures of recall and precision do not depend on the user interaction with the system. The measures are calculated based solely on the result set the search system returns. This is appropriate when the properties of the retrieval engine are studied, but if we are interested in how the user can evaluate the result listing, we need a different approach.

Another issue with the recall measure is that the measure is hard to calculate in the Web environment. For computing recall for a query, the total number of relevant documents in the collection (the Web) should be known. This is feasible only in limited collections such as those provided in TREC (Text Retrieval Conference).

Veerasamy and his colleagues (Veerasamy & Belkin, 1996; Veerasamy & Heikes, 1997) used slightly modified measures in a study on a graphical

user interface for accessing the search results. The measures are related to recall and precision, but they are based on the document selections made by the users. The measures are called *interactive recall* and *interactive precision*. Interactive recall indicates the percentage of the relevant documents in the result set that were selected by the user. Interactive precision indicates the proportion of relevant documents within the user selected documents.

We adopted these measures in our experiments. In the studies, we refer to them simply as recall and precision as the meaning of the measures is obvious in the context. In addition, the word 'interactive' seems inappropriate in the context of HCI studies. Interactivity is such a central concept in HCI that using it to describe a measure seems confusing.

5.3 CONTRIBUTED MEASURES

In addition to these well established measures, we developed a few measures of our own for the studies. In the first study, it became apparent that measures typically used in HCI studies, like time and success, may not be enough in studying search user interfaces. To alleviate the problem, we developed three new measures for the evaluation of interaction with and usability of search user interfaces. The measures are specially targeted at studying the result evaluation phase of the search process and they are presented in Paper II.

The first suggested measure is *search speed*, that is measured in answers per minute. The measure is analogous with physical speed like kilometers per hour. The second measure is closely related and adds a quality dimension to the measure. *Qualified search speed* states how fast the user can find results of certain relevance, for example, how many *relevant* documents the user is able to gather in a minute. One important property of these measures is that they are proportional making the comparison of the results slightly easier.

The third new measure is *immediate accuracy* that captures the success in typical Web search tasks. Web searchers select commonly only one or two results for each query (Spink et al., 2002). In such a situation, the limiting resource is not time, but rather the number of result selections. It matters how many result selections (clicks) the user needs in order to find the first relevant document for the information need. This is exactly what the immediate accuracy measures. It states the percentage of cases where at least one relevant document is found by the n^{th} document selection.

These three new measures were utilized in appropriate places throughout the individual studies of the thesis. They address the problem noted earlier about the lack of concreteness in HCI measures.

5.4 EXPERIMENTAL DESIGN

The measures for evaluating the success of a design are of great importance, but the experimental design of the evaluation is not self-evident either. In our case, the independent variable is clear: the user interface. Although we had two categorization algorithms available, each experiment provided the participants with only one. Thus the user interface and the categorization algorithm were treated as one experimental variable. In each experiment the independent variable had two values: suggested user interface and the baseline user interface. As the baseline user interface, we used a Google interface imitation that displayed Google results in the original order, ten results per result page.

In addition to the independent variable, the actual experiment situation and its constraints play a major role. To obtain reliable results we tested multiple experimental settings. First, we aimed to maximize the external validity by emphasizing the naturalness of the situation. We simply provided the participants with search tasks and let them do the searches as they wished. We controlled the tasks, user interfaces, and topical knowledge (using students from a particular class and tasks related to the topics of the class), but not the search behavior. We treated task completion times as dependent measures and logged the selected results.

Such a test setup did not yield any interesting information about the phenomena that we were interested in (accessing the search results). By looking at the collected data, we saw that the participants did not utilize the new user interface features (categories) and that most of the time was spent evaluating the actual documents accessed through the search result list. This was undesirable for our purposes and compromised the validity of the results by introducing a lot of noise into the data.

In the second step, we added more control to the setup. We addressed the problems by not allowing the participants to open the result documents and requiring them to use the categories in the category condition. The latter was achieved simply by disabling the automatic selection of the 'All results' built-in category. Normally, the selection of this category makes the category system appear almost like the normal ranked result list, because all the results are immediately shown to the user. However, the exciting situation of being in an experiment (although participants were explicitly told that the user interface is the target of the study, not the participants) was likely to cause the participants of our first test to follow the familiar way of accomplishing the task. That is, using the ranked results. Exciting or stressful situation may impair the human performance and cause so-called tunnel vision, referring to the narrowing of the useful field of view (UFOV) (Matthews et al., 2000, pp. 164–165; Ware, 2004, p. 147).

With these refinements we conducted another pilot study. Again we saw that the level of control was still insufficient. The collected data contained such a wide variation that it was not possible to measure the effects created by different user interfaces in the result evaluation phase of the search process. By looking at the data we concluded that the variation was caused by the differing query formulation skills of the participants.

The third approach was adopted from the literature where the queries for each task were predefined by the experimenters (Chen & Dumais, 2000). In information search tasks, this is a fairly radical solution, but the focus of our research allowed this. Because the focus was on understanding the effects of the user interface on the *result evaluation* phase, controlling the query formulation phase did not invalidate the measurements. This setup allowed us to measure the effects of the variation in the user interface properly. The actual tasks and the associated predefined queries can be seen in Appendix 1.

We considered this issue from the point of view of internal and external validity. Increasing the control in the test situation increases the internal validity of the setup at the expense of external validity. Because we increased control only in the phases of the process not included in the interesting phenomena, we concluded that the external validity was not compromised too much.

5.5 TASKS

The early pilot tests were based on fact finding tasks where it was enough to find one document that contained the answer to the question. In the course of pilot testing it became evident that such a task type may be a possible source of misleading results. It is rare that result categorization helps users in fact finding tasks. As the clustering hypothesis suggests, categories bring similar documents together and thus provides the users with a more comprehensive set of results on the desired topic. Because this is the main area of contribution of our proposed solution, the type of tasks should reflect this fact.

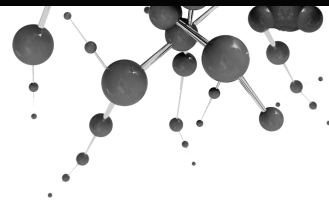
To alleviate the issue, we reformulated search tasks by requiring users to collect as many documents as possible for a given task. This aims to mimic a certain type of searches that users engage in regularly in the Web. The Web search types have been classified by Broder (2002) and Rose and Levinson (2004). According to their taxonomies, multiple results are often helpful in informational (in particular undirected informational) searches. Rose and Levinson give an example of an undirected informational query: 'color blindness'. Such a query aims to cover a broad topic (a phenomenon) and multiple result documents can help users in achieving the understanding of it.

In normal settings, Web searchers are not simply searching for as many documents for a task as they can. There is a fine balance between the thoroughness and the time spent in searching the documents. There are many factors affecting this balance, which we cannot properly control. To simulate the balance, we asked the participants in a pilot study to carry out the task as fast as possible and with the thoroughness they felt appropriate. The inclusion of subjective judgment turned out to be a mistake. Participants favored thoroughness excessively over time. In practice, they could evaluate all 150 results that we use for categorization searching for the relevant answers.

This is clearly not normal behavior, as other studies report that users typically consider fewer than 30 results per search in about 80% of the searches (Jansen et al., 2000; Hoelscher, 1998). We concluded that the somewhat artificial experiment situation affects the participants' performance and encourages them to carry out the tasks with exceptional thoroughness. To compensate for this we chose to impose a time limit for the tasks to simulate normal behavior. A one-minute limit was seen and pilot tested to be an appropriate limit. It allows moderate thoroughness while still being short for the participants to be overly accurate in the task.

Our one-minute time limit is supported by the figures reported by Aula and Nordhausen (forthcoming). Their figures indicate that Web searchers use about 1.5 minutes for evaluating the result listing of a query. Completing a search task took 5.5 minutes in their study and it consisted of multiple queries and evaluation of the actual result pages (57% of the time). Our time limit is shorter, but the experimental setting focusing only on the result evaluation compensates for this. In a normal situation (such as that in Aula's and Nordhausen's study), user's attention must shift from evaluating the result listing to evaluating the result documents and back, but in our tests, this did not happen. This reduces the required time for evaluating the results in our experiments.

After the fact, we can see that our participants saw on average 3.6 result pages (with ten result per page) while using the reference user interface. This is consistent with 30 evaluated results reported earlier (Jansen et al., 2000; Hoelscher, 1998).



6 Studies

6.1 OVERVIEW OF THE STUDIES

Figure 19 illustrates the research process and displays the temporal relationships between the phases. The starting point for the studies is the search framework that enables us to formulate queries, execute them, and to categorize the results. Implementing such a framework was the first major constructive part of the research and produced the Findex search user interface. The development of the first categorization algorithm was a part of this work.

The first experimental study was designed to evaluate the effectiveness and usefulness of the statistical categorization approach. Because the measuring practices for evaluating the search user interfaces were somewhat limited, we developed new measures. The results of the first experiment were used in testing the new measures along with the results found in the literature.

As the first experiment indicated the utility of our categorization approach, we looked deeper into the system. In the next phase, we studied the effect of the number of categories presented to the user. We learned that relatively few categories yield a better performance.

The next step was to address the question of the external validity of the studies. Initial studies were carried out in a laboratory and little was known about the use of the system in real settings. This was addressed by a longitudinal study where 16 participants were allowed to use the system for an extended period of time. Before the study, a new Web based interface for Findex was implemented.

The experiences from the work with the first categorization scheme gave us valuable insights. We noted that good quality categories (category names) tend to contain a query word in them. This observation was then implemented in a working prototype with inspiration from keyword-in-context (KWIC) indices. After this construction phase, the new solution was integrated into the Findex user interface and evaluated in an experiment.

The final study was concerned with the properties of the two categorization algorithms. It described the details of the algorithms for future development and presented various result of their performance.

6.2 STUDY I: EXPERIMENT OF STATISTICAL CATEGORIES

Reference

Mika Käki and Anne Aula (2005). Findex: improving search result use through automatic filtering categories. *Interacting with Computers*. Elsevier, Volume 17, Issue 2, pages 187–206. (Paper I, page 85)

Objective

The aim of the first study was to directly contribute to the main issue of the thesis: enhancing search result access. The testing phase was preceded by design and implementation of the first categorization scheme that aimed to enhance the users in accessing the search results.

The first categorization scheme (statistical algorithm) was initially based simply on word frequencies. The simple approach was strongly motivated by observations from previous clustering systems that appeared incomprehensible to the end users who are unaware of the underlying technology. The first approach of using single words was found to be too restrictive. Although single words may be descriptive, the inclusion of multi-worded category names (phrases) seemed appropriate. Because the logic of selection is the same for words and phrases, the addition did not complicate the system much.

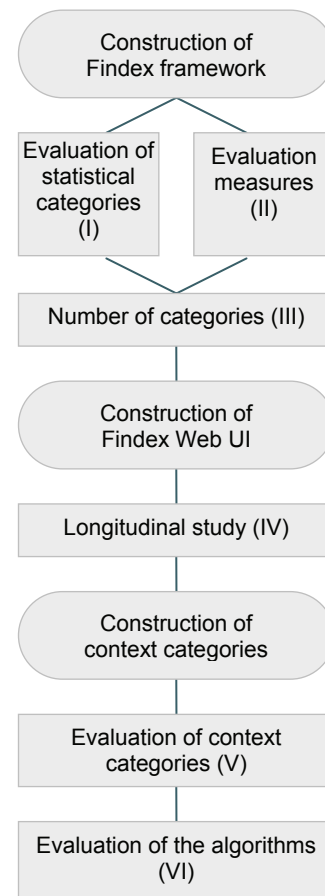


Figure 19. The main phases of the research process. Ovals denote construction and rectangles evaluation.

One important design decision was the selection of the user interface model. A two-piece user interface that shows an overview on the left and contents of the selected item on the right is a widely used solution. Overview and details type user interfaces have been popular in the research literature and they have been shown to be beneficial for the users. In addition, this type of user interface allowed the users to take advantage of the ranked result listing when it is profitable. Our aim was to provide extra tools for interacting with the search results and thus this solution was a good match with the objectives.

The question for the first experiment was if the automatic categories are beneficial for the end user or not. It played an important role for the entire research project. The result was an important indication that our approach worked and that it was worth exploring further. The answer was sought via an experiment where the new category solution was compared to the ranked list (de facto standard) approach. We recruited 20 participants for the controlled study that was carried out in a laboratory environment.

Results and Discussion

The results indicate the success of the selected user interface and categorization scheme. The participants were able to locate the relevant results up to 40% faster with the new user interface. In addition, participants were 21% more accurate (in terms of relevant results) and they showed positive attitudes towards the proposed system.

The results were positive for our research. They showed that we were on the right track in the pursuit of making searching easier. Thus the first and most important conclusion was to carry on studying the techniques. In addition, the performance benefit was fairly high in our experimental setting. The results can also be seen to be in line with the cluster hypothesis and the assumptions about the profitability of an overview + detail type of user interface.

The study also raised a number of questions, such as the number of categories to present to the user and the ability to generalize the results in other situations. These issues were addressed in the subsequent studies.

6.3 STUDY II: SEARCH USER INTERFACE EVALUATION MEASURES

Reference

Mika Käki (2005). Proportional search interface usability measures. In *Proceedings of NordiCHI 2004 (Tampere, Finland), 23–27 October 2004*. ACM Press, pages 365–372. (Paper II, page 107)

Objective

In the course of conducting and analyzing the results of the first study, we discovered a lack of descriptive measures for our needs. In some well defined areas in HCI there are commonly established measures for evaluating the success of user interface solutions. For example, in text entry studies measures like keystrokes per character (KSPC) or error rates are routinely used (MacKenzie, 2002; Soukoreff & MacKenzie, 2003). The same cannot be said about evaluating search user interfaces and the aim of the second study was to provide new, useful measures.

The data from the first study were available for experimenting with the new measures. From the literature review it became apparent that presenting raw numbers on the amount of time spent and the number of results gathered were popular measures in search user interface studies. Such measures capture important properties of the measured systems, but the results are hard to interpret and compare.

Based on the literature review and the experiences from our first study, we set two goals for the new measurements. First, the new measures must make comparisons and understanding of the results easier. Second, they must capture the special characteristics of Web searching. In particular, it is common that Web searchers stop the search process when one or two good enough answers are found. None of the previously used measures capture the success in such a situation.

Results and Discussion

The results of the study include three new measures for evaluating search user interfaces, which were evaluated by applying them to the results of previous studies (our own and those found in the literature). The first two are designed to make the results easier to understand and to compare. *Search speed* and *qualified speed* measures are proportional measures for describing how fast the search user interface is. Search speed is a simpler version that describes a raw measure without considering the quality of the results while qualified speed employs accuracy information. Both of these measures are stated in *answers per minute* (APM). Accuracy information in qualified speed adds an extra modifier to the measure by giving, when it states, e.g., the number of *relevant* answers per minute.

The third measure addresses special characteristics of Web search behavior. *Immediate accuracy* is the proportion of the cases where at least one relevant answer is found by n^{th} result selection. This aims to capture the success in typical rather impatient Web search behavior.

The evaluation of the measures was based on applying them to the data of the first study as well as to the data reported in one of the Scatter/Gather studies. In the comparison, we showed that these measures can separate systems and that they are easier to compare than the old ones. The

conclusion was that the new measures are useful additions to the toolbox of the search user interface evaluator and we employed these measures in the later studies where appropriate.

There are, however, a few problematic issues in the results of this study. The evaluation method for the new measures is not clear because there are no widely accepted ways of demonstrating their utility. Conducting a conventional experiment is problematic, because the measures largely constitute the experiment; the result of an experiment is expressed by them. Thus, the evaluation of the measures must be largely grounded on the intuition about their descriptiveness. However, this does not mean that the experiment is futile in evaluating new measures. It plays an important role in forming an impression of the descriptiveness and utility of the measure.

Second, the applicability of the measures can be limited. Our own need implies an emphasis on the result evaluation phase and the measures reflect the fact. For example, it is not clear if they can be used in evaluating the utility of query reformulation aids. However, we think that the applicability is not seriously compromised. For example, a system with novel query refinement aids can be evaluated with speed measures with a different test setup where the user has an opportunity to make multiple queries. If the query refinement aids work, the effect should be measurable in the users' ability to find meaningful results, for example in qualified speed.

6.4 STUDY III: THE EFFECT OF THE NUMBER OF CATEGORIES

Reference

Mika Käkki (2005). Optimizing the number of search result categories. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2005 (Portland, USA), April 2005*. ACM Press, pages 1517–1520. (Paper III, page 117)

Objective

The success of the first evaluation of the Findex search user interface encouraged us to look deeper into the phenomena of using categories as the result list overview. Because the overall objective is to enhance the user's performance, an obvious question is how the number of categories presented to the user affects it. In other words, what is the optimal number of categories? In the first experiment, the number of categories was somewhat randomly chosen (fifteen), based on our intuitive conception of the performance of the system.

We first tried to find the answer from previous menu selection studies, but they were not quite on target. The automatically computed categories are

more of a moving target and thus the users' evaluation process of them may be considerably different from the search of menu items. Notably, meaningful ordering and grouping are not possible with automatically formed categories, and this changing nature complicates the situation. Thus, we decided to investigate the issue in a new study.

The experiment compared three conditions: 10, 20, and 40 categories while the other parts of the user interface were constant. The controlled study was carried out with 27 participants in a laboratory. The test setup was similar to that of the first experiment. The setup was seen to be robust and appropriate also for the current problem.

Results and Discussion

The results of the experiment showed that fewer categories are better, but the measured differences between the conditions were relatively small. The subjective opinions were clearly against many categories and the participants found 40 categories to be clearly too many. Although 20 categories also received negative subjective feedback, the objective measures could reveal only small or no differences in the level of performance in comparison to 10 categories. In the end, our original estimate of 15 categories turned out to be fairly good.

As the main result indicates that fewer categories results in better performance the question about fewer than 10 categories readily arises. Unfortunately, the condition with fewer than 10 categories had to be excluded from the study because of practical reasons. Increasing the number of conditions increases the need for participants and thus the need for time. We simply did not have all this available. However, we do know from the first study that zero categories results in poor performance. In addition, with fewer than 10 categories, it is probable that the categories would not support the user's task.

One of the practical conclusions from this study was that the users may need a way to control the number of categories presented in the list. A number around 10 or 15 seems appropriate for the default setting, but it does make sense to let users control the number of categories to a certain extent. Indeed, such functionality is implemented, for example, in the Vivísimo search engine, where the user can get more categories on demand.

6.5 STUDY IV: LONGITUDINAL STUDY OF FINDEX

Reference

Mika Käki (2005). Findex: search result categories help users when document ranking fails. In *Proceedings of the SIGCHI Conference on Human*

Objective

In the third phase of the studies, we turned our attention to the issue of internal and external validity. As the previous experimental settings were strictly controlled, some questions were unanswered.

First, already from the first experiment we already knew that categories are not beneficial in all situations, but the frequency of such cases is unknown. This is the case, because we controlled the query formulations and thus the distribution of the tasks. Thus, the query formulations in the experiments do not necessarily comply with real use. Second, the users' actual use habits are not known from laboratory experiments. The experimental setting forced the participants to use the categories at least once for each task, but in a real situation there is no such constraint.

To address these issues, we conducted a longitudinal study. We implemented a Web-based version of our search user interface and recruited 16 participants from Finnish universities. Universities were used as the recruitment source because we wanted to involve users who are fairly active Web searchers and university personnel were assumed to need information frequently in their work. The study was carried out during the summer and we collected usage information on two months of use. To compensate for the vacations, the system was available for the participants for three months. All interaction with the system was logged and the behavior of the participants was not restricted in any way. In fact, they were encouraged to use the system any way they saw appropriate.

Results and Discussion

The results of the study showed that the utility of the categories is a more complex matter than the first experiment suggested. In the controlled setup, the participants were required to use the categories at least once in a task, but in the real situation the categories were used on average in every fourth query. Although this may seem little, we find it encouraging. The categories were used regularly over a long period of time, indicating their consistent ability to help users in certain situations.

By examining the log files we concluded that the categories are most likely used in situations where the result ranking does not support the user's task. The time required to select the first result is about twice as long as the access time when categories are not used. This means that the users have time to first read a screen full of results and evaluate them. If this does not produce results, they scan the short category list, select a category and evaluate few results in the category before opening a result page.

Although the user behavior could be seen as a disappointment regarding the categories, it can also be seen in a positive light. This scenario means that the users can utilize their old search habits when working with the new user interface. Users can exploit the success of the rank ordering, when possible, while categories help them in problem situations. As categories are used regularly, we believe that there is a need for categories and they are employed as part of search habits. Thus it seems that categories are beneficial in real settings.

We consider the results of the study to be fairly strong despite the shortcomings of the test setup. The longitudinal studies are often loosely controlled, which was also the case here. However, in our case we can combine the results with the results from the experiments with the same system. We know from the experiment that the use of categories enhances performance. It means, among other things, that users tend to select meaningful categories. From the longitudinal study we know that categories are used regularly. Because the use of categories does not diminish over time, the category selections are likely also to be beneficial in real settings.

One interesting question that we were not able to address in the given time frame concerns the usage patterns. In particular, it would be interesting to know in what kind of situations the categories are used. For example, one might assume that they are used in the query refinement phase, when the user is experiencing difficulties in formulating the query. Our data could provide insight into this question, and this is obviously an interesting topic for future studies.

6.6 STUDY V: EXPERIMENT WITH CONTEXT CATEGORIES

Reference

Mika Käki (forthcoming). fKWIC: frequency based keyword-in-context index for filtering web search results. Accepted for publication in *Journal of the American Society for Information Science and Technology*. Wiley. (Paper V, page 135)

Objective

In the course of using and testing the first version of Findex we noted that the most meaningful categories tended to contain query terms in their names. This gave rise to an association with keyword-in-context (KWIC) indices and led to the idea of displaying the most frequent keyword contexts as an index to the results.

The implementation of this fKWIC indexing system proved to be notably different from the initial categorization scheme. The development was an iterative process where design and implementation were followed by an

informal evaluation. The requirement of having query terms in the category names posed new challenges as the number of category candidates was reduced and the algorithm for removing and merging similar candidates was changed considerably compared to the initial categorization algorithm.

Because the categorization algorithm changed so much, it was not clear if the new approach would be beneficial for the user. Thus the objective of this study was to ascertain if this new categorization algorithm enhances the user performance. We conducted a controlled experiment with 36 participants in a usability laboratory. The new system was compared to the ranked list user interface (baseline solution). The setup was largely the same as in the first experiment, because the research question is basically the same and the setup was seen to be sound.

Results and Discussion

The results confirmed the utility of this new approach. The results showed a 29% increase in the speed of finding relevant results and the proportion of relevant results among the selected results increased by 19%. In addition to these objective measures, we obtained evidence about positive attitudes towards the proposed user interface. These facts support the hypothesis that the proposed system enhances the users' performance in accessing the search results.

Due to slight changes in the test setup and the demographics of the participants, the results are not exactly comparable with the first experiment. However, it is fairly safe to say that the performance of the systems is at about the same level. Based on this study, we cannot say which of these systems is better. Even if we could, the difference would probably be fairly small.

Although the comparison of the systems would be interesting, we abandoned this approach considering it too radical for these prototype systems. The results of such a study could be too largely influenced by small design or implementation flaws and thus lead to false conclusions. Small differences in the system performance could be exaggerated in a comparison setup. Instead, we judged that the most important point was to establish a relation between the new and the currently dominant systems.

6.7 STUDY VI: EVALUATION OF THE CATEGORIZATION ALGORITHMS

Reference

Mika Käki (2005). Findex: properties of two web search result categorizing algorithms. Accepted for publication in *Proceedings of the IADIS*

International Conference on World Wide Web/Internet (Lisbon, Portugal), October 2005. IADIS Press, pages 93–100. (Paper VI, page 153)

Objective

The first five papers had a strong human-computer interaction bias in their research methodology and approach. Because the topic of the research is situated at the intersection of HCI and IR, we adopted a more IR-oriented method for this study. In our previous publications, the fine details of the categorization algorithms were left slightly fuzzy and the computational performance was largely uncovered. In addition, the relationship between the two categorization systems was unclear. Both systems are beneficial for the users, but intuitive experience indicated that both algorithms may have situations where they perform better than the other.

To address these needs, we performed a study on the algorithms. The study involved mathematical measures such as coverage, overlap, recall, and precision of the category algorithms. A heuristic evaluation was included to identify the criteria for the situations where the algorithms work best. In addition, the algorithm descriptions were published to ensure that the acquired information can be utilized in future research.

Results and Discussion

The results of the study revealed benefits and downsides in both algorithms. Both were seen to deliver acceptable computational performance, given that the current implementations are not highly optimized. The first categorization algorithm performed better with respect to ensuring the coverage and overlap of the categories. Context categories (*fKWIC*), in contrast, were strong on measures involving the quality dimension, but were not able to cover as large part of the results. This supports our hypothesis that there are differences between the methods.

The heuristic assessment revealed situations where the categories were successful and unsuccessful for both algorithms. Typically a situation that is hard for one algorithm is not as difficult for the other. Thus, it could be possible to compensate the flaws in one algorithm by a reasonable selection of the used categorization method. Such work is left for the future.

6.8 DIVISION OF LABOR

One of the publications mentions a co-author. The first paper was done in collaboration with Anne Aula, whose contribution for the whole system is important. The central ideas behind the categorization approach were developed together with her. In addition, Ms. Aula had central role in the

design of the experimental setting and in conducting the pilot studies in which the settings were tried out. The experiment reported in the paper was carried out by myself. The paper was mostly written by me and Ms. Aula had an important role in commenting it.

Although the other papers do not mention co-authors, it does not mean that they were made in isolation. Colleagues have contributed countless ideas and comments for each of the papers. However, all the experiments, software artifacts, and original text for the papers were produced by the present author.



7 Conclusions

We have presented the Findex Web search user interface concept consisting of user interface functionality and two novel result categorization schemes. The categorization approach for accessing the search results was evaluated in four user studies and in one theoretical study. In addition, we presented three new measures to be used in the evaluation of search user interfaces with user studies.

The contribution of the work is two-fold: 1) a search user interface concept (user interface functionality and the categorization schemes) and 2) new information for the scientific community about the usefulness of categorizing search user interfaces. The latter is the main contribution of the work.

The following lists the conclusions from each of the studies:

1. The first study (Paper I) evaluated the basic categorization approach and the statistical categorization algorithm in particular. The study shows that the approach is beneficial as users were 40% faster in finding relevant results and the relevance of initial selections is higher. We can conclude that the approach increases users' performance in certain conditions (such as those used in the experiment). However, the experimental setup leaves us in the dark as to how useful the system would be in a normal use situation. Given the search queries that were formulated for the participants and search tasks not initiated by the searchers, the generalization of the results cannot be entirely taken for granted. This issue was addressed in the longitudinal study (Paper IV).
2. The second experiment (Paper III) studied the effect of the number of categories on the user performance. The main conclusion is that fewer

categories result in a slightly better user performance. However, the performance penalty with more categories is not great leaving some room for new designs. In practice, 10–20 categories appears to be an acceptable number. Note that the categorization algorithm has an affect on the quality and the coverage of the categories and thus on the user performance. We assume that our results can be used as guidelines, but new categorization algorithms may require new studies.

3. The third study (Paper IV) addressed the issue of using the categorization system in a normal situation. This longitudinal study shows that categories become a part of users' search habits and that they are used in roughly in every fourth query. We conclude that users can see the benefit of the categories in normal situations and that they can take advantage of them. Unfortunately, we were not able to (due to time limits) analyze the use situations and use patterns related to the category use. This would be an interesting analysis and is left for future studies.
4. The experiment on context categories (Paper V) compared our second categorization algorithm to the *de facto* standard ranked results list user interface. Results show that the context categories increase users' speed of finding relevant results and their accuracy in selecting meaningful results. We conclude that this alternative categorization algorithm is a viable solution and enhances the users' performance. Being a laboratory experiment the study faces the challenges of external validity. However, the use of context categories is much like the use of statistical categories that were seen to perform well in the longitudinal study. We assume that this is also the case with context categories.
5. Theoretical evaluation of the categorization algorithms (Paper VI) studied the properties of them. Based on the evaluation, the computational performance of the algorithms is acceptable. The quality of the categories depends on the underlying result set and both algorithms have strengths and weaknesses. We conclude that the algorithms are a good starting point and provide benefits as they are. However, there is room for improvement and the algorithms should not be considered to be finalized products.
6. Three new search user interface performance measures were proposed in Paper II. The measures were used in multiple experiments during the studies. The measures reveal interesting details and differences in the user interfaces. We conclude that the measures are applicable in studying search user interfaces. However, some assumptions the measures make may limit their applicability. For example, the application of the immediate accuracy assumes multiple result selections, which is not always achievable in a test setup.

In summary, result categorization enhances users' performance. The degree of advantage depends on the query, the user's information need, and the results returned by the underlying search engine. Categories are not needed when the result ranking supports the user's information need. If the top of the result list does not provide relevant results, the users cope with the situation using the categories.

Although we saw that our categorization algorithms perform acceptably according to multiple measures in the computational evaluation, large scale applications are not simple. If we consider a commercial search engine such as Google that processes hundreds of millions of searches a day, the performance requirements are enormous. Assuming 200 million queries a day and an extra load of 200 milliseconds per query for the categorization, we are facing over a year of computation each day. The cost of implementing such a system is obviously high. Although this simple calculation suggests problems in scaling the system, we do not have all the information to draw firm conclusions. Our studies did not contain in-depth performance examinations in terms of processor and memory resources. It is possible that these problems can be solved or reduced easily.

The large scale Web searches, however, are only one application domain. The techniques can surely be applied in other environments such as intranet searches or other search facilities that utilize the processing power of the local computer. We expect the solution to be easily applicable in such cases.

Since the work for this study commenced, new methods have been published that aim to increase the quality of the categories. The results of Zeng and colleagues (Zeng et al, 2004) are especially promising. Their technology is based on features assigned to the candidate categories and learning methods in selecting appropriate weights for the features. Although this complicates the selection process of the categories, it does not reveal the complexity to the end users because categories are still simply words or phrases appearing in the results. This kind of approach is desirable from our premises, where the comprehensibility for the end user is vital.

Improving the quality of the cluster names is the most important area of future work for our system. The complete removal of stop words from the final category names may not be the optimal solution, although it is efficient in certain situations. Another issue concerns uninformative words that are not stopwords, such as 'information' or 'world'. In some contexts they can be meaningful, but not generally. Perhaps feature based measures (such as TFIDF) on the word significance could solve some of these problems, as Zeng and colleagues have demonstrated.



8 References

- Allen, Obry, & Littman (1993): An Interface for Navigating Clustered Document Sets Returned by Queries. In *Proceedings of the Conference on Organizational Computing Systems, COOCS'93 (Milpitas, USA)*. ACM Press, pp. 166-171.
- Anick, P. (2003). Using Terminological Feedback for Web Search Refinement - A Log-based Study. In *Proceedings of the Annual International ACM/SIGIR'03 Conference (Toronto, Canada)*. ACM Press, pp. 88-95.
- Anick, P. & Tipirneni (1999). The Paraphrase Search Assistant: Terminological Feedback for Iterative Information Seeking. In *Proceedings of the Annual International ACM/SIGIR'99 Conference (Berkeley, USA)*. ACM Press, pp. 153-161.
- Aula, A. & Nordhausen, K. (forthcoming). Modeling Successful Performance in Web Search. To appear in *Journal of the American Society for Information Science and Technology (JASIST)*. Wiley.
- Barker, K. & Cornacchia, N. (2000). Using Noun Phrase Heads to Extract Document Keyphrases. In *Proceedings of the Thirteenth Canadian Conference on Artificial Intelligence, LNAI 1822 (Montreal, Canada)*. Pp. 40-52.
- Bates, M. (1989). The Design of Browsing and Berrypicking Techniques for the Online Search Interface. *Online Review*. Vol. 13, No. 5, pp. 407-424.
- Belkin, N., Cool, C., Head, J., Jeng, J., Kelly, D., Lin, S., Lobash, L., Park, S., Savage-Knepshield, P., & Sikora, C. (1999). Relevance Feedback versus

Local Context Analysis as Term Suggestion Devices: Rutgers' TREC-8 Interactive Track Experience. *TREC-8, Proceedings of the Eighth Text Retrieval Conference (Washington, D.C., USA)*.

- Berkhin, P. (2002). *Survey of Clustering Data Mining Techniques*. Accrue Software, San Jose, California. Available at:
<http://citeseer.ist.psu.edu/berkhin02survey.html>
- Boley, D., Gini, M., Hastings, K., Mobasher, B., & Moore, J. (1998). A Client-Side Web Agent for Document Categorization. *Journal of Internet Research*. Vol. 8, No. 5.
- Broder, A. (2002). A Taxonomy of Web Search. *SIGIR Forum*. ACM Press, Vol. 36, No. 2., pp. 3-10.
- Bruza, P. & Dennis, S. (1997). Query Reformulation on the Internet: Empirical Data and the Hyperindex Search Engine. In *Proceedings of RIAO'97 (Montreal, Canada)*. Pp. 500-508.
- Bruza, P., McArthur, R., & Dennis, S. (2000). Interactive Internet Search: Keyword, Directory and Query Reformulation Mechanisms Compared. In *Proceedings of Annual International ACM/SIGIR'2000 Conference (Athens, Greece)*. ACM Press, pp. 280-287.
- Card, S., Mackinlay, J., & Shneiderman, B. (1999). *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers, San Francisco.
- Carey, M., Heesch, D., & Rüger, S. (2003). Info Navigator: A visualization tool for document searching and browsing. In *Proceedings of the International Conference on Distributed Multimedia Systems (DMS, Florida, Sept 2003)*. Pp 23-28.
- Carey, M., Kriwaczek, F., & Rüger, S. (2000). A Visualization Interface for Document Searching and Browsing. In *Proceedings of NPIVM'2000 (Washington, D.C., USA)*. ACM Press.
- Chau, M., Zeng, D., & Chen, H. (2001). Personalized Spiders for Web Search and Analysis. In *Proceedings of JCDL'01 (Roanoke, USA)*. ACM Press.
- Chekuri, C., Goldwasser, M., Raghavan, P., & Upfal, E. (1997). Web Search Using Automatic Classification. In *Proceedings of the 6th International World Wide Web Conference, WWW6 (Santa Clara, USA)*.
- Chen, H. & Dumais, S. (2000). Bringing Order to the Web: Automatically Categorizing Search Results. In *Proceedings of the ACM SIGCHI*

- Conference on Human Factors in Computing Systems, CHI'2000 (The Hague, Netherlands)*. ACM Press, pp. 145–152.
- Chen, L. & Chue, W. (2005). Using Web Structure and Summarization Techniques for Web Content Mining. *Information Processing and Management*. Elsevier, Vol. 41, No. 5, pp. 1225–1242.
- Chen, M., Hearst, M., Hong, J., & Lin, J. (1999). Cha-Cha: A System for Organizing Intranet Search Results. In *Proceedings of the 2nd USENIX Symposium on Internet Technologies and SYSTEMS (USITS)*.
- Chen, M., LaPaugh, A., & Singh J. (2002). Predicting Category Accesses from a User in a Structured Information Space. In *Proceedings of the Annual International ACM/SIGIR'02 Conference (Tampere, Finland)*. ACM Press, pp. 65–72.
- Chi, E., Pirolli, P., Chen, K., & Pitkow, J. (2001). Using Information Scent to Model User Information Needs and Actions on the Web. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, CHI'01 (Seattle, USA)*. ACM Press, pp. 490–497.
- Cho, E. & Myaeng, S. (2000). Visualization of Retrieval Results Using DART. In *Proceedings of the International Conference RIAO (Paris, France)*.
- Choo, C., Detlor, B., & Trunbull, D. (2000). Information Seeking on the Web - An Integrated Model of Browsing and Searching. *First Monday* (<http://www.firstmonday.org>). University Library at the University of Illinois at Chicago, Vol. 5, No. 2.
- Cui, H. & Zaïaine, O. (2001). Hierarchical Structural Approach to Improving the Browsability of Web Search Engine Results. In *Proceedings of the 12th International Workshop on Database and Expert Systems Applications (DEXA'01)*. IEEE Computer Society.
- Cutting, D., Karger, D., & Pedersen, J (1993). Constant Interaction-Time Scatter/Gather Browsing of Large Document Collections. In *Proceedings of the Annual International ACM/SIGIR'93 Conference (Pittsburgh, USA)*. ACM Press, pp. 126–134.
- Cutting, D., Karger, D., Pedersen, J., & Tukey, J. (1992). Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. In *Proceedings of the Annual International ACM/SIGIR'92 Conference (Copenhagen, Denmark)*. ACM Press, pp. 318–329.
- Dennis, S., McArthur, R., & Bruza, P. (1998). Searching the World Wide Web Made Easy? The Cognitive Load Imposed by Query Refinement Mechanisms. In *Proceedings of the Third Australian Document Computing*

- Symposium (ADCS'98)*. Department of Computer Science, University of Sydney, TR-518, pp. 65–71.
- Dumais, S. & Chen, H. (2000). Hierarchical Classification of Web Content. In *Proceedings of the Annual International ACM/SIGIR'2000 Conference (Athens, Greece)*. ACM Press, pp. 256–263.
- Dumais, S. Cutrell, E., & Chen, H. (2001). Optimizing Search by Showing Results in Context. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, CHI'01 (Seattle, USA)*. ACM Press, pp. 277–283.
- Edgar, K., Nichols, D., Paynter, G., Thomson, K., & Witten, I. (2003). A user evaluation of hierarchical phrase browsing. In *Proceedings of the European Conference on Digital Libraries ECDL 2003 (Trondheim, Norway)*.
- Egan, D., Remde, J., Gomez, L., Landauer, T., Eberhardt, J., & Lochbaum, C. (1989). Formative Design-Evaluation of SuperBook. *AMC Transactions on Information Systems*. ACM Press, Vol. 7, No. 1, 30–57.
- Fox, E., Hix, D., Nowell, L., Brueni, D., Wake, W., Heath, L., & Rao, D. (1993). Users, User Interfaces, and Objects: Envision, a Digital Library. *Journal of the American Society for Information Science (JASIS)*. Wiley, Vol. 44, No. 3, pp. 480–491.
- Google Search Engine. <http://www.google.com>
- Google Timeline (2005)
<http://www.google.com/corporate/timeline.html>
- Granizer, M., Kienreich, W., Sabol, V., & Dösinger, G. (2003). WebRat: Supporting Agile Knowledge Retrieval through Dynamic, Incremental Clustering and Automatic Labelling of Web Search Result Sets. In *Proceedings of the Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'03)*. IEEE Computer Society.
- Gutwin, C., Paynter, G.W., Witten, I.H., Nevill-Manning, C., & Frank, E. (1999). Improving browsing in digital libraries with keyphrase indexes. *Journal of Decision Support Systems*. Elsevier, Vol. 27, No 1–2, pp. 81–104.
- Hearst, M. (1995). TileBars: Visualization of Term Distribution Information in Full Text Information Access. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, CHI'95 (Denver, USA)*. ACM Press, pp. 59–66.

- Hearst, M. (1999). User Interfaces and Visualization. Chapter in Baeza-Yates, R. and Ribeiro-Neto, B. (eds.). *Modern Information Retrieval*. Addison Wesley, Edinburgh Gate, England.
- Hearst, M. & Karadi, C. (1997). Cat-a-Cone: An Interactive Interface for Specifying Searches and Viewing Retrieval Results using a Large Category Hierarchy. In *Proceedings of the Annual International ACM/SIGIR'97 Conference (Philadelphia, USA)*. ACM Press, pp. 246–255.
- Hearst, M., Karger, D., & Pedersen, J. (1995). Scatter/Gather as a Tool for the Navigation of Retrieval Results. In *Proceedings of the American Association for Artificial Intelligence (AAAI) Conference, Fall Symposium "AI Applications in Knowledge Navigation & Retrieval"*. Cambridge, MA, pp. 65–71.
- Hearst, M. & Pedersen, J. (1996). Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results. In *Proceedings of the Annual International ACM/SIGIR'96 Conference (Zurich, Switzerland)*. ACM press, pp. 76–84.
- Heath, L., Hix, D., Nowell, L., Wake, W., Averbach, G., Labow, E., Guyer, S., Brueni, D., France, R., Dalai, K., & Fox, E. (1995). Envision: A User-Centered Database of Computer Science Literature. *Communications of the ACM*. ACM Press, Vol. 38, No. 4, pp. 52–53.
- Hoelscher, C. (1998). How Internet Experts Search for Information on the Web. In *Proceedings of the World Conference of the World Wide Web, Internet, and Intranet (Orlando, USA)*.
- iBoogie Search Engine. <http://www.iboogie.com>
- ISO/IEC 9241-11 (1998). *Ergonomic requirements for office work with visual display terminals (VDT)s - Part 11 Guidance on usability*. ISO/IEC 9241-11: 1998 (E).
- Jansen, B. & Pooch, U. (2000). A Review of Web Searching Studies and a Framework for Future Research. *Journal of the American Society of Information Science and Technology (JASIST)*. Wiley, Vol. 52, No. 3, pp. 235–246.
- Jansen, B. & Spink, A. (2006). How Are We Searching the World Wide Web? A Comparison of Nine Search Engine Transaction Logs. *Information Processing & Management*. Elsevier, Vol. 42, No. 1, pp. 248–263.
- Jansen, B., Spink, A., & Saracevic, T. (1998). Searchers, the subjects they search, and sufficiency: A study of a large sample of Excite searches.

In *Proceedings of WebNet-98, World Conference on the WWW, Internet and Intranet (Orlando, USA)*.

Jansen, B., Spink, A., & Saracevic, T. (2000). Real Life, Real Users, and Real Needs. A Study and Analysis of User Queries on the Web. *Information Processing and Management*. Vol. 36, No. 2, pp. 207-227.

Jardine, N. & van Rijsbergen, C. (1971). The Use of Hierarchic Clustering in Information Retrieval. *Information Storage and Retrieval*. Pergamon Press, Vol. 7, No. 5, pp. 217-240.

Jiang, Z., Joshi, A., Krishnapuram, R., & Yi, L. (2000). *Retriever: Improving Web Search Engine Results Using Clustering*. University of Maryland, Technical Report, October 2000.

Jones, S. (1999). Design and Evaluation of Phrasier, an Interactive System for Linking Documents using Keyphrases. In *Proceedings of Human-Computer Interaction INTERACT'99 (Edinburgh, UK)*. IOS Press, pp. 483-490.

Jones, S. & Mahoui, M. (2000). Hierarchical Document Clustering Using Automatically Extracted Keyphrases. In *Proceedings of the Third Asian Conference on Digital Libraries (Seoul, Korea)*. Pp. 113-120.

Jones, S. & Paynter, G. (1999). Topic-based browsing within a digital library using keyphrases. In *Proceedings of the fourth ACM Conference on Digital Libraries'99 (Berkeley, USA)*. ACM Press, pp. 114-121.

Jones, S. & Paynter, G. (2002). Automatic Extraction of Document Keyphrases for Use in Digital Libraries: Evaluation and Applications. *Journal of the American Society for Information Science and Technology (JASIST)*. Wiley, Vol. 53, No. 8, pp. 653-677.

Jones, S., Jones, M., & Deo, S. (2004). Using Keyphrases as Search Result Surrogates on Small Screen Devices. *Personal Ubiquitous Computing*. Springer, Vol. 8, No. 1, pp. 55-68.

Kartoo Search Engine. <http://www.kartoo.com>

Kaski, S., Honkela, T., Lagus, K., & Kohonen, T. (1998). WEBSOM - Self-organizing Maps of Document Collections. *Neurocomputing*. Elsevier, Vol. 21, pp. 101-117.

Kohonen, T. (1997). Exploration of Very Large Databases by Self-organizing Maps. In *Proceedings of the IEEE International Conference on Neural Networks*. Vol. 1, pp. 1-6.

- Koller, D. & Sahami, M. (1997). Hierarchically Classifying Documents Using Very Few Words. In *Proceedings of the 14th International Conference on Machine Learning, ICML (Nashville, USA)*. Pp. 170–178.
- Kules, B. & Shneiderman, B. (2005). Categorized Graphical Overviews for Web Search Results: An Exploratory Study Using U.S. Government Agencies as a Meaningful and Stable Structure. In *Proceedings of the Third Annual Workshop on HCI Research in MIS*. Technical report HCIL-2004-38, CS-TR-4715, UMIACS-TR-2005-23, ISR-TR-2005-71.
- Kummamuru, K., Lotlikar, R., Roy, S., Singal, K., & Krishnapuram, R. (2004). A Hierarchical Monothetic Document Clustering Algorithm for Summarization and Browsing Search Results. In *Proceedings of the Thirteenth International World Wide Web Conference (New York, USA)*. Pp. 658–665.
- Leouski, A. & Croft, B. (1996). *An Evaluation of Techniques for Clustering Search Results*. Department of Computer Science, University of Massachusetts, Amherst, Technical Report IR-76.
- Lin, X., Soergel, D., & Marchionini, G. (1991). A Self-organizing Semantic Map for Information Retrieval. In *Proceedings of the Annual International ACM/SIGIR'91 Conference*. ACM Press, pp. 262–269.
- Maarek, Y., Jacovi, M, Shtalhaim, M, Ur, S., Zernik, D., & Shaul, I. (1997). WebCutter: A System for Dynamic and Tailorable Site Mapping. In *Proceedings of the 6th International World Wide Web Conference, WWW6 (Santa Clara, USA)*. Pp. 713–722.
- MacKenzie, S. (2002). KSPC (Keystrokes Per Character) as a Characteristic of Text Entry Techniques. In *Proceedings of the Fourth International Symposium on Human-Computer Interaction with Mobile Devices (Heidelberg, Germany)*. Springer-Verlag, pp. 195–210.
- Matthews, G., Davies, R., Westerman, S., & Stammers, R. (2000). *Human Performance: Cognition, Stress and Individual Differences*. Psychology Press, Hove, UK.
- Mauldin, M. (1997). Lycos: Design Choices in an Internet Search Service. *IEEE Expert*. Vol. 12, No. 1, pp. 8–11.
- MeSH <http://www.nlm.nih.gov/mesh/meshhome.html>
- Nielsen, J. (2004). When Search Engines Become Answer Engines. At <http://www.useit.com/alertbox/20040816.html>
- Nowell, L., France, R., Hix, D., Heath, L., & Fox, E. (1996). Visualizing Search Results: Some Alternatives to Query-Document Similarity. In

- Proceedings of the Annual International ACM/SIGIR'96 Conference (Zurich, Switzerland)*. ACM Press, pp. 67–75.
- Osdin, R., Ounis, I., & White, R. (2002). Using Hierarchical Clustering and Summarization Approaches for Web Retrieval: Glasgow at the TREC 2002 Interactive Track. In *The Eleventh Text Retrieval Conference (TREC 2002)*. Pp. 640–644.
- Pirolli, P. & Card, S. (1995). Information Foraging in Information Access Environments. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, CHI'95 (Denver, USA)*. ACM press, pp. 51–58.
- Pirolli, P. & Card, S.K. (1999). Information Foraging. *Psychological Review*. APA, Vol. 106, No. 4, pp. 643–675.
- Pirolli, P., Pitkow, J., & Rao, R. (1996). Silk from a Sow's Ear: Extracting Usable Structures from the Web. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, CHI'96 (Vancouver, Canada)*. ACM press, pp. 118–125.
- Pirolli, P., Schank, P., Hearst, M., & Diehl, C. (1996). Scatter/Gather Browsing Communicates the Topic Structure of a Very Large Text Collection. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, CHI'96 (Vancouver, Canada)*. ACM press, pp. 213–220.
- Popescul, A. & Ungar, L. (2000). Automatic Labeling of Document Clusters. Unpublished manuscript, available at: <http://citeseer.ist.psu.edu/popescul00automatic.html>
- Pratt, W. & Fagan, L. (2000). The Usefulness of Dynamically Categorizing Search Results. *Journal of the American Medical Informatics Association*. Elsevier, Vol. 7, No. 6, pp. 605–617.
- Remde, J. Gomez, L., & Landauer, T. (1987). SuperBook: An automatic tool for information exploration - hypertext? In *Proceedings of the ACM Conference on Hypertext and Hypermedia, Hypertext'87 (Chapel Hill, USA)*. ACM Press, pp. 175–188.
- Robertson, S. (1977). Theories and Models in Information Retrieval. *Journal of Documentation*. Emerald, Vol. 33, No. 2, pp. 126–148.
- Rose, D. & Levinson, D. (2004). Understanding User Goals in Web Search. In *Proceedings of the Thirteenth International World Wide Web Conference, WWW2004 (New York, USA)*. ACM Press, pp. 13–19.

- Roussinov, D. & Chen, H. (2001). Information Navigation on the Web by Clustering and Summarizing Query Results. *Information Processing & Management*. Elsevier, Vol. 37, pp. 789–816.
- Sahami, M., Yusufali, S., & Baldonado, M. (1998). SONIA: A Service for Organizing Networked Information Autonomously. In *Proceedings ACM Conference on Digital Libraries'98 (Pittsburgh, USA)*. ACM Press, pp. 200–209.
- Salton, G. (1989) *Automatic Text Processing: the Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley Publishing Company, Inc. Reading, Massachusetts.
- Saracevic, T., Kantor, P., Chamis, A., & Trivison, D. (1988). A Study of Information Seeking and Retrieving. I. Background and Methodology. *Journal of the American Society for Information Science*. Wiley, Vol. 39, No. 3, pp. 161–176.
- Shneiderman, B., Byrd, D., & Croft, B. (1997). Clarifying Search – A User-Interface Framework for Text Searches. *D-Lib Magazine*, January 1997.
- Shneiderman, B., Byrd, D., & Croft, B. (1998). Sorting Out Search – A User-Interface Framework for Text Searches. *Communication of the ACM*. ACM Press, Vol. 41, No. 4, pp. 95–98.
- Shneiderman, B., Feldman, D. & Rose, A. (1999). Visualizing Digital Library Search Results with Categorical and Hierarchical Axes. CS-TR-3993, UMIACS-TR-99-12. <ftp://ftp.cs.umd.edu/pub/hcil/Reports-Abstracts-Bibliography/99-03html/99-03.html>.
- Soukoreff, W. & MacKenzie, S. (2003). Metrics for Text Entry Research: An Evaluation of MSD and KSPC, and a New Unified Error Metric. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, CHI'2003 (Lauderdale, USA)*. ACM Press, pp. 113–120.
- Spink, A., Jansen, B., Wolfram, D., & Saracevic, T. (2002) From E-Sex to E-Commerce: Web Search Changes. *IEEE Computer*. IEEE Computer Society, Vol. 55, No. 3, pp. 107–109.
- Spink, A., Wolfram, D., Jansen, B., & Saracevic, T. (2001) Searching the Web: The Public and Their Queries. *Journal of the American Society for Information Science and Technology (JASIST)*. Wiley, Vol. 52, No. 3, pp. 226–234.
- Spoerri, A. (1994a). InfoCrystal: A Visual Tool for Information Retrieval & Management. In *Proceedings of the ACM SIGCHI Conference on Human*

- Factors in Computing Systems, CHI'94 (Boston, USA)*. ACM Press, pp. 11-12.
- Spoerri, A. (1994b). InfoCrystal: A Visual Tool for Information Retrieval and Management. In *Proceedings of Information Knowledge and Management (CIKM'93)*. ACM Press, pp. 150-157.
- Spoerri, A. (2004a). Visual Search Editor for Composing Meta Searches. In *Proceedings of ASIST'2004 (Providence, USA)*.
- Spoerri, A. (2004b). MetaCrystal: Visualizing the Degree of Overlap between Different Search Engines. In *Proceedings of the Thirteenth International World Wide Web Conference, WWW2004 (New York, USA)*. ACM Press.
- Sutcliffe, A, & Ennis, M. (1998). Towards a Cognitive Theory of Information Retrieval. *Interacting with Computers*. Elsevier, Vol. 10, No. 3, pp. 321-351.
- Teoma Search Engine. <http://www.teoma.com>
- Tombros, A. & Sanderson, M. (1998). Advantages of Query Biased Summaries in Information Retrieval. In *Proceedings of the Annual International ACM/SIGIR'98 Conference (Melbourne, Australia)*. ACM Press, pp. 2-10.
- Turney, P. (2000). Learning Algorithms for Keyphrase Extraction. *Information Retrieval*. Kluwer Academic Publishers, Vol. 2, No. 4, pp. 303-336.
- Vakkari, P. (1999). Task Complexity, Problem Structure and Information Actions: Integrating Studies on Information Seeking and Retrieval. *Information Processing and Management*. Elsevier, Vol. 36, No. 6, pp. 819-837.
- Veerasamy, A. & Belkin, N. (1996). Evaluation of a Tool for Visualization of Information Retrieval Results. In *Proceedings of the Annual International ACM/SIGIR'96 Conference (Zurich, Switzerland)*. ACM Press, pp. 85-92.
- Veerasamy, A. & Heikes, R. (1997). Effectiveness of a Graphical Display of Retrieval Results. In *Proceedings of the Annual International ACM/SIGIR'97 Conference (Philadelphia, USA)*. ACM Press, pp. 236-245.
- Vélez, B., Weiss, R., Sheldon, M., & Gifford, D. (1997). Fast and Effective Query Refinement. In *Proceedings of the Annual International ACM/SIGIR'97 Conference (Philadelphia, USA)*. ACM Press, pp. 6-15.

Vivísimo Search Engine. <http://www.vivisimo.com>

Voorhees, E. (1985). Cluster Hypothesis Revisited. In *Proceedings of the Annual International ACM/SIGIR'85 Conference (Montreal, Canada)*. ACM Press, pp 188–196.

Ware, C. (2004). *Information Visualization – Perception for Design (second edition)*. Morgan Kaufmann Publishers, San Francisco.

Weiss, D. & Stefanowski, J. (2003). Web Search Results Clustering in Polish: Experimental Evaluation of Carrot. In *Advances in Soft Computing, Intelligent Information Processing and Web Mining, Proceedings of the International IIS: IIPWM'03 Conference (Zakopane, Poland)*. Vol. 578 (XIV), pp. 209–220.

White, R., Jose, J., & Ruthven, I. (2001). Query-Biased Web Page Summarization: A Task-Oriented Evaluation. In *Proceedings of the Annual International ACM/SIGIR'2001 Conference (New Orleans, USA)*. ACM Press.

WiseNut Search Engine. <http://www.wisenut.com>

Wittenburg, K. & Sigman, E. (1997). Integration of Browsing, Searching, and Filtering in an Applet for Web Information Access. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, CHI'97 (Atlanta, USA)*. ACM Press.

Wu, Y., Shankar, L., & Chen, X. (2003). Finding More Useful Information Faster from Web Search Results. In *Proceedings of Information Knowledge and Management, CIKM'03 (New Orleans, USA)*. ACM Press, pp. 568–571.

Yahoo! Search Engine. <http://www.yahoo.com>

Zamir, O. (1998). *Visualization of Search Results in Document Retrieval Systems - General Examination*. University of Washington, SIGTRS Bulletin, Vol. 7, Num. 2 (6.2001).

Zamir, O. & Etzioni, O. (1998). Web Document Clustering: A Feasibility Demonstration. In *Proceedings of the Annual International ACM/SIGIR'98 Conference (Melbourne, Australia)*. ACM Press, pp. 46–54.

Zamir, O. & Etzioni, O. (1999). Grouper: A Dynamic Clustering Interface to Web Search Results. In *Proceedings of the International WWW Conference WWW'8 (Toronto, Canada)*. Elsevier Science, pp. 1361–1374.

Zamir, O., Etzioni, O., Madani, O., & Karp, R. (1997). Fast and Intuitive Clustering of Web Documents. In *Proceedings of the ACM SIGKDD*

International Conference on Knowledge Discovery and Data Mining (Newport Beach, USA). ACM Press, pp. 287–290.

Zeng, H., He, Q., Chen, Z., Ma, W., & Ma, J. (2004). Learning to Cluster Web Search Results. In *Proceedings of the Annual International ACM/SIGIR'04 Conference (Sheffield, UK)*. ACM Press, pp. 210–217.

Appendix 1

The tasks and queries used in the studies. All the tasks were presented to the participants in Finnish. For this table, the tasks were translated in English. Queries are reported as sent to the search engine with translations in parenthesis.

Task	Query
Experiment of Statistical Categories (Paper I)	
Find information about the space shuttle challenger accident	challenger
Find picture about the volcano Pinatubo	pinatubo
Find information about the terrorist attack on World Trade Center	world trade center
Find information sources about growing tulips	tulppaani (tulip)
Find pages that deal generally with the city of Oulu	oulu
Find information about the things that should be considered when buying a used car from Finland	käytetty auto (used car)
Find information about the Finnish national opera (kansallisooppera)	kansallisooppera (national opera)
Find information about sinking of Titanic ship	titanic
Find pages concerned with the Kobe earth quake	kobe
Find information about the terrorist attack to Pentagon	pentagon
Find information about growing crocus	krookus (crocus)
Find pages that concern the university of Oulu in general	oulu
Find pictures of the Jupiter planet	jupiter
Find reviews of the sound track of the film 'Pahat Pojat'	pahat pojat
Find sources from where you could get a free email address	sähköposti (email)
Find as many Finlandia-prize winners as you can (avoid collecting the same author many times)	finlandia palkinto (Finlandia-prize)
Experiment on the effect of the number of categories (Paper III)	
Find information about the space shuttle challenger accident	challenger
Find picture about the volcano Pinatubo	pinatubo
Find information about the terrorist attack on World Trade Center	world trade center
Find recipes of American Apple Pie	apple pie
Find opportunities to get a summer job as a sales person	kesätyö (summer job)
Find pictures of the planet Mars	mars
Find information about the things that should be considered when buying a used car from Finland	käytetty auto (used car)
Find information about sinking of Titanic ship	titanic
Find pages concerned with the Kobe earth quake	kobe
Find information about the terrorist attack to Pentagon	pentagon
Find recipes of minestrone soup	minestrone
Find information about the new Miss Finland (2004)	miss suomi (Miss Finland)

Task	Query
Find pictures of the planet Jupiter	jupiter
Find sources from where you could get a free email address	sähköposti (email)
Find information about the flight accident happened over Lockerbie in Scotland	lockerbie
Find pages concerning volcano eruption happened in the mid 1990s in Iceland	Iceland eruption
Find reasons for climate warming	climate warming
Find recipes for making tiramisu	tiramisu
Find pages concerning the composing of a will	testamentti (will)
Find pictures of Moon	moon
Find instructions for wool washing	wool washing
Experiment on the context categories (Paper V)	
Find information about the space shuttle challenger accident	challenger
Find picture about the volcano Pinatubo	Pinatubo
Find ideas (instructions, recipes) about what can be done from chocolate	chocolate
Find information about what the world health organization (WHO) is doing to cure river blindness	river blindness
Find pages that deal generally with the city of Oulu	oulu
Find pictures of the planet Venus	venus
Find pages where you get information about preventing influenza	influenza
Let's imagine that you want to buy a mobile phone with a camera. Find pages where you find the prices of such products.	kamerapuhelin (camera phone)
You think you have seen a barnacle goose. Find pages with which you can confirm your observation (a picture, identification information).	valkoposkihanhi (barnacle goose)
Find information about sinking of Titanic ship	titanic
Find information about the hurricanes appeared this autumn (2004) in United States and in Caribbean	hurricane
Find ideas about what else can be done from tea leaves other than normal tea	tea
Find information about the actions the world health organization (WHO) takes against tuberculosis	tuberculosis +who
Find pages that concern the university of Oulu in general	oulu
Find information about colored contact lenses	contact lenses
Let's imagine that you want to buy a DVD-player. Find price information of various products	dvd-soitin (dvd-player)
You think you have seen a goldeneye. Find pages with which you can confirm your observation (pictures, identification information)	telkkä (goldeneye)