

Support for Information-Seeking Strategies

**Von der Fakultät für Ingenieurwissenschaften
der Universität Duisburg-Essen
Abteilung Informatik und Angewandte Kognitionswissenschaft
zur Erlangung des akademischen Grades eines
Doktors der Ingenieurwissenschaften (Dr.-Ing.)
genehmigte Dissertation**

von

Dipl.-Inform. Matthias Jordan
aus Oberhausen

1. Gutachter: Prof. Dr.-Ing. Norbert Fuhr
2. Gutachter: Prof. Dr.-Ing. Jürgen Ziegler
Tag der mündlichen Prüfung: 11. September 2015

Contents

1. Introduction	1
1.1. Information needs	1
1.2. Search as sequences of actions	3
1.3. The research question	4
1.4. The notion of support	5
1.5. Focus	6
1.6. Outline	7
2. Search Classifications	9
2.1. Introduction	9
2.2. Longitudinal models	10
2.2.1. Ellis' interaction model	10
2.2.2. Kuhlthau's stage model	12
2.2.3. Spink's IR interaction model	12
2.3. Latitudinal models	13
2.3.1. Levels of search activities by Marcia Bates	13
2.3.2. Belkin's, Marchetti's and Cool's classification of information seeking strategies	14
3. Support for Scanning	25
3.1. Exploratory search	25
3.1.1. The evolving information need and sense-making	26
3.1.2. Use cases	26
3.1.3. Needed features	27
3.1.4. Subjunctive designs	28
3.1.5. Trailblazer	28
3.1.6. mSpace	28
3.1.7. Flamenco	29

Contents

3.1.8. MedioVis	30
3.1.9. The source of improvement by browsing	33
3.1.10. Evaluating exploratory search	33
3.2. The relationship between exploratory search and scanning	34
3.3. System-oriented support	34
3.3.1. Recall-oriented searching	34
3.3.2. Query term expansion	35
3.4. User-oriented support	36
3.4.1. Berrypicking tray	37
3.4.2. Result list support mechanisms	38
3.5. Conclusion	40
4. Support for Searching	41
4.1. How searching relates to the other facet values	41
4.2. Query biased summaries	42
4.3. Conclusion	42
5. Support for Recognition	43
5.1. Introduction	43
5.2. Related work	45
5.3. Experiment 1: linear and table-based result presentation	47
5.3.1. Experimental conditions	48
5.3.2. Method and apparatus	49
5.3.3. Tasks	51
5.3.4. Lists	54
5.3.5. Sample	55
5.3.6. Results	55
5.3.7. Discussion	63
5.4. Experiment 2: baseline, linear and table-based result presentation	64
5.4.1. Method and apparatus	64
5.4.2. Sample	65
5.4.3. Results	66
5.4.4. Discussion	68
5.5. Collective analysis	69
5.6. Conclusion and outlook	69

6. Support for Specification	71
6.1. Query input	71
6.1.1. Query languages	72
6.1.2. Query forms	76
6.1.3. Query operators	78
6.1.4. Incremental query building	79
6.2. Spelling correction	80
6.2.1. Prevalence of spelling errors	80
6.2.2. The influence of hard-to-spell terms	81
6.2.3. Spelling corrections in ezDL	81
6.3. Query translations	82
6.4. Query suggestions	82
6.5. Proactive support	84
6.5.1. Agent to Improve Information Retrieval Systems	84
6.5.2. Proactivity and reactivity	85
6.5.3. Proactive suggestions in DAFFODIL and ezDL	85
6.5.4. How to display term suggestions of multiple kinds?	86
6.6. Conclusion	87
7. Combining Optimal Support Mechanisms	89
7.1. Research question and hypotheses	89
7.1.1. Research question A	89
7.1.2. Research question B	90
7.2. Related work	90
7.3. Selected support mechanisms	92
7.3.1. Scanning	93
7.3.2. Recognition	97
7.3.3. Specification	100
7.4. The three variants and experimental conditions	101
7.4.1. Baseline	102
7.4.2. Experimental A (adaptive)	102
7.4.3. Experimental B (integrated)	104
7.5. Pilot test	104
7.6. Operationalization	106
7.6.1. Participants	106
7.6.2. User model and tutorials	107

Contents

7.6.3. Collection	107
7.6.4. Tasks	107
7.6.5. Metrics	108
7.6.6. Procedure	114
7.6.7. Problems encountered	119
7.7. Results	119
7.7.1. Preliminary data analysis	119
7.7.2. The participants	120
7.7.3. The SUS baseline task	120
7.7.4. Main measure	121
7.7.5. Did conditions differ for single ISSs?	124
7.7.6. Did stress vary between groups?	127
7.7.7. Were there any differences in user satisfaction?	127
7.7.8. Were the support features of any help?	129
7.7.9. Demographics and search	130
7.7.10. Does Heap’s law hold for the pool?	131
7.8. Discussion	132
7.8.1. Main measures	132
7.8.2. Differences in the SUS scores	135
7.8.3. Tasks, ISSs, and participants	135
7.8.4. Limitations	136
7.9. Conclusion	137
8. Conclusion	139
8.1. Summary	139
8.2. Outlook	141
Appendix	145
A. Tasks	145
A.1. Set Training	145
A.2. Set A	145
A.3. Set B	146
A.4. Set C	147
B. Handouts used in experiment 1	149

C. Handouts used in experiment 2	151
D. Handouts used in the final experiment	153
E. Original screenshots from the final experiment	159
F. The custom thesaurus	163
G. Problems encountered in the final experiment	165
G.1. Client replacements and bug fixes	165
G.2. Rejected participants	168
H. Strange results	169
List of Figures	171
Bibliography	175

1. Introduction

1.1. Information needs

The central problem in the user-centered view on information retrieval is to bridge the gap between the user's discovery of an anomalous state of knowledge (ASK, [15]) and the documents that resolve the ASK. In this setting, the user's most important—and at the same time most difficult—task is to make the step from his perception of the information need that causes the ASK to a request that they can use to query an IR system. In many cases, taking this step is not trivial: the ASK itself is the reason for the user lacking the means to express what they would have to know to resolve it.

To overcome this problem, information searchers often use sequences of different search activities, some of which involve paraphrasing the information need in different ways and a host of other methods. Current IR systems restrict the degrees of freedom of the user in this respect: they often only support a few search activities well and other search activities have to be performed using external means for support, making the search less integrated, less pleasant for the user and more error-prone.

For example, consider the popular Google search engine. The interaction supported by its interface is entering a query and then examining the linear result list, possibly followed by entering a loop of reformulating the query and examining the new result set. This is an interaction paradigm that supports just a few types of information needs, such as goal-directed search, and apparently serves the average web user very well. But searches that go beyond looking up single instances of popular items are not well-supported. Exploratory search, in particular, is not well-supported by Google.

Even known-item search is in some cases not well-supported by this paradigm. Suppose the user can describe the look of a web page and roughly what it is about. In Google,

1. Introduction

The image shows a search form titled "Books Search" on the Amazon website. The form is divided into two columns by a vertical dashed line. The left column contains text input fields for "Keywords", "Author", "Title", "ISBN(s)", and "Publisher". Below these is a dropdown menu for "Subject" with "All Subjects" selected. The right column contains several dropdown menus: "Condition" (All Conditions), "Format" (All Formats), "Binding" (All Bindings), "Reader Age" (All Ages), and "Language" (All Languages). Below these is a date selection section with "Pub. Date" (All Dates), "Month", and "Year" dropdowns. At the bottom right of the right column is a "Sort Results by:" dropdown menu with "Relevance" selected. A yellow "Search" button is centered below the two columns.

Figure 1.1.: The search form on www.amazon.com

that would mean querying by related keywords and then visiting each site to see if it looks like the site the searcher has in mind. The online shop Amazon solves a similar problem by showing thumbnails of book covers in the result list (see Figure 1.2). This way, searchers can immediately recognize a familiar book by quickly glancing over the list.

Amazon is one of the most popular databases of book meta-data, but issues similar to those with Amazon occur when considering a digital library or an OPAC system. Here, too, users try to find items based on sometimes vague descriptions, and information needs also sometimes encompass situations when users can best use the visual appearance of a document to describe it. A user of the ACM Digital Library might look for a paper of an Asian author that has a graph right on the front page that the user wants to find. Or the user might remember that the paper was in single-column layout and remembers only a few terms from the title.

So, even though there are many different search activities performed by searchers, only few of them are well-supported by each existing retrieval system. The systems that support all activities in some degree do not support all activities equally well. Systems aimed at exploratory search perform worse in supporting specification [138].

1.2. Search as sequences of actions



The screenshot shows the top of an Amazon search results page. At the top, it says "Showing 1 - 12 of 1,561 Results" and "Sort by Relevance". There are two search results listed:

- 1.** **Introduction to Information Retrieval** by Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze (**Hardcover** - Jul 7, 2008)
Buy new: ~~\$60.00~~ **\$48.00** 48 Used & new from \$38.81
Get it by **Tuesday, April 28** if you order in the next **31 hours** and choose one-day shipping.
Eligible for **FREE** Super Saver Shipping.
★★★★☆ (9)
- 2.** **Search Engines: Information Retrieval in Practice** by Bruce Croft, Donald Metzler, and Trevor Strohman (**Hardcover** - Feb 16, 2009)
Buy new: ~~\$86.67~~ **\$69.33** 19 Used & new from \$69.33
Get it by **Tuesday, April 28** if you order in the next **31 hours** and choose one-day shipping.
Eligible for **FREE** Super Saver Shipping.

Figure 1.2.: The results on `www.amazon.com`

1.2. Search as sequences of actions

What users of search systems do can be described as a series of actions, many of which are interactions with the search system. These actions can be described by behavioral models. Simple models describe the user behavior as a cycle of querying the system and examining the result list, while more elaborate models differentiate between either search progress or a set of possible actions without regarding progress.

These models sometimes serve as a source of requirements for search systems: Scholars within the information retrieval community have long since proposed that a system should support as many user actions as possible for the support to be optimal [8], and behavioral models are sometimes used to define which actions are to be considered. Supporting all search actions can be accomplished in a variety of ways. In the dimension of the amount of user support, the one extreme—no support—is providing a user interface with just one widget: a text box. In this text box, a general language can be used to express search actions in a way that allows the user to make the system do exactly what is needed. If one carried this idea ad absurdum, this would mean leaving users with a general-purpose programming language, such as Java, which is provable to be very versatile, but has a steep learning curve and is not very efficient to use if every user were required to first program their own search system each time they want to search for something. The other extreme, perfect support, is offering a button that does exactly what the user wants. Clicking it returns exactly the desired documents, no more and nothing less. The problem with this hypothetical perfect support is that there would have to be an infinite number of these buttons available

1. Introduction

to cover all information needs. So at the one end of the spectrum, the user interface is quite clean but the user has to do everything by herself. On the other end, the user only has to click a button, but the interface is overcrowded with buttons beyond being usable.

So the right solution is somewhere between the Java compiler and the infinite number of buttons: a search system that is capable of supporting many actions and that balances complexity and versatility by offering support features for common problems so that the user can choose and combine them as needed for each situation.

This creates a problem in designing such a balanced search system: It is possible that some of the support mechanisms provided by the search system are in conflict with each other—e.g. in terms of the user's resources, such as attention, or the system's resources, such as screen estate. This means that the design space is the range between having all support mechanisms for all possible actions at once laid out in front of the user (with potentially many conflicts) on the one end, and having a single specific user interface for each possible action (without conflicts) on the other. The latter extreme, in turn, results in another problem: The user will move from one action to another during the search process. In order to give the user optimal support for each step, the user interface would need to adapt to the action the user is going to take next. Who will be in charge of that adaptation? Is the system going to guess what the user will do next? Or will the user be required to explicitly make the system adapt?

These are two different questions. The first one is whether a general interface or multiple specific interfaces are better. The second question is if multiple interfaces are needed, how can the system adapt the interface to the user's (upcoming) actions?

1.3. The research question

The question in the focus of this work is the first one: How can a system support each action of the user optimally? By using multiple specific user interfaces or a general one?

To answer this question, interfaces for each of a set of search actions and a general interface have to be implemented and compared with each other. For this purpose, it

1.4. The notion of support

is necessary to know which search actions users perform and how these search actions can be supported.

If there are features known to support each action, the question can be examined whether an adaptive IR system, that is built to support all these actions, is better than a traditional system, and if so, how much.

As a first approximation, this is a naïve question: If a system that is built to support all actions does not work better than the baseline, then maybe the mechanisms used to support certain actions do not, in fact, support them. But it might also be not better than the baseline system if the cost associated with the adaption outweighs the benefits that come with having all actions supported. It has been shown that users can compensate differences in MAP scores between different retrieval systems by their interaction [67]. This might also be the case for differences in their user interfaces. Building an IR system that supports all actions might on the one hand be very expensive but on the other hand result in very small benefit. This cost-benefit ratio might not be reasonable. The gain by supporting the two most common actions may be similar to supporting more or even all actions.

Even if the implementation costs of an adaptive search system are ignored, the costs that cannot be ignored are the possible costs of the interaction between the users and the adaptive system. An important cost factor in this regard is the confusion due to an ever-changing user interface. So a follow-up question is: Can a system that supports all actions be built without confusing its users?

Let us again assume there is an IR system that supports all actions in a single, static interface. The number of these actions could be very high so that even the static interface might be confusing to the user. In this case, the gain through supporting more actions than the baseline system might be counteracted by the confusing user interface. Maybe the confusion on the user's side is worth the gain in retrieval quality and can be reduced by training.

1.4. The notion of support

In many passages in this thesis, the term “support” is used in some form. However, “support” can have various meanings that range from enabling something to mere

1. Introduction

assistance. According to the Oxford Encyclopedic English Dictionary, “to support” can mean to keep something from failing:

support, *v.* 2 keep from falling, sinking or failing [...] 6 give help or countenance to, back up – The Oxford Encyclopedic English Dictionary [63]

It can also mean to give help. The latter meaning is what is meant by the term in this thesis: Assuming that many search actions can be performed even without the help of a computer, support means that a search action can be performed more easily. But where does support begin? Does a directory of plain-text files along with `grep` in a Unix command shell already mean support for exploratory search? If not, why not? In comparison to working with paper versions of these documents, indexed on micro fiche, one could contend that `grep` should be faster and easier to use, and thus be a support mechanism. Compared to a more complex system, though, like mSpace or other modern exploratory search systems introduced in Section 3.1, `grep` in a Unix shell seems antiquated and more like an impediment when it comes to exploratory search. To determine whether a feature of a search system is regarded as a support feature, it is compared with a baseline system in the form of popular web search engines. If the feature provided more help for a given search action than is available in mainstream web search engines, it is considered to be a support feature.

1.5. Focus

Information search is an extensive field that covers different types of searchers, many application domains, and multiple modalities of search. Searchers can be expert searchers or casual searchers; they can be blind or normal-sighted; they can be adults or young children. They might search in collections of books, audio and video files, images and even genomes. Some of them use systems that follow the query paradigm, while some use systems that offer a classification for browsing, and still others use systems with ostensive browsing based on similarities between documents.

The research question can therefore not be answered globally for all of these variations, but it has to be answered separately for each specific case. This thesis focuses on searchers without any handicap, using the query paradigm for searching literature:

this population is easy to recruit participants from in a university setting and querying for literature search is a common task that many students are familiar with.

1.6. Outline

Ideally, a search system should have mechanisms to optimally support all possible search actions. Of course, supporting each single search action optimally is very difficult, if not impossible, since there are so many different actions a searcher could choose from. For each of these actions a system with optimal support would have to be determined empirically. To get closer to this unachievable goal, the requirement to find optimal support mechanisms is relaxed and replaced by the requirement to find good (i. e. better than something else) ones. Additionally, the idea of supporting each individual action is dropped, assuming that similar search actions can be supported by similar means. This reduces the task at hand to choosing a classification of search actions and finding good support mechanisms for each of its classes.

To that end, the early version of the classification of information seeking strategies (ISS) by Belkin, Marchetti and Cool is used [16, 17, 18, 32]. The ISS classification is a faceted classification with only 16 classes defined by four facets, which makes it easy to use. Chapter 2 introduces classifications of search actions found in the literature with a special emphasis on the ISS classification. Existing ambiguity concerning the ISS classification is resolved by providing proper definitions of its facets and facet values.

The 16 classes of the ISS classification are not all equally relevant when it comes to designing user interfaces (see Section 2.3.2). This author decided to focus on the facets “method” and “mode” and their values “scanning”, “searching”, “recognition”, and “specification.” The chapters after Chapter 2 examine the question of what are good support mechanisms for each of these facet values:

Chapter 3 summarizes research literature concerning exploratory search, since this is a concept that has some similarities with scanning as defined in the ISS classification.

Chapter 4 describes a baseline interface without any additional support. It is assumed that searching is supported by all IR systems.

1. Introduction

Chapter 5 describes experiments that examined possible support mechanisms for recognition. The mechanisms examined were two experimental result list designs—a table-based design and a design that used highlighting to guide the eye of the user to surrogate parts of interest.

Chapter 6 describes research results on specification actions, including experiments that evaluated proactive suggestions aiming for supporting query specification.

Chapter 7 reports an experiment that examined the main question of this thesis: Is it necessary or beneficial to provide specialized user interfaces for each action to the user, or is a combined user interface enough?

Finally, Chapter 8 summarizes the findings of this thesis, discusses its possible implications and lists the open questions that remain to be examined in further studies.

2. Search Classifications

2.1. Introduction

The ultimate goal of optimally supporting all search actions is unachievable. The relaxed goal of supporting classes of similar search actions should be easier to achieve, since the number of classes of search actions is much smaller than the number of individual search actions themselves.

To examine this question, one first has to define and classify search activities. A large body of work exists in the area of models in information behavior [19, 72, 140, 141], so there are many classifications of search activities to choose from. Classifications with an explicit focus on searching or seeking that are often cited are:

- Ellis' Behavioural Model of Information Seeking Strategies [47]
- Kuhlthau's stage process model [88]
- Spink's interaction model [113]
- Bates' four-tier model of move, tactics, stratagem and strategy [7, 9]
- Belkin's, Marchetti's and Cool's classification of interactions with information [16, 17, 18, 32]

Some of these classifications are actually behavioral models of search, but they can also be considered as classifications since each step in such a model describes a kind of behavior, or action, of a search system user. These kinds of actions can be interpreted as classes, so that behavioral models sometimes double as classifications, with the bonus that they also describe relationships between the classes. For instance, Ellis

2. Search Classifications

suggested that a search episode can be “validating” [47]; this “validating” step can also be regarded as the class of all search behaviors that try to establish the trustworthiness of the information obtained. Interestingly, Ellis’ original publication did not explicitly state a temporal relationship between the classes of the model; this relationship was later extrapolated by Wilson [141], a fact that shows that the distinction between model and classification is sometimes not easy.

In the further parts of this thesis, the term “search action” is used to refer to a general notion of an interaction with a search interface—regardless of the classification that might be appropriate to use in the context.

Two types of models or classifications can be distinguished: latitudinal and longitudinal models. Latitudinal models distinguish between different types of single search activities at one point in the search process and longitudinal models distinguish between different stages in a search process.

2.2. Longitudinal models

This section takes a look at interaction models that can be seen as classifications that organize its classes along a timeline representing the search progress: Ellis’ interaction model, Kuhlthau’s stage model, and the interaction model by Spink.

2.2.1. Ellis’ interaction model

The first model presented here is the one by Ellis [47]. This model was developed after interviewing researchers from social science and psychology working groups at the University of Sheffield. Ellis described six “broader characteristics” of the features of the search patterns that the researchers reported: starting, chaining, browsing, differentiating, monitoring, and extracting.

Starting describes the stage where the searchers would try to find initial documents that help in any way with the search, either by looking them up in an IR system or by consulting their own collection of texts. The help provided by the initial documents could be ideas, references, or an overview of a subject area.

2.2. Longitudinal models

Chaining is to follow connections between documents and can be either “backward chaining” or “forward chaining.” Backward chaining describes tracing references in footnotes or the bibliography back to older documents and forward chaining is identifying newer documents that reference the one in hand (i. e. “forward” with regard to the time axis).

Browsing means to perform “semi-directed [...] searches in areas of [...] interest” [47]. An example for this activity is having a look at the tables of contents of conference proceedings or lists of relevant authors.

Differentiating between document collections—in Ellis’ terms, “sources”—is another characteristic. It often applies to discriminating between collections or single documents based on their quality, principle topic or other aspects and is used to focus the search on those collections that are most likely to contain relevant documents.

Monitoring is the act of keeping up-to-date with an information need. This is done by either manually searching for relevant documents from time to time or by configuring an IR system to automate this task.

Extracting denotes concentrating the search process on a single collection to find relevant documents.

These characteristics were further examined and expanded in later studies by Ellis and colleagues who interviewed chemists and physicists [48] and engineers [49], adding the characteristics *verifying* and *ending*:

Verifying is the checking of information in found documents.

Ending occurs at the end of a (long-term) search and includes comparing documents with documents found earlier

Wilson [141] noted that Ellis did not suggest any particular order of these “characteristics” and suggested organizing the characteristics sequentially using logic. He suggests that “starting” should be the first stage and “ending” the last and organizes the others according to their most likely place in the search process based on logical interdependencies.

2. Search Classifications

2.2.2. Kuhlthau's stage model

As with Ellis's model, Kuhlthau [88] examined the different stages a searcher goes through during a search and the feelings they experience in each stage. The stage model differentiates six stages: *Initiation*, *Selection*, *Exploration*, *Formulation*, *Collection*, and *Presentation*.

According to Kuhlthau, the searcher's task in the initiation stage is to recognize an information need and perform preparation for the search, e. g. by generating ideas about how to search. The initiation stage is dominated by uncertainty and apprehension. The selection stage is named in this way because the searcher has to select what to search for. During this stage, worries decrease and optimism sets in. The exploration stage is about exploring the search topic to gather enough information to be able to ask the right questions about the information need. Searchers in this stage of search often feel inadequate and frustrated. After learning enough about the topic of the search, users enter the formulation stage where they tend to feel more confident and have a sense of clarity. Searchers in this stage have learned enough about their information need to formulate concrete questions they are able to communicate to a search system. Following this, they enter the collection stage where they know how to specify what they are looking for and try to get the required information. They get an even stronger sense of clarity and feel more confident. The last stage is the presentation stage, which centers around using the information that was found during the search process.

2.2.3. Spink's IR interaction model

Spink presented a model [113] of interactions in the IR process. She suggested a layered view of the process, with the top layer being the search process as a sequence of search strategies. The search strategies themselves are sequences of cycles in the search process, each of which consists of one or more interaction dialogs between the user and the search system and ends with a result list. Part of the dialogs consists of the feedback that the user gathers from the search system and interprets regarding several factors like the relevance of the content and the terms or the magnitude of the result list. This model differs from those of Ellis and Kuhlthau in that it does not imply a start or an end of the process, but concentrates on the iterative aspect of the middle part of the process.

2.3. Latitudinal models

The previous sections summarized behavioral models that consider temporal organization of search actions in the search process. In contrast to these models, there are also models of search activities or search processes that examine the act of searching or seeking at a particular point in time without taking into account the stage of the search process. The following sections summarize Bates's model and the model of Belkin, Marchetti, and Cool [16, 17, 18, 32].

2.3.1. Levels of search activities by Marcia Bates

Marcia Bates described actions to extend a search, grouping them into four different levels: move, tactic, stratagem, and strategy [7, 9].

A move is a basic, atomic action by the user during the course of the search. These actions can be observable, but also unobservable (e.g. thoughts). Examples of moves are entering a term, clicking the search button, and considering options.

A tactic is a sequence of one or several moves that is goal-directed. Tactics have a specific intention with respect to the search. Entering a term might be described as a move, but entering a term in order to expand the query is a tactic. Other examples of tactics are replacing a term by an intentionally mis-spelled variant to find more documents and translating a term into another language to cover documents by foreign authors. Bates compiled a list of tactics [7] and extended the list with tactics on generating ideas to improve searches [6].

Stratagems are sequences of tactics and moves that exploit a structure that connects pieces of information in a given domain. One such structure is the citation relationship. If a searcher has a relevant document at hand, they can follow citations in both forward and backward direction to discover new documents that are relevant (see Section 2.2.1). Other structures that can be used are journal content tables and co-author relationships.

Strategies are complex plans for searches and include both stratagems, tactics, and moves. Strategies can be stated in advance only in simple cases, like known-item

2. Search Classifications

instantiation. In more complex cases, such as exploratory searches, where the search actions depend in part on things learned during the search [8] and are difficult to plan in advance, the plan of the search can only be determined after the search: then, the “strategy” is what the user happened to do during the search.

2.3.2. Belkin’s, Marchetti’s and Cool’s classification of information seeking strategies

Another example of latitudinal models is the classification of information seeking strategies (ISS) by Belkin, Marchetti and Cool [16, 17, 18, 32].

Belkin [16] stated that the traditional IR model consists of a static information need, a core process of comparing a query representation with a text representation, a one-step interaction process, and the idea that the user is outside of the system and merely a passive responder to the system’s output. He argues that this model is deeply flawed.

According to Belkin, one flaw is that the information need is not static but changes by engaging with the text. Another flaw is that the interaction is often different from the query-answer scheme prevalent in the traditional IR model. Instead, Belkin argues, searchers use a variety of information-seeking behaviors, of which only few are supported by existing information retrieval systems. Belkin concludes that an adequate IR model has to include the user as part of the system. This view seems to be supported by Ingwersen [71]. He noted that “direct and real information retrieval [...] is only possible by the individual user himself.”

Belkin, Marchetti and Cool [17, 18] developed a classification of search activities that they called “information-seeking strategies”, or ISS. Even though the word “strategy” sounds as if it is related to the strategy concept of Marcia Bates’ classification (see Section 2.3.1), the concepts are unrelated. An ISS is a single step during a search process. In the terminology of this concept, a sequence of multiple information-seeking strategies forms an information-seeking episode.

The variety of information-seeking behaviors, or information-seeking strategies, is divided into 16 categories along the four facets of “goal”, “method”, “mode” and “resource” [16, 18]. Table 2.1 shows the classification scheme.

Since the terms used in the classification are not exactly defined anywhere, the following paragraphs sum up what can be found in the literature. Remaining ambiguities are defined for use in this thesis.

Method

The value *searching* in the method facet is described as “looking for a specific known item” [18]. It is used to describe the same concept that also goes by the names of “navigational search”, “lookup” and “known-item instantiation”,

Scanning was described by Belkin et al. as “looking around for something interesting” [18]. This formulation is somewhat similar to the definition of the “recognition” value in the “Mode” facet given in the same work. Since it has to complement the “searching” value, it is not far-fetched to assume that it denotes “informational search” or “exploratory search”, which is defined as a search characterized by vaguely defined goals, changing and complex information needs or poor index systems [134]. Section 3.2 elaborates on the relationship between scanning and exploratory search.

Goal

The goal facet has the possible values “learning” and “selecting.” In this context, *learning* means to answer a temporary question in order to make progress with the actual search. It was described by Belkin [18] as “learning about the relevant issues”, but learning is also explained as “learning about some item or resource” [17]. As an example for learning, Wilson et al. mentioned the attempt to identify the key author in a given area [139].

Selecting, in contrast, is “identifying useful items” [18], e. g. by adding documents to a list, or by bookmarking or visiting web pages.

Mode

The facet “mode” can assume the values *specification* and *recognition*. These terms are explained as “searching for items on some identified topic” and “looking around” [18].

2. Search Classifications

Belkin et al. [17] drew the distinction between “looking for identified items” and “identifying relevant items through stimulated association”. In other words, “specification” can be described as “whatever found is relevant” and “recognition” as “knowing it when seeing it.”

The question arises whether “specification” means that the searcher is able to specify the searched-for information informally—to a colleague or in his or her mind—or if it means that the searcher is actually specifying the information need by interaction with the retrieval system. This discussion is closely related to the representation of the user’s problem as described by Mizzaro [96].

Mizzaro suggests that there is a hierarchical relationship between real information need (RIN), perceived information need (PIN), request and query, all of which describe to some extent the problem of the user. The RIN is the information need as observed by an omniscient observer. The PIN is the version of that information need that is perceived by the user. This might be closely related to the RIN (e. g. lacking some aspect) or entirely unrelated (e. g. in the case where the user misinterprets an observation). The user then formulates the PIN in a natural-language representation—the request. The request is then formalized as query, frequently a text-based formal representation of the information need used to communicate with the IR system.

In Mizzaro’s framework, each step—from RIN to PIN to request to query—is neither automatic nor lossless. As mentioned above, the user has to perceive the RIN correctly to have a PIN that is close to the RIN. The user’s ability to formulate the request based on the PIN is determined by his or her ability to think logically and, among others, to be proficient in a natural language. The ability to compose a suitable query, though, is influenced both by the user’s proficiency in the target query language and the IR system’s functionality. As an example, a person with an eidetic memory might be able to perfectly describe the cover of a book they are looking for. Unless the IR system provides a way to formulate a query based on such a description, the user is unable to communicate their request to the IR system, resulting in far less information in the query than would be possible considering the level of detail of the request.

Thus, the question is if the term “specification” relates to Mizzaro’s request level or the query level. This author contends that the ISS classification is about actions by search users: Belkin et al. called an ISS a “complex activity” [17]. For that reason, an ISS in this thesis is something a user does and what can be observed and interpreted.

Thus, “specification” refers to the act of specifying an information need at query level. There is also a pragmatic argument: the subject of this thesis is the support of users of IR systems and the IR system can, in many cases, only observe the query that the user enters. At that level, a “specification” is a query.

So, the term “specification” is defined in the context of ISS as submitting enough information in a query to accurately describe the information need. In other words, “specification” means to use some amount of the known information to identify missing information that is relevant, or at least to make an honest attempt at doing so, even if this attempt is unsuccessful.

Specification has to be differentiated from merely filtering the document collection in order to retrieve a manageable subset that is to be processed by recognition. In most cases, exclusively iterating over even a medium-sized corpus trying to recognize a wanted item is not feasible. In these cases, the first step of the searcher is often to narrow down the set of documents using a preliminary query which will contain information that is just enough to exclude numerous irrelevant items. This query will not be precise enough to exactly define what kind of document or information is needed. There is no exact separation between such a preliminary query and one intended as specification, but the fewer documents returned, the more the query has properties of an attempt to specify. This also means that an ISS that involves “recognition” is very likely to also involve some query to be issued. White and Roth [136, p.6] stated that “in exploratory search, people usually submit a tentative query to navigate proximal to relevant documents in the collection, then explore the environment to better understand how to exploit it, ...”

Another argument for viewing a broad “filter query” as part of a recognition activity—as opposed to being a separate step—is a *reductio ad absurdum*: If the specification of a filter query were a separate step and recognition solely consisted of actions entirely based on intellectual processing of result lists, then every single interaction with a retrieval system would be described as alterations between specification and recognition steps. In query-based systems specification would be entering a query, followed by recognition in the sense of looking at the result list. In faceted, exploratory search systems, such as mSpace explorer [60, 107] or Flamenco [146], specification would mean clicking on an item, followed by recognition being looking at the next output of the system. Also, the classification, being a faceted one, would imply values for the other facets. Since nothing else changed during the interaction, the other facets’

2. Search Classifications

values would be exactly the same in each of the two steps. So if the mode facet was interpreted in strict terms, each cycle in the human-computer interaction would be described by two different ISS classes that differed only in the mode facet and the order of the mode facet values would always be the same, essentially resulting in no information being gained by the facet at all.

Situations in which recognition is the only way for the user to succeed are the following: the user is not able to specify a vital part of the request (e. g. the book cover); the user is too busy to specify all synonyms and hyponyms of a term; the user does not know enough synonyms, hyponyms, and terms to differentiate from polysemic concepts but tries to infer from the document description if a document is relevant. In the first case, the specification might be impossible either due to technical constraints of the search system or due to a searcher's inability to memorize relevant details.

Resource

Another differentiation made in the classification is between “information” and “meta-information”, which is “information items themselves” and “descriptors or organization schemes of items” [18]. In another work using the ISS classification [17], Belkin et al. defined meta-information as “resources that describe the structure and contents of information objects”, while leaving “information” basically undefined.

Since the literature is vague concerning the definition of the values of the resource facet, the question arises as to how the facet values can be interpreted. The following two interpretations are possible ways to define “information” and “meta-information” in the context of an IR system: a) “information” is the artifacts themselves (e. g. books), while “meta-information” is the information about those artifacts that is stored in the index of the IR system; and b) “information” is the content of the artifacts (e. g. the text and pictures in a book), while “meta-information” is information about the content—e. g. author and title.

The first way to interpret this dichotomy is only plausible for digital artifacts like e-mail messages or digital-first publications. In cases where the artifacts are physical, such as books, using the resource “information” would never happen, since actual artifacts are nothing an IR system is concerned with.

Dimension	Values
Method of Seeking	Scanning, Searching
Goal of Seeking	Learning, Selecting
Mode of Seeking	Recognition, Specification
Resource Used	Information, Meta-information

Table 2.1.: Classification of ISS's

The second interpretation, saying that both information and meta-information regard the pieces of information that are searchable in an IR system, implies that information and meta-information are very similar: some meta-information, such as the title, is part of the textual content of a book and from the point-of-view of an IR system, both are stored in the search index, possibly just with different designations¹. Thus, viewed in this way, differentiating between meta-information and information is just a technicality, observable only in minor differences in the user's actions (e.g. using a different query form field or typing "title=keyword" instead of "keyword"). Both values refer to information that the searcher can access using the IR system.

In either way, the facet is of minor interest if the ISS classification is used for designing user interfaces for search systems: it either does not make any sense at all, as in one case, or it is redundant in the other.

Extension

The basic ISS classification has been extended to a classification of general interactions with information that also contains communication acts [32, 69].

Work done by Marcia Bates [9] shows that there are plenty of search methods information workers employ.

Yuan and Belkin showed [148] that an adaptive system that supports multiple ISS

¹For instance, when using Solr, the full text could be stored, indexed and searched using a "fulltext" field, while pieces of meta-information such as the title could be handled by similar fields like "title."

2. Search Classifications

ISS	Method	Goal	Mode	Resource
1	Scanning	Learning	Recognition	Information
2	Scanning	Learning	Recognition	Meta-information
3	Scanning	Learning	Specification	Information
4	Scanning	Learning	Specification	Meta-information
5	Scanning	Selecting	Recognition	Information
6	Scanning	Selecting	Recognition	Meta-information
7	Scanning	Selecting	Specification	Information
8	Scanning	Selecting	Specification	Meta-information
9	Searching	Learning	Recognition	Information
10	Searching	Learning	Recognition	Meta-information
11	Searching	Learning	Specification	Information
12	Searching	Learning	Specification	Meta-information
13	Searching	Selecting	Recognition	Information
14	Searching	Selecting	Recognition	Meta-information
15	Searching	Selecting	Specification	Information
16	Searching	Selecting	Specification	Meta-information

Table 2.2.: Complete list of ISS classes

classes is superior to statically configured systems. That study, however, was limited by the number of ISS classes supported by the improved system.

Criticism

The ISS classification is not without problems. Some of them have already been presented: lack of a precise definition for its terms and the difficult and hardly useful distinction between information and meta-information. In the form presented here, the classification includes the additional problem of not being able to describe all search actions that are possible. For example, it is possible to use Google to search for both meta-information and information of web pages² in a single search action, but there is no straight-forward method to describe this using the ISS classification. One way to solve the problem is to add a third value, “information and meta-information”, to the facet.

The relationship between method and mode

The facets method and mode describe seemingly similar things, a fact that Belkin et al. conceded [18]. One reason for this is that one definition of the “search” facet value is “looking for a specific known item” (ibid.), which is a choice of words closely related to the value “specification” in the mode facet. Additionally, “scanning” as well as “recognition” are defined using the phrase “looking around” in one of the papers [18].

This latter paper also introduces another source of confusion while giving examples to illustrate that these facets are indeed independent. The example for scanning by specification, for example, can also be understood as search by recognition. According to Belkin et al., an example for scanning by specification is that “one knows precisely what one is looking for, but not where it is located.” This example is in conflict with the definition of “searching” on the first page of the paper, where searching is defined as “looking for a specific known item.”

Below is an attempt by this author to illustrate the independence of the two facets by giving an example for each of the four possible combinations.

²An example for this is `intitle:lecture AND retrieval`

2. Search Classifications

Scanning and Specification A researcher wants to conduct a meta-analysis of papers about treatments for a rare disease. He assembles a list of relevant terms and collections and queries those collections for documents containing the terms. The documents found this way are examined for inclusion in the study. (Scanning, because it is not known which documents will be relevant. Specification, because the list of relevant terms specify the scope of the search.)

Scanning and Recognition A researcher is trying to identify new papers that might be relevant to his current research subject. He does so by browsing the Table of Contents of the latest proceedings of a pertinent conference. (Scanning, again, because the documents to be found are unknown in number and type. Recognition, because the researcher has no specific query but merely collects documents that he finds interesting.)

Searching and Specification A student is trying to buy a book that was recommended for a lecture in an online book store. The details (ISBN, author, title) are cited in the lecture material that is available to him. (Searching, because the book to find is known. Specification, because all information needed to find that book is known and can be communicated to the system.)

Searching and Recognition A user is trying to find a book on wine tasting in an online book store. She saw the book in a TV program but the book was only mentioned briefly and she cannot recall the exact name and author. So she narrows down the book list and scrolls down the list of hits to see if it is somewhere on that list. (Searching, because the book to find is known. Recognition, because there is not enough information available to the searcher to find precisely the desired book, but she has to look through a list of possible search targets to identify what she is looking for.)

A definition of ISS classes for the scope of this thesis

After summing up and analyzing available literature on the ISS classification, this section outlines some considerations regarding the ISS classes which are not further investigated here and defines those classes that are examined by this thesis.

The facets “goal” and “resource used” are not considered any further here. “Goal” is hard to operationalize since one aspect of it is learning and that is notoriously hard to

measure. “Resource used” is of limited value in modern IR systems, as contended in Section 2.3.2.

The remaining facet values will be defined as follows:

Specification The searcher issues enough information to the search system to narrow down the list of relevant items to only few ones so that finding the relevant items in the list is easy.

Recognition If the searcher uses a query at all, the query is broad and vague. The searcher’s main focus is on intellectually examining the found documents to determine if they fit the information need. The objects of the intellectual examination by the user are of the type of things the user is trying to find.³

Searching The searcher wants to find a known finite set of items.

Scanning The searcher cannot state a priori which or how many items will be relevant to the search or after how many found documents he will stop: it might be one document, but it might as well be 50.

The author of this thesis concedes that these definitions are also vague. One problem is that it is difficult to draw a clear line between a filter query with subsequent recognition on the list and a (bad) specification that makes the (badly) specified item appear farther down in the result list. In other words, it is hard to tell what has to be the maximum number of items in a result list so that one can assume that the search was done by specification. In either case, the user enters a query and then examines the result list. The experiment described in Chapter 7 dealt with this vagueness by using task descriptions that drew clear distinctions between the facet values.

The relationship between ISS classes and other views on search

There is a body of ongoing research on how best to support users in different search activities. Early examples include the work of Marcia Bates [7]. White and Roth [136]

³This last constraint is used to distinguish between the relevant recognition acts and small-scale recognition acts such as looking for the right button to submit the query. If every glance over the screen is recognition, then the the value “recognition” becomes meaningless.

2. *Search Classifications*

published a long treatise on exploratory search and how to support it. But the notion of “exploratory search” considers search on a much higher level than ISS classes.

An ISS is basically a pair of steps: first, to define a search and, secondly, to process the answer of the machine. By contrast, exploratory search consists of many iterations and different types of sub-searches and general information behavior. Exploratory search might include asking colleagues, browsing a possibly relevant journal, following some citations, clarifying the use of a newly found term using a short web search, and saving relevant documents along the way.

The whole exploratory search cannot be expressed as a single ISS but it can be expressed as a single “strategy,” to use Bates’ terminology. Bates’ strategies, however, consist of “stratagems,” “tactics” and “moves,” none of which is close to an ISS: Moves are too finely granulated, containing even single key strokes. An ISS describes much more than actions on this level. Even tactics do not fit exactly, since they are granulated more finely than an ISS: while an ISS comprises two steps, query and result list examination, the query formulation alone might consist of multiple tactics. And stratagems are too coarse-grained. They consist of many actions and require the exploitation of a structure between pieces of information. An ISS usually consists of several actions (e. g. tactics), but is not required to exploit a structure. So ISS’s might be located between tactics and stratagems.

The facets of the ISS classification are independent by definition and the remaining facets, method and mode, refer to different kinds of interaction with the search system. Moreover, the values within each facet are also independent by definition. For this reason, it is assumed that support for each ISS class can be composed by independent support mechanisms for the involved facet values. The following chapters examine support mechanisms for each value of the relevant two facets of the classification, scanning, searching, recognition and specification.

3. Support for Scanning

As defined in Section 2.3.2, scanning denotes the act of trying to find an unknown number of unknown documents. While each ISS, and therefore the term “scanning”, only describes a single step in a search process, the intent (not necessarily the methods) of a single scanning step is similar to that of exploratory search, which is the search for an unknown number of unknown things with a high degree of vagueness in the information need. This close relationship is exploited in this chapter because the ISS classification, and therefore the concept of scanning, is rarely used in studies of support mechanisms. The body of literature on exploratory search, however, is sizeable and it is hoped that the findings on exploratory search can be applied to the question of scanning.

The following sections initially provide a general round-up of exploratory search and elaborates on the relationship between exploratory search and the scanning facet value. After that, several support mechanisms that were proposed for exploratory search are described and examined concerning their suitability for the facet value scanning.

3.1. Exploratory search

Marchionini [94] divided search into three parts: lookup, learn and investigate. The latter two in combination are what is usually termed “exploratory search”: “learn” is the activity aimed towards acquiring new knowledge and “investigate” is finding new things to learn.

Lookup searches are generally thought to be easy: the searcher often knows important properties of the item to look up, the desired items are few and well-defined, and the searcher knows when all items are found and the search is complete. Exploratory

3. *Support for Scanning*

searches, by contrast, are generally not easy. As the extent of vagueness in the information need is high, the searcher is often not very familiar with the subject matter, and it is unclear when to end the search.

In the past, several solutions for problems related to exploratory search have been studied.

3.1.1. The evolving information need and sense-making

White and Roth [136] stated (p. 12) that exploratory search is characterized by an evolving and changing information need. They argued (pp. 52–59) that tools for exploratory search should help the user make sense of information, define the problem, support multiple sessions, provide progress updates, explanations for system actions and summaries for major themes in encountered information.

According to White and Roth, key parts of exploratory search are learning and investigation (p. 13). It is a combination of analytical strategies (lookup searches) and browsing (p. 13). The behavior associated with exploratory search is akin to wayfinding. It may be more concerned with recall than precision, so current web search engines are not well suited for supporting exploratory search (p. 15). Since exploratory searches span multiple queries that reflect the evolution of the information need, systems designed to support exploratory search should offer features for query specification and refinement (p. 17). White and Roth stated that berrypicking is a common strategy in exploratory searches (pp. 7 and 29, see also Section 3.4.1). They argued that cognitive aspects (p. 37) are important since the extent of vagueness at the beginning of the search and the learning activities needed to overcome it are high, and both involve cognitive effort on the searcher’s side. To help the searcher make sense of information found during the search, systems should offer visualizations and use context. White and Roth assumed that, for some of these issues, clustering might provide a good solution (p. 44).

3.1.2. Use cases

Patent search is an area of search that is often exploratory: it is characterized by a great amount of vagueness caused by intentional linguistic obfuscation of patent

3.1. Exploratory search

specifications, and searches may take days to complete. A study of the requirements that patent analysts have for search systems [4] found that, among more mundane features like Boolean operators, query expansion was rated important by 56% of the participants and query translation by 46%.

Even systems for exploratory search of personal information, like Phlat, have been developed [34]. Phlat supports exploration by integrating search and browsing on both content and meta-information. Since the amount of information people store on personal computers keeps increasing, personal information management is no longer an issue of just re-finding things previously known, but also of finding things that have been stored but have not yet been used.

3.1.3. Needed features

From an engineering perspective, it is not enough to know how important exploratory search is for searchers. In order to build exploratory search systems, the features needed by users have to be determined.

White and Roth [136, p. 41] listed eight features that exploratory search systems must (sic!) have:

- Rapid query reformulation
- Result filtering using facets and meta-information
- Making use of search context
- Visualization of the collection
- Support for learning and understanding
- Collaboration support
- Session support with search histories
- Task management

3. Support for Scanning

Vakkari argued [129] that exploratory search systems should support the user in structuring their search process, a requirement that is in agreement with White and Roth, who called for task management in these systems. The support mechanisms might also depend on the target group of the search system: Loizides and Buchanan [90] reported a study that found novice academic searchers rely heavily on document titles in result lists while triaging sources for a research proposal. In this case, searchers might benefit more from adding keywords to surrogates than from sophisticated session support features.

3.1.4. Subjunctive designs

Exploratory search can not only be supported by adding specialized tools to existing interfaces or by introducing completely different interaction paradigms, such as browsing, but also by augmenting existing, traditional interfaces. Bron et al. [28] examined an exploratory interface for media studies researchers and found that a subjunctive design helps users. Subjunctive interfaces allow the users to follow alternative leads at the same time [92]. Villa et al. used a similar design [133] for working on multiple aspects (subordinate information needs) at once and found significant improvements over a baseline interface.¹

3.1.5. Trailblazer

Nitsche and Nürnberger [98] described an exploratory search interface that used keyword search as its conceptual basis and focused on the visualization of search paths along found documents. Besides formative evaluation, no hard data was reported on efficiency measures or comparisons with known interfaces.

3.1.6. mSpace

One system that caters explicitly to exploratory searchers is mSpace [60, 107]. mSpace makes use of ontologies and provides a user interface for browsing the knowledge domains for which semantic information is available. The interface uses faceted browsing

¹One possible problem with this study is that the statistical significance was measured using Wilcoxon's rank-sum test on 12 differences but no mention of having corrected for multiple comparisons (and thus, alpha inflation) was found.

3.1. Exploratory search

(see Figure 3.1) to let the user navigate through a multi-dimensional information space. Previews of documents and categories give the searcher an easy overview over the information landscape.

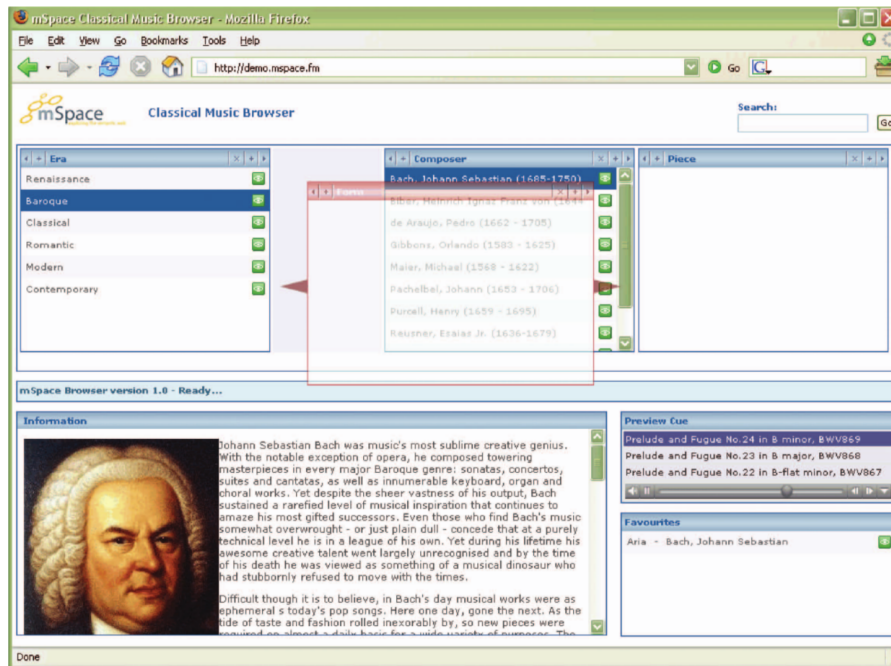


Figure 3.1.: The mSpace browser

3.1.7. Flamenco

Flamenco (see Figure 3.2) is another system that uses a faceted browsing interface. It operates on image meta-information and also offers query-based search. Yee et al. studied how well the system compares to a keyword-search baseline in terms of several satisfaction measures as well as recall-oriented success measures [146]. They found that users were more satisfied and successful with the experimental interface, but also needed more time to finish the tasks.

3. Support for Scanning



Figure 3.2.: Flamenco interface later on in a search [146]

3.1.8. MedioVis

An effort to build a complex system for working with multimedia items is the Medio-Vis system (see Figure 3.4). It incorporates Shneiderman's visual information seeking paradigm [110] and integrates multiple synchronized zoomable interfaces into one complex system, configurable by the user [66].

HyperGrid

One of the zoomable interfaces in MedioVis is HyperGrid, presented by Jetter et al. [77]. It is an interactive table representation of documents aimed at exploratory search [101]. The initial view of HyperGrid looks like an ordinary table with cells of equal height. In the table, each row represents an item (e.g. a movie) and each column

3.1. Exploratory search

represents an aspect of each item, such as business data or content. Following Shneiderman’s zooming paradigm, HyperGrid presents coarse summaries of each aspect in the initial state. For example, the content column might list one movie as “Drama.” The user can zoom into each cell, making HyperGrid give more details of the cell’s column for all rows, but especially for the one cell zoomed into, and for all columns of the row zoomed into. For instance, the movie initially summarized as drama would be listed with detailed information on genre, plot and target audience. On the last zoom level, a cell would be detached from the table and changed into a browser view. See Figure 3.3 for a screenshot of this state. The user study reported in the paper was limited by the small sample (five participants) and focussed towards user experience.

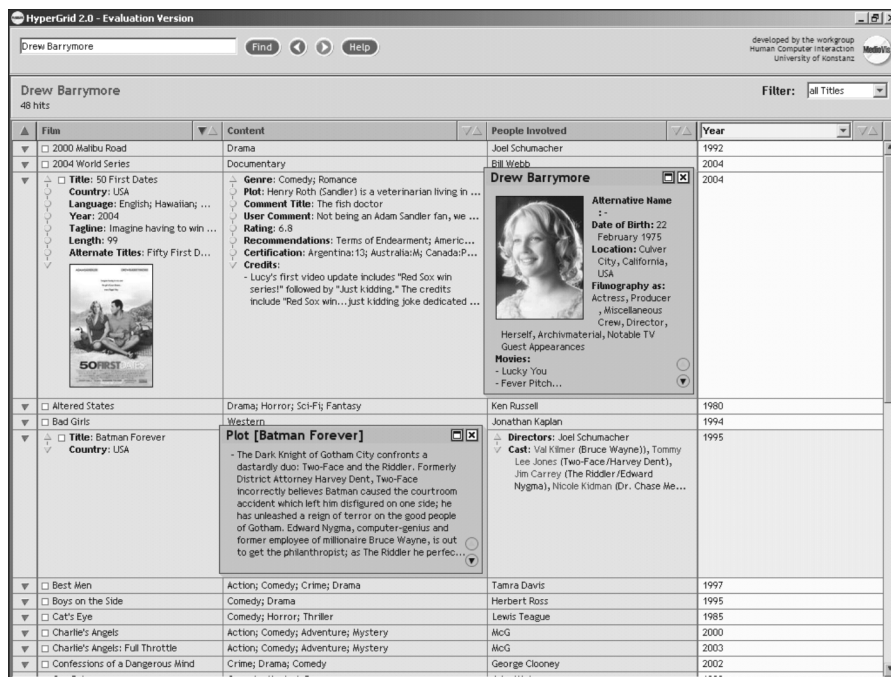


Figure 3.3.: HyperGrid with a detached browser view [77]

Similar visualizations

Gerken et al. [53, 54] discussed many visualizations and systems (LevelTable, Circle Segment View, Parallel bargrams, INSYDER, MedioVis/HyperGrid, VesMeB, Hyper-

3. Support for Scanning

Scatter, Fisheye, network visualization, ZOIL) used in information search interfaces on a conceptual level. In a two-week exploratory remote study, the participants reported that HyperGrid is “better suited to searching for one specific object” [54, p. 58] than HyperScatter, because the table presentation offers a better overview of the data².

MedioVis compared to a web library catalogue

Grün et al. [58] compared MedioVis with a university library search system (KOALA) using 24 university students conducting six fact-finding tasks with each interface. They found that users were faster completing the tasks with MedioVis than with the KOALA interface and more satisfied with MedioVis, but did not report detailed reasons for the differences.

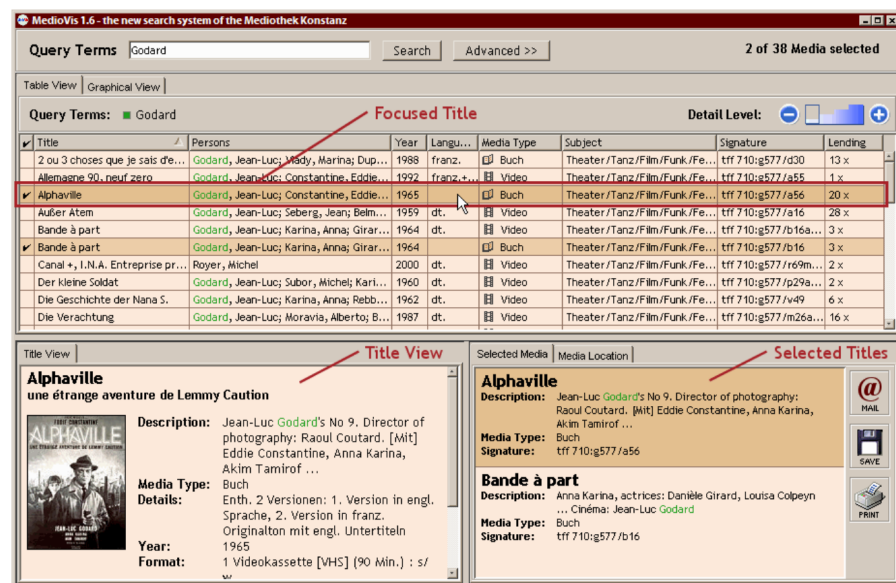


Figure 3.4.: MedioVis main screen [58]

²“Als Gründe für die Bevorzugung der HyperGrid wurde genannt, dass hiermit eine bessere Suche nach einem spezifischen Film oder Schauspieler möglich ist, da die Tabelle einen schnelleren Einblick in die Daten liefert.” [53, p. 8]

Conclusion

MedioVis bundles several innovative ways of visualizing document collections that improve aspects of exploratory search. Due to issues of study design, the exact causes for these improvements are not known. Furthermore, the interaction in MedioVis focuses on browsing. Since this study focuses on the query paradigm, the MedioVis findings are difficult to apply to it.

3.1.9. The source of improvement by browsing

The exploratory search systems shown here are all browsing interfaces that need either special meta-information, such as ontologies, or meta-information rich data, such as that from the movie database, to be applicable. Hughes-Morgan and Wilson [68] noted that faceted searching and browsing, amongst other new interaction styles, actually often need new meta-information. They posed the question if any advantages of browsing interfaces over traditional search query boxes come from the interaction style or from the metadata. Under control of the metadata, they found that clustering improved task times significantly during simple tasks and exploratory tasks, so the interaction style itself improves the outcome, not the additional metadata.

3.1.10. Evaluating exploratory search

Studying exploratory search is inherently complicated since the usual paradigm of assigning search tasks to participants does not work well for the volatile nature of exploratory search sessions: it is hard to simulate the shifting information need in a laboratory setting. White and Roth suspected [136, p.63] that time on task might be less important in the evaluation of exploratory search, as this kind of search involves a high amount of learning. They concluded that learning time might be a better measure and that [136, p.15] recall might be more important than precision. However, later in the same work they noted that, depending on the difficulty of the task, recall might be as important as precision [136, p.63]. Other metrics they deemed important are engagement and enjoyment, information novelty, task success, task time (sic), learning, and cognition [136, pp.64ff]. Beresi et al. [22] examined a graphical way to represent and analyze relevance criteria used by participants in exploratory IR studies. They

3. Support for Scanning

also suggested that these techniques be used to find common relevance criteria in exploratory searches, and to adapt search systems to use these criteria for ranking.

3.2. The relationship between exploratory search and scanning

Exploratory search is a complex activity consisting of many subactivities that may span a long time. Systems that are intended to support exploratory search need specific features that go beyond simply processing queries and listing found documents. The ISS facet value “scanning” shares some of the properties of exploratory search: Both exploratory search and scanning involve a relatively high degree of uncertainty: In the ISS classification, scanning is the facet value complementary to searching, which is the attempt at finding something known. So, scanning means finding something unknown, which implies uncertainty with regard to important aspects of the search targets [20]. Because of this relationship between exploratory search and scanning, support mechanisms for exploratory search might also be applied for scanning.

In the following sections, some support mechanisms are examined that were suggested to support exploratory search in query-based systems. These mechanisms are thought to also support scanning and can be included in a keyword search system without introducing an entirely new interaction paradigm.

3.3. System-oriented support

To support users in exploratory searches, the search system can be extended in two places: on the user interface level and in the search engine part. Examples of the latter are recall-oriented searching and query term expansion.

3.3.1. Recall-oriented searching

The core retrieval models are independent from questions like scanning or searching; they just consider the similarity of a query representation and the representation of

3.3. System-oriented support

documents. The preprocessing pipeline, however, involves steps that can be omitted or configured in different ways. This is true in particular for the normalization step.

Terms are usually normalized to make searches more robust against spelling variations (British English: “colour”, American English: “color”; “inflection”, “inflexion”), inflection (“color”, “colors”, “colored”, “coloring”) and compounds (“colorcode”, “color code”, “color-code”), phenomena that can even occur in combination (“colouring code”, “color-code”). The idea is to map all of these related words to the same type in the index to remove the burden of explicitly stating all possible term variations from the user. Sometimes this mapping goes awry and connects unrelated or just remotely related concepts, e.g. by using stemming instead of lemmatization: while lemmatization maps each word to its dictionary form (nouns to the singular, verbs to the infinitive), stemming maps a word to its stem. This latter mapping is much more aggressive, because it might map merely remotely related terms to the same stem (e.g. “computation” and “computerization” to “comput”). Thus, a user who is interested in the question of the computerization of the work environment might not be interested in all aspects of computers at the workplace, but, due to stemming, the query `computerization work` might also find documents about the latter subject area. These documents might still deal with the issue, even if they do not mention “computerization” directly. So, to help with scanning searches, a retrieval engine could intentionally switch from lemmatization to stemming. This would sacrifice some precision to increase the recall.

Another example, from the field of meta-search, is the translation of a user query to the query languages of remote resources. Consider a query for documents containing a term and a remote resource that has a full-text index alongside of keywords. The query could be translated verbatim, searching only for the terms in the full-text index. In order to increase the document yield, the translated query could contain the search term also in the keyword field. In this case it would be possible to find documents that do not use the term itself in the text but still cover the subject (e.g. foreign-language texts).

3.3.2. Query term expansion

The previous techniques do not change the original user query; only the decisions in the system design, which normalization to use, how to transmit queries to remote

3. *Support for Scanning*

systems, are made so that they lean toward greater recall.

To achieve greater recall, the user's query can also be explicitly altered, e. g. by query term expansion. Expanding query terms can benefit the user in open search tasks with vaguely defined information needs and when the knowledge domain is new to the searcher.

It depends on the search engine's implementation and the IR model used how term expansion can be leveraged in open searches. If the search engine uses some kind of Boolean subsystem (e. g. for filtering documents prior to ranking), query terms can be expanded by additional terms both disjunctively (using an **OR** operator) and conjunctively (using an **AND** operator). The **OR** connected expansion terms can be used to broaden the search if the expansion terms are chosen correspondingly. Synonyms and frequently cooccurrent terms are good choices for this, but also antonyms, meronyms holonyms, and even superterms can be used. These choices relate to search tactics discussed by Bates (see Section 2.3.1).

Query expansion can be performed at system level, but also at user interface level. Automatic query expansion at system level has been studied by many authors. Efthimiadis [45] collected some methods. Even though this approach is interesting, it is not further detailed here since the focus of this thesis is not so much on how to calculate good query expansions but rather on the communication between search system and user. If the query expansion is communicated with the user on the user interface level, it is called Interactive Query Expansion. Some of its aspects are summarized in Chapter 6 in the context of support for specification.

3.4. User-oriented support

While the previously discussed support mechanisms are system-oriented and affect mainly the internal workings of the IR system, some support mechanisms for scanning concern the interaction between user and search system.

3.4.1. Berrypicking tray

Marcia Bates [8] used the metaphor of berrypicking to describe information seeking. In Bates' opinion, the traditional system-oriented model of IR is not a good model for actual searches since it "represents some searches, but not all, perhaps not even the majority." Bates argued that actual searches are much more iterative and comparable to picking berries: A person picking berries wanders from bush to bush, picking berries that look tasty here and there, instead of traversing the whole forest in search of the best bush. Similarly, searchers issue multiple queries that either change gradually (exchanging a keyword with each attempt) but can also be composed of entirely different terms, connected only by the underlying information need. With each query, the searcher might find new documents that they use to learn about their topic and to advance the search in new directions. For supporting berrypicking searches, Bates suggested that search systems should have a facility to store documents:

The interface design should make it easy to highlight or otherwise flag information and references to be sent to a temporary store. Said store can then be printed out when the searcher is ready to leave off searching. The necessity otherwise either to write information down by hand or print out information in bits and pieces interspersed between search commands would be tiresome and would reduce search effectiveness.

Marcia Bates [8]

O'Day and Jeffries [99] studied clients of librarians and found that also in this setting searches were interconnected and the result set was accumulated over all search sessions, instead of being the result of the last search. White and Roth, too, argued that berrypicking is a concept that exploratory search systems should support [136, p. 7].

Unfortunately, there is not much literature that supports a berrypicking feature for exploratory search with quantitative data. Schraefel et al. examined Hunter Gatherer [106], a web-based tool for collecting snippets found in web documents, but did not report effects beyond user satisfaction. A similar browser extension for collecting snippets from web pages used extraction patterns [40]. The authors reported only an accuracy-based evaluation of the pattern extraction engine and results from a small-scale ($n = 9$) qualitative user study. Bharat [23] described a web-based system

3. *Support for Scanning*

called SearchPad, that supports berrypicking to some extent. SearchPad supports web searchers by offering ways to keep track about found documents and the queries that found them. The system was tested in a four-month observational study using logs. The only data collected was the number of result pages visited by SearchPad users and how many users the system could attract during that period.

Kerne et al. tested the combinFormation system against Google and Word for academic teaching tasks that involved exploratory search [86]. combinFormation is a creativity support tool that incorporates meta-search using several general-purpose web search engines. They found that students using combinFormation achieved better grades than those using Google and Word. A problem with this study is that two very different systems were compared without examining the exact source of the performance difference. Another problem is that no information on how the study was blinded was provided. This is not an unimportant factor because the performance was measured, among other methods, by the grades the students received from human teaching assistants. If the teaching assistants knew which experimental condition each student was assigned to, their grading could have been influenced by this.

All in all, there seems to be a trend towards including a berrypicking tray into systems to support exploratory search, but there does not seem to be any empirical evidence that allows estimating a cost-benefit ratio.

3.4.2. Result list support mechanisms

Other ways of supporting exploratory search include using a result list presentation whose design is optimized for exploration.

Relevance indicators

Shani and Tractinski examined the effect of different types of relevance indicators in result lists [109] and found that inserting graphical indicators (bar graphs) into result lists makes users examine more results in an exploratory search.

Adaptive visualization of the result list

Ahn and Brusilovsky studied an adaptive visualization for exploratory search [2] using a two-dimensional representation of the similarity of documents to each individual query term. In the visualization, the query terms are placed as draggable labels on a plane. Represented as dots, documents are placed in such a manner that their position relative to each query term is a function of its similarity with each term. The user could drag terms, see which documents moved the most and infer the grade of relatedness. The system was found to be liked less than the baseline system but having better nDCG@10, while P@10 was not significant. The paper seems to be limited by its unusual use of precision-oriented measures for an explicitly exploratory search setting.

Automatic facets

Hearst [64] compared result list clustering and hierarchical faceted categories (HFC). In her view, clustering has benefits in the process of “clarifying and sharpening a vague query” and in term disambiguation, but clusters are hard to predict and labeling is still problematic. HFC is a faceted classification whose facets consist of category hierarchies. The advantage of HFC is that several aspects can be specified independently while still preserving the power of hierarchical classifications. The downside is that creating an HFC still requires some manual work [65, 115].

Tag clouds

A tag cloud is a representation of the frequency of tags in a document collection. One popular design variant orders the tags alphabetically; the frequency in the corpus is represented by the font size each tag is printed in. Tag-cloud-like designs can also be used to summarize documents in a result list. Schrammel et al. [108] found no significant interaction between the way a tag cloud is laid out (four methods were put to trial) and the time users needed to complete search tasks.

Markers for documents already found

An idea proposed by White and Roth [136, p. 57] is to help the user keep track of the progress they made during their search. Approaches to this are possible at both result

3. Support for Scanning

list level and document level.

Malik [93, p. 79] examined search in structured documents. In one experimental system, sections already visited by the user were not marked as such. 24 participants made negative remarks about this. In the study with the revised system, that included markers for these elements, three users remarked that they found this feature helpful.

Golovchinsky et al. [55, 56, 57] examined search systems that had “retrieval histograms” included, icons in result lists that show how relevant a document was in past searches. Since these histograms have never been tested individually, it is not known how much they affect user performance.

3.5. Conclusion

Scanning is related to exploratory search, because both involve information needs with an increased amount of vagueness, resulting in searches whose end is not easy to predict. Some support mechanisms for both the user interface and the system level were summarized. It seems that browsing interfaces offer good support for exploratory search. Unfortunately, scanning is an activity that is defined in the context of search systems using the traditional query-result paradigm. For open searches in this paradigm, only scarce data is available regarding the effectiveness of possible support mechanisms.

4. Support for Searching

The value “searching” in the method facet describes trying to find a single, or very few, possibly known items (see Section 2.3.2). This is the baseline activity of all searching that is widely supported by many IR systems and web search engines.

However, in the ISS classification “searching” describes only how many and what kind of search targets are involved in the search task. The searcher can and must also choose how the search is to be performed: by specification or by recognition. While the support mechanisms for these aspects will be examined in the chapters following this one, the current chapter examines support mechanisms for searching alone.

4.1. How searching relates to the other facet values

When trying to find support mechanisms for searching, the question arises what differentiates searching from scanning from the point of view of the user. As detailed in the previous chapter, scanning is about finding multiple items, regardless of how the search is performed. So support for both scanning and searching neither includes support mechanisms for query formulation (specification) nor visual search (recognition), because this is subject of a different facet. Mechanisms for open-ended searches (scanning) are also irrelevant. While finding multiple items is inherently more complicated than finding a single document, the two types of search can still be considered related: The act of finding multiple documents (scanning) can be thought of as multiple acts of finding a single document (searching). This means that support mechanisms for searching might also benefit users engaged in scanning activities, but not necessarily the other way around.

4. Support for Searching

4.2. Query biased summaries

Current IR systems¹ are both unable to find all desired documents and to find only the desired documents in the first iteration of the query. Latter fact is the reason why searchers have to scrutinize each result list to check if the search target is present in the list, and where. This is, in the general case, also true even if the searcher formulates a correct and precise query. This means that the act of checking individual document surrogates in the result list is an activity that occurs in all searches, regardless of method, mode, and other factors.

To help users evaluate whether a given document is indeed a good fit for the query, Tombros and Sanderson [119] examined query biased summaries. Prior to their study, IR systems usually displayed the title and the first few sentences of each document along with other pieces of information to summarize a hit in a result list. Query biased summaries are document summaries that are composed of document sentences that are relatively similar to the terms in the query.

The benefit of query biased summaries over displaying just the first few sentences is that the searcher can see the context in which the query terms appear in the document. Tombros and Sanderson found that this benefit translates to an improvement in judging the relevance of documents: Statistically significant increases in both recall and precision were measured.

4.3. Conclusion

Searching is the baseline activity that all search systems are aimed for. The difference between searching and scanning is that searching has a very narrow goal, a goal so constrained that it is difficult to find support mechanisms specifically for searching that do not also support scanning. The query biased summaries introduced in this chapter do support searching. That they also support scanning underlines that searching can be considered a baseline activity for search systems.

¹And we can safely assume, for reasons inherent with natural language and the act of searching itself, also future IR systems

5. Support for Recognition

Recognition, as defined in Section 2.3.2, refers to the act of identifying relevant documents by intellectual examination. Support for recognition is something that lets a user better recognize items of interest. Enabling recognition, which makes recognizing possible in the first place, stands in contrast to this. Showing a cover image in a book search engine, so that users can recognize the book by its cover, is enabling: Without showing the cover, recognizing books by their covers would be impossible. Supporting recognition for book covers would improve on the baseline of just making it possible. But recognition does not only refer to images; texts, such as titles and author names, can also be recognized. This chapter describes two experiments that explored ways to support recognition for text-based tasks.

5.1. Introduction

Specification is an act of communication between the user and the computer that is observable from the outside. Whereas the act of recognition is an activity that cannot easily be observed since it occurs mainly in the cognitive space of the user. It can be considered as an example for human-based computation, where the computer outsources tasks that are difficult for a computer but easy for a human user to handle. In the case of recognition, this outsourced task is the relevance assessment of documents. The difficulty that a computer has with recognition activities is that, by definition of the ISS class “recognition”, the computer does not have sufficient information about what the user is looking for (see Section 2.3.2). This, in turn, means that making a computer substantially support this activity is not easy, since the computer probably does not have enough information about what the user is interested in.

While it is generally not easy to support recognition, there are tasks imaginable for which the computer could really offer support. One such task is the search for a

5. Support for Recognition

book that shows a specific person on the cover. A support mechanism could filter for books that show a face on the cover and enlarge the face or the whole cover for better perception. The problem with this approach is that the user would have to communicate their task, i. e. searching for a face, to the computer. This would mean to offer some way to specify the search target, and the boundary between this specification activity and the activity of recognition, actually under scrutiny, would be difficult to draw. To avoid this discussion, this chapter examines only instances of recognition where the difference to specification is rather clear.

Considering all the difficulties with recognition-type activities, the question arises if supporting recognition is worthwhile at all: If recognition were a fringe case, developers of search systems could safely sidestep the whole issue.

In interactive information retrieval, a query is the result of a process that starts with a real information need (RIN, see Section 2.3.2). The problem that the user is often not fully aware of their RIN is an obstacle the user has to overcome when creating a query. The other transcoding steps (PIN \rightarrow request, request \rightarrow query) can be looked at from the viewpoint of communication theory. One model in communication theory is that two parties exchange information using some kind of code to transfer an internal state (e. g. a thought) from one party to the other. This process can only succeed without loss of information if the code used by both sender and receiver is identical. In the given situation the user realizes to have some kind of information need. They might not even be able to express this thought verbally in their natural language to communicate it to a hypothetical human expert. For instance, the information need might involve a book whose details the user has forgotten. They remember only the general subject area and some aspects about the cover design. Not being able to remember exact details makes it hard to verbalize the information need, even if the user was a skilled painter. And even if the user was able to paint the book cover, the IR system might not offer a way to specify it. Not sharing a common code, the user would not be able to communicate the information need to the IR system.

Such problems can also occur in the realm of textual search. A user might remember an academic paper about some topic and that the author had a name that sounded Chinese. Now, the user can formulate a request about that paper but the vague concept of Chinese-sounding author names is hard to explain to a machine.¹

¹And even human beings can have different concepts of Chinese-sounding names.

5.2. Related work

If users are unable to communicate their information need to the IR system, regardless of reason, they can try to put as much information into the query as possible and then iterate through the result list trying to find the document they are looking for. In cognitive psychology this process is called “visual search” and has been studied by many authors. In this field, the search process often focuses on finding graphical items in a two-dimensional space, such as target markers in a geographic map used by pilots. While a few properties of visual search are known, supporting users of IR systems engaged in visual search activities is at best poorly understood. To study possible support mechanisms for users engaged in visual search in the context of information retrieval, two exploratory studies focussing on text search were conducted.

In the following section, some relevant studies from cognitive psychology are summarized. Then, the two studies exploring different approaches to support for visual search in the context of interactive information retrieval are presented. The first of these studies tried to find out which of two result list variants is better at supporting visual search in result lists. The second study extends on this and compares the two result list variants with a basal result list that lacks any advanced features. Because the results of the two studies were inconclusive, another analysis is presented that examined data from both experiments merged into one data set.

5.2. Related work

Looking at result lists is a very common action of users during a search. Xie and Joo [145] found that examining result lists is one of the most frequent search actions, albeit concerning web search. Tran and Fuhr [122, 123] developed a Markov model of gaze transitions in a (non-web) search setting. They reported average durations and transition probabilities between the query form, result list items, the basket (used for collecting documents) and the document details. Considering the steady state of their Markov chain, users spent 54% of the time working with the result list.

Information-seeking by visual search has been studied extensively in the field of cognitive psychology. Smith [112] studied visual search on maps with objects that contained both textual and graphical elements. Subjects were asked to search items on a 2D map that consisted of an uppercase letter, a three-digit number and a vector. They were

5. Support for Recognition

given the letter and initial two digits and asked to report the third digit. The vectors were added to increase the visual clutter. In one condition the items were color coded so that each letter mapped to a different color. In the other condition the color coding was absent. He found that increasing the number of objects increased both the time needed for each task and the error rate. Introducing color coding decreased time on task and errors. Beck et al. [10] studied searching on aeronautical maps: Participants had to search for markers on cropped aeronautical maps. For targets of low saliency, visual searches are slower if the global clutter (the number of different identifiable things) or the number of distractors is greater. They report that targets of high saliency are located in the same time regardless of the amount of local and global clutter. Local clutter distracts more than global clutter.

Some researchers (e.g. Townsend [121] and Treisman and Gelade [124]) made a case for the distinction between serial and parallel search. In this context, parallel search is the act of finding a number of search targets in an instant, so that the time needed does not depend on the number of items to find. In serial search, a searcher finds each target one by one, so that the time needed to find all targets depends on the number of targets. Wolfe [142], though, conducted a meta-study of “a few hundred trials” of visual search and found no evidence for the hypothesis that visual search can be divided into serial and parallel search. Duncan and Humphreys [43] studied how participants search for certain geometric shapes. They came to a similar conclusion as Wolfe when they found that the difficulty of the task increases continuously with increasing similarity between targets and other shapes (distractors).

Forlines and Balakrishnan studied visual search [51] using images containing letters. They examined different ways to present low-prevalence search targets to a visual searcher. They concluded that it is best to present smaller parts of an image sequentially (reduction of false-negatives by 60%) followed by reordering into a layout that is easier to follow with the eye (reduction of 30%) and the generation of composite images that—hopefully—contain more search targets than each individual image alone (reduction by 28%).

An interesting property of visual search is the prevalence effect studied by Wolfe et al. [143, 144] and other researchers. The prevalence effect is the phenomenon that searchers miss more targets when the targets are relatively few (e.g. 1% of all objects) than when they are plenty (e.g. half of all objects). Wolfe et al. showed that this is not an issue of the searcher becoming less vigilant: different searchers tend to miss

5.3. Experiment 1: linear and table-based result presentation

identical targets. The effect could be mitigated by inserting “bursts of high prevalence” with feedback [144].

Fleck and Mitroff [50] found that participants were able to report rare targets more reliably if they were offered a chance to revert a “present/not-present” decision. Doing so, however, has an impact on task completion time so the prevalence effect might just move from missing rare targets to a higher time effort to find them.

The study reported by Menneer et al. [95] found that visual search incurs costs associated with dual-target search. This implies that retrieval by visual search might be governed by different mechanisms if conducted for a single item or for multiple items.

The previously summarized cognitive science studies examined visual search in two-dimensional spaces (e. g. images and maps). Because of this, the findings might not be directly applicable to the examination of result lists. Studies in visual search that consider result lists could not be identified. The study related most closely to information retrieval was the one by Duggan and Payne, who examined skim reading [42] of texts rather than lists. Their findings are also not directly applicable to recognition in result lists because they focussed on memory-oriented questions and not on finding things.

While not all of these findings are directly applicable in the context of text retrieval, there are some conclusions to draw: One result of the research done in cognitive psychology is that visual search is difficult—especially if there are few targets and many distractors, which is the case in a usual result list. Since the search targets in text retrieval look very much like all the distractors (after all, all of them are text), we cannot hope for recognition tasks to be easy.

5.3. Experiment 1: linear and table-based result presentation

The first of the two studies reported in this chapter examined two different assumed support features for visual search tasks. These two experimental conditions were not compared to a baseline to simplify the experiment design and because it was assumed that each of the two versions was an improvement over the (assumed) baseline of not supporting visual search at all.

5. Support for Recognition

1	Arunamata, A Punn, R Cuneo, B Bharati, S Silverman, NH	Echocardiographic Diagnosis and Prognosis of Fetal Left Ventricular Noncompaction.	2011
2	Chong, Yih Tng Chen, Chun-Hsien	An Investigation into Dynamic Customer Requirement using Computational Intelligence.	2009
3	Bowick, GC Barrett, AD	Comparative pathogenesis and systems biology for biodefense virus vaccine development.	2010

Figure 5.1.: Screenshot of the table variant

5.3.1. Experimental conditions

During visual search on result lists, the user alternates between locating the next possible target (e.g. an author name) and evaluating its relevance. Since the system does not know enough about what the user is looking for, it seems difficult to support the relevance assessment itself. Instead, the focus of support in this experiment is on locating all the possible targets. Two result list design variants were created for the first experiment: a list with highlighting and a table-based variant.

Table-based result list

Jetter et al. developed HyperGrid, a table-based result list presentation [54, 77] that implements Shneiderman's zooming paradigm [111]. Since HyperGrid is a very powerful interface, it would have introduced too much variance to the experiment. So, in order to test table-based result lists in general, a trimmed-down table-based result list was used. A screenshot of this variant is presented in Figure 5.1. The effect of this presentation is that all possible targets of a kind are listed below one another so that the user does not need to look around to see where, for example, the next author name might be found.

Traditional result list with highlighting

The other result list variant in the first experiment is a traditional result list with highlighting. Depending on the task, suitable parts of the result item surrogates are highlighted in order to help the user. Figure 5.2 shows a screenshot of a part of a

5.3. Experiment 1: linear and table-based result presentation

1. Echocardiographic Diagnosis and Prognosis of Fetal Left Ventricular Noncompaction.
A Arunamata, R Punni, B Cuned, S Bharati, NH Silverman
2011 (ACM)
2. An Investigation into Dynamic Customer Requirement using Computational Intelligence.
Yih Tng Chong, Chun-Hsien Chen
2009 (ACM)
3. Comparative pathogenesis and systems biology for biodefense virus vaccine development.
GC Bowick, AD Barrett
2010 (ACM)

Figure 5.2.: Screenshot of the list variant

result list presented in this design. The example in the screenshot shows a result list configuration that was used for tasks in which the searcher had to recognize information in the authors section. Therefore, author names in all surrogates are highlighted. This variant is based on the idea that the user has to make some effort in navigating between those parts in the surrogates that are relevant for the search. It was hoped that highlighting reduces this effort since it transforms the sequential search process of finding new target regions into a parallel one. The study by Smith [112], summarized in Section 5.2, provided evidence that color-coding helps with visual search.

5.3.2. Method and apparatus

Each experimental session was about 90 minutes long and guided by a script to ensure that each participant received the same instructions and followed the same steps. Each session was conducted by the same experimenter. Participants were greeted and instructed briefly on the goal of the experiment and the different steps involved. After signing a consent form, a d2-R concentration test [26] was conducted. The procedure for this test was strictly in accordance with the d2-R test manual. After the participant finished the test, the participant completed a short pre-experiment questionnaire that asked for demographic data to describe the sample (see Figure B.1 in Appendix B).

Following the questionnaire, each participant was further instructed regarding the intention of the experiment and about the software used in it. Informing the participant about the experiment's intention was considered not harmful for the measurements since it applied to all participants and also because the experiment design was a repeated measures design and not focussed on user behavior. As an incentive, all par-

5. Support for Recognition

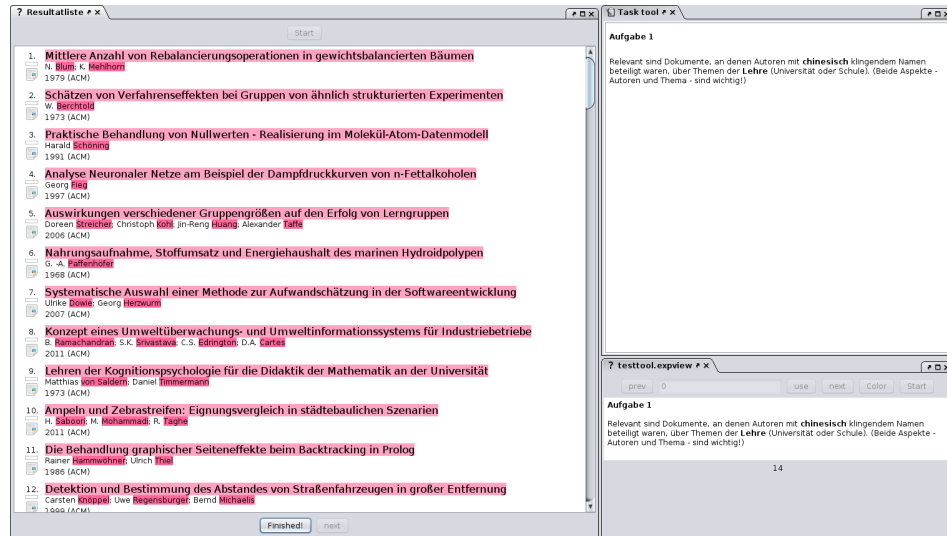


Figure 5.3.: Screenshot of the ezDL system used in the experiment

Participants were told that the best four participants were awarded another 10 euros in cash when all experiments were completed and that gathering of personal (address) details was not necessary in order to be able to collect the award.

The actual experiment was conducted using a modified version of the ezDL system [13, 14] that had only one area for showing the search tasks and another for the result lists. One window of the system was located on the monitor of the experimenter, invisible to the participant, and used to control the experiment. See Figure 5.3 for a screenshot of the user interface showing the list variant in the left pane and the task display in the upper right-hand corner. In this figure, the experimenter window is docked in at the bottom right.

Before the participants began working on the visual search tasks, they could choose their favorite highlighting color using a mock result list in the style of the highlighting variant.

The experiment consisted of a series of visual search tasks (see Section 5.3.3) that the participants were asked to complete. For each task a hand-crafted static result list (see Section 5.3.4) was presented to the participant along with a description of the documents that the participant was asked to mark by clicking on them.

5.3. Experiment 1: linear and table-based result presentation

Measurements taken were the documents each participant selected as relevant to the task and the time needed to complete a task; this time was allowed to be a maximum of 120 seconds. Participants were made aware of the time limit prior to the experiment. During each task, all participants were informed about the progress of the time after one minute (“one minute left!”) and 30 seconds before the end of the task (“30 seconds left!”). Participants were asked to end the tasks if they thought they were done with finding relevant items. They could indicate so by clicking on the “Finished” button at the end of the result list. After two minutes participants were asked to stop working on the task by clicking on the button. For this reason, the recorded completion times could exceed 120 seconds and were adjusted after the experiment to eliminate timing variance due to different response speeds of the participants.

Each participant completed 28 tasks including four training tasks.² The first 14 tasks were completed using the one result list variant, the last 14 tasks using the other. Half of the participants started using the list variant, while the other half started using the table variant. The participants were assigned randomly to one of the two experimental conditions.

At the end of each session, participants completed a short post-experiment questionnaire with four opinion items about the interface and the tasks (see Figure B.2 in Appendix B). The questionnaire was kept to a minimum since the primary data was collected using event logging and the questionnaire was mostly used as a device for closing the session.

Some sessions were recorded using a remote eyetracker. Because eyetracking data was not essential for the experiment, participants could decide if they wanted to have their eye movements recorded. All participants agreed, but in some cases technical problems prevented recording eye-tracking data.

5.3.3. Tasks

The intention of the experiment was to find out which of the two result list variants is better in supporting users with visual search tasks. These tasks occur sometimes when users cannot specify certain important document details (such as the book cover) or

²The training task was meant for the participant to get to know the task type and result list variant; no measurements were used in these tasks.

5. Support for Recognition

when they have difficulties to recall exact details of the documents they are looking for.

In the experiment, the participant would ideally be able to recognize a certain item but not able to retrieve details from memory about it at will. It is very hard to reliably manipulate participants in a lab setting so that they exhibit this state of memory since probing for the memorized datum might result in their learning it again or refreshing the memory of it.

The strategy to work around this problem was to assign two different kinds of tasks to the participants and deprive them of usual tools for working with result lists (e. g. filtering the result list) so that the participants could only find the searched-for documents by looking at them and recognizing them.

The two different kinds of tasks used were open and closed tasks. The closed tasks quoted certain names or keywords the participant could find exactly as quoted in the result list (e. g. “Find documents by Felix Riepe.”). They were used to simulate the user being able to recall exact details but not able to use them for specification: since the user interface did not offer a way to search or filter, the only way the participant could find the targets was by visual search.

The open tasks used vague descriptions of the documents to find (e. g. “Find documents about car manufacturing.”) in an attempt to increase the cognitive load of the participants. They were used to simulate the memory recall problem, but it was not clear before the experiment if this kind of task would result in good enough variance.

Part of the tasks with vaguely defined information needs were tasks about person names of different cultural backgrounds. To introduce a certain amount of variety into the experiment and to work around the probability that participants have systematic trouble recognizing a certain cultural background, two different sets of vaguely-defined names were used in the experiment: Chinese and Japanese names.³

To guard the experiment from the problem that different persons have different ideas of what a Chinese or Japanese name looks and sounds like, a pre-experiment was

³Using different name sets in different experimental conditions could have introduced a confounding variable. However, an in-experiment test about the names showed that participants did not have any trouble recognizing either Japanese or Chinese names in a list with those names and distractors.

5.3. Experiment 1: linear and table-based result presentation

conducted. In the pre-experiment, seven participants were asked to assign the labels “Chinese” and “Japanese” to names from various cultural backgrounds after having been shown lists with example names to learn the concepts. The labels were assigned correctly in nearly all cases by almost all participants. Those names that caused participants to mislabel were changed to make them easier to label.

All task descriptions were given in German because the participants were recruited at a German university. The participants were required to be able to read German in order to avoid confounding the experiment by different language proficiencies. For all open tasks the result lists contained only documents with German titles, whereas the result lists in the closed tasks also contained documents with English titles because these were easier to obtain and the participants were expected to be able to at least recognize English terms and names. Moreover, this kind of task is not unusual in real search situations.

Tasks were grouped into two groups with seven tasks each. The first task in a group was a training task, followed by two tasks dealing with only author names, then two dealing with document titles and finally two tasks that involved both author names and document titles (e.g. the German equivalent of “Find documents authored by Felix Riepe and containing ‘car manufacturing’ in the title.”).

For each result list variant, participants completed one task group with open tasks and one task group with closed tasks. A permutation scheme made sure that half of the participants started each interface with open tasks and the other half started with closed tasks.

To prevent a learning effect from re-using the tasks for the second result list variant, two sets of tasks were provided that were very similar in structure but had different names and keywords and used completely different result lists.

The permutation scheme resulted in eight different orders in which the experiment was conducted. Since the experiment design was based on a repeated-measures setup, the number of participants had to be a multiple of eight to measure full rotations of the permutation scheme.

5. Support for Recognition

5.3.4. Lists

There are two different corner cases regarding target prevalence in the lists: low and high prevalence. At low prevalence, the error rate is expected to be high so a good support mechanism could provide benefits. But assuming that the improvement is only small, the number of found targets in lists with only one target might not increase at all. At high prevalence, the error rate is usually low so a support mechanism might not provide any benefits. Also, the training effect of high-prevalence lists might influence search success in low-prevalence lists.

For these reasons the number of search targets per list was in the range between 1 and 3 and unknown to the participant to a) simulate a usual search situation and b) be in the range where visual search performance is usually low.

For each task type (author names, titles, combination of the two) the participants processed two lists; the sum of targets of the lists per type was always four (i. e. combinations of 1+3, 2+2 and 3+1).

The lists were built using pseudo-random search terms querying five⁴ different digital libraries using ezDL [14]. For title term tasks, lists were generated searching digital libraries for a number of German keywords. The resulting lists were merged and shuffled into a seed list. Using the seed list, a small Java program generated the experimental lists, trimmed them to 100 items and inserted the targets and distractors as defined previously in task description files. The title length was chosen to be in the range between 70 and 90 characters and short enough to be rendered without needing horizontal scrolling.

Great care was used to ensure that the search targets were in the list only as intended in terms of quantity and location. Incomplete items were removed or completed and bogus target items inadvertently imported from the seed list were changed so they were no longer search targets.

The position of the search targets was determined by using a random number generator since no other plausible model for the distribution of the targets could be found. The numbers were generated until a minimum distance of 5 was found to combat the effects of attentional blink [100].

⁴ACM, IEEE, PubMed, DBLP and Springer

5.3.5. Sample

24 participants were recruited at the University of Duisburg-Essen between July and September 2012 using flyers, posters and posts in three different Internet forums. The only requirement for participating was being proficient in reading German texts.

As compensation, each participant was offered either 15 euros in cash right after the experiment or a certificate of taking part in the experiment, which is required for some students. All of the participants chose the 15 euros for compensation.

The participants' ages were in the range between 21 and 31 years, with outliers at 19 and 32 years (mean = 25).⁵

20 participants were enrolled in a university, either as undergraduate or graduate, three had a university-level education and were working in the industry, and one participant had professional training.

Participants had on average 10.2 years of experience with search systems (e.g. web search). On a scale between 1 ("beginner") and 5 ("expert") the median self-reported experience level was 4. 18 were male, 6 were female.

5.3.6. Results

In the evaluation of binary classification tasks, classification choices are often described as "true positives," etc. In this section, a true positive denotes a document that a user correctly identified as relevant. Accordingly, a false negative is a document incorrectly identified as irrelevant and a false positive is a document incorrectly identified as relevant.

In total, none of the documents in the closed tasks were mistakenly marked as relevant by the participants, while 335 documents were false positives in the open tasks. In relation to the 28,800 possible mismarkings, this means about 1% of the documents were wrongly marked as relevant in the open tasks. While the percentage is not that large, the difference between the task types is statistically significant with $p < 0.0001$.

⁵"Outlier" in this respect are data points at least 1.5 times the interquartile range below the 25th percentile or above the 75th percentile.

5. Support for Recognition

A χ^2 test reported that the data of all three task sub-sets (open, closed, all) for the main metrics was probably not normally distributed. So a non-parametric test was chosen as a significance test of the success metrics, namely a randomized version of Wilcoxon's signed rank test⁶.

Success measured by using both open and closed tasks

As the main measure of success, the number of true positives per time normalized to 120 seconds (tppt) over all task groups for each interface was used.

The mean *tppt* for the table variant was 1.78 (*sd* = 0.58), for the list variant it was 1.75 (*sd* = 0.46)—see Figure 5.4. This and the following graphs show the means and 95% confidence intervals of the respective data.

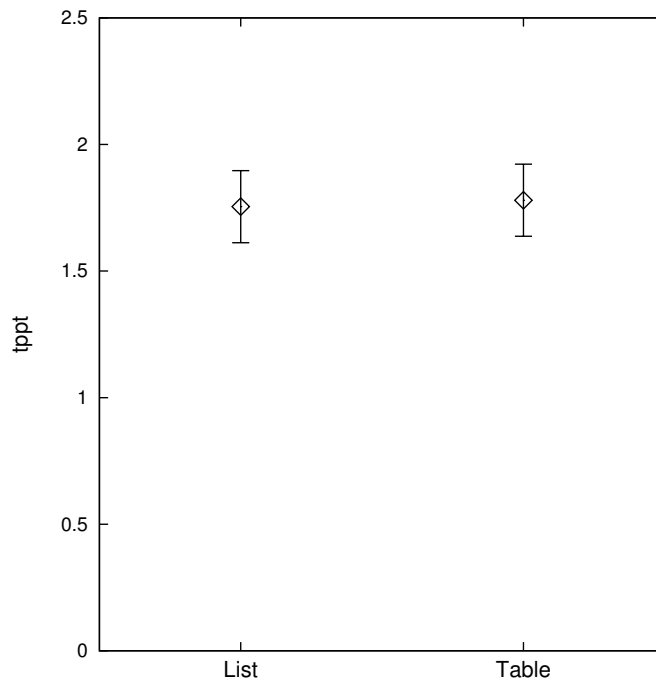


Figure 5.4.: True positives per 120s in both open and closed tasks

⁶Using the R package `coin`

5.3. Experiment 1: linear and table-based result presentation

The randomized Wilcoxon test gave a p-value of 0.66, indicating that the difference measured between the two variants is likely due to chance.

In the following sections, other variables are examined exploratorily.

Success measured by using only the closed tasks

As a secondary measure of success, *tppt* over the closed task groups was used for each interface. The closed task mean *tppt* for the table variant was 1.92 ($sd = 0.63$); for the list variant it was 2.09 ($sd = 0.66$). See Figure 5.5 for graphs of the data.

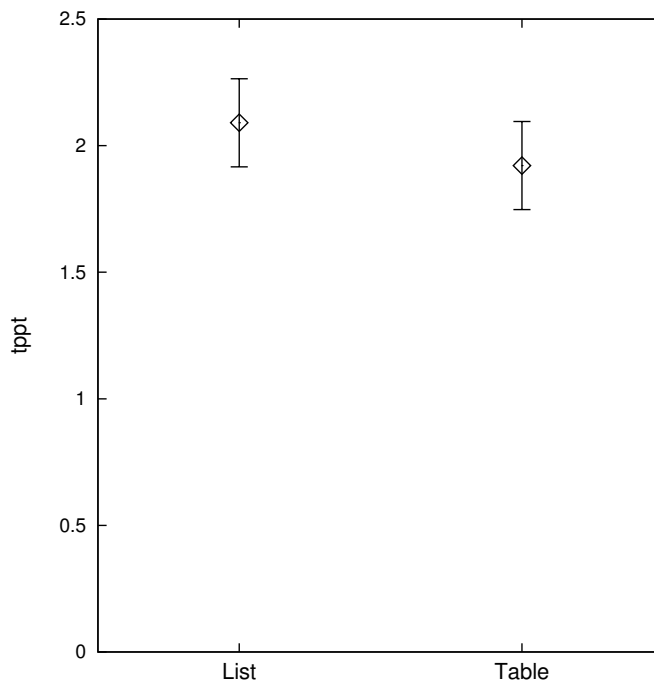


Figure 5.5.: True positives per 120s in closed tasks

The randomized Wilcoxon test gave a p-value of 0.12, indicating that the difference measured between the two variants is likely due to chance.

5. Support for Recognition

Success measured by using only the open tasks

As another secondary measure of success, *tppt* over the open task groups was used for each interface. The open task mean *tppt* for the table variant was 1.66 ($sd = 0.61$); for the list variant, it was 1.49 ($sd = 0.40$). See Figure 5.6 for graphs of the data.

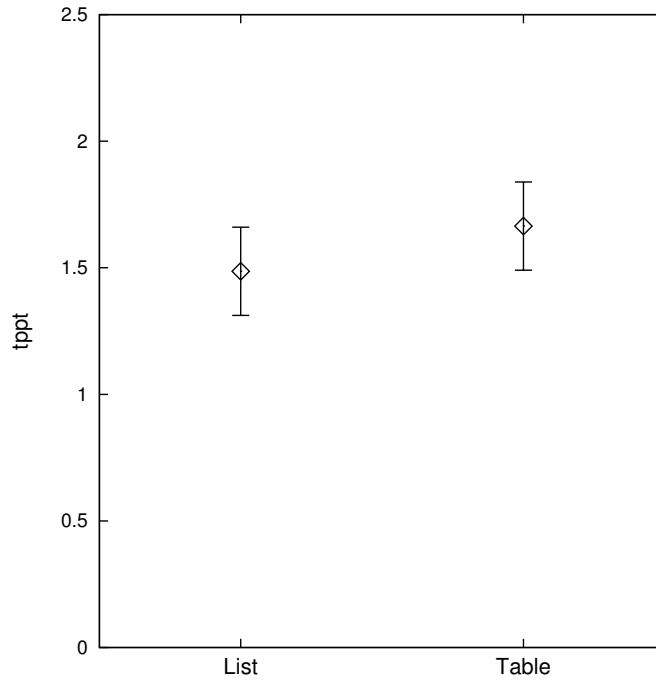


Figure 5.6.: True positives per 120s in open tasks

The randomized Wilcoxon test gave a p-value of 0.18, indicating that the difference measured between the two variants is likely due to chance.

Fatigue

Ackermann and Kanfer [1] reported that their participants suffered from fatigue after longer testing times. Due to the design of the experiment, fatigue effects were not expected. To examine whether the participants' fatigue still influenced the results, the

5.3. Experiment 1: linear and table-based result presentation

success metrics of the tasks grouped by processing order were compared. Wilcoxon's signed rank test reported a p-value greater than 0.6 for all task subsets, indicating that there is no significant difference in visual search success between the first and the last task sets, so a fatigue effect is unlikely.

General observations

One participant noticeably changed his working strategy during the experiment. He began processing the result lists top-down and switched to bottom-up mid-experiment.

Each participant was told that the result list scrolling was implemented so that one notch of mouse wheel rotation translated to the next result item being displayed at the upper border of the result list (i. e. the scrolling did not result in half items being at the upper border or items being skipped). Still, most participants' eye movement pattern was to first visually go down the list and then, upon arriving at the lowest visible item, further scrolling down the list.

Relationship between being able to focus and success

To find out if there is a connection between the ability to concentrate and general success in the tasks, a Spearman rank correlation test was chosen due to its better robustness against outliers in comparison with Pearson's test. The general success was measured as the time, in seconds, spent per true positive in all tasks. Figure 5.7a shows an overview of the data. Just by looking at the scatter plot, a correlation seems unlikely.

A hypothesis with good face validity is that the better a person is able to concentrate, the more successful they are in completing the visual search tasks. This would lead to a positive correlation coefficient ρ . In fact, the Spearman test for the alternative hypothesis of ρ being greater than 0 gave a p-value of 0.11 with a ρ of 0.27, which means only little correlation, backed by little evidence.

The same test performed with data from only the closed tasks (see Figure 5.7b) yields a p-value of 0.012 with $\rho = 0.46$, being significant at the 0.05 level.

5. Support for Recognition

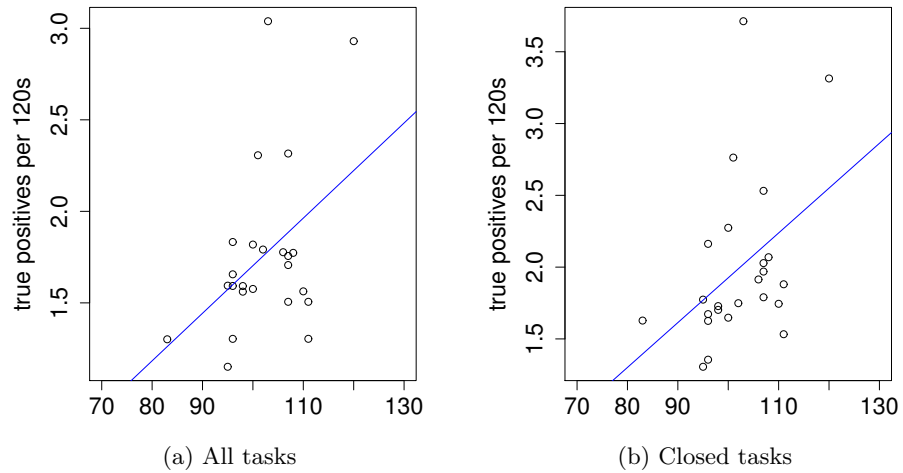


Figure 5.7.: Correlation between ability to concentrate and true positives found per time

Suitability of each variation for subjects with either high or low concentration score

To find out if there is a connection between the participants' concentration score, KLSTD, and the design variant they were most successful with, the data for the closed tasks was divided into participants with low and high concentration. The border between low and high concentration scores was determined using two different methods: The first one was dividing at the median KLSTD measure; the second one was clustering using K-means.

Splitting at median KLSTD

Using the first method, median, the data was split at 101.5. For each group (below median KLSTD and above median KLSTD) Wilcoxon's signed rank test was performed between the success measures for the list variant on the one hand and table variant on the other hand. In this section, success was measured again as true positives per 120 seconds.

5.3. Experiment 1: linear and table-based result presentation

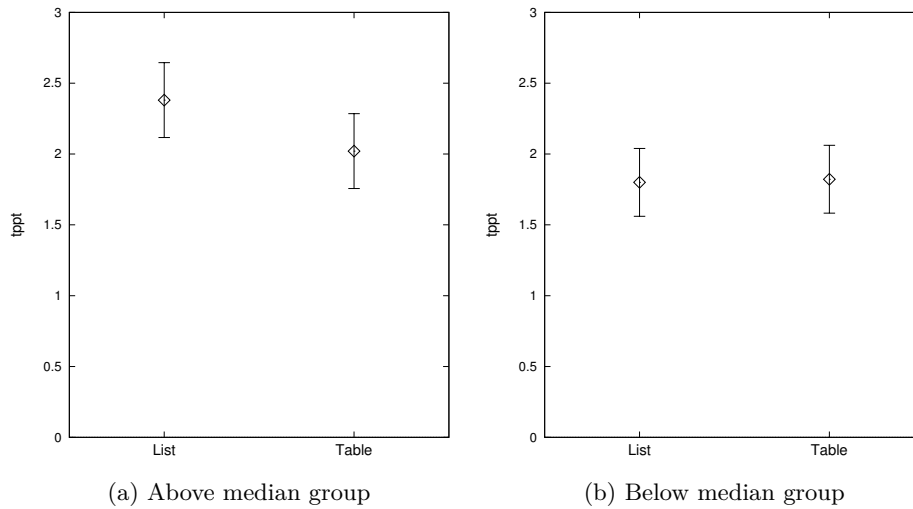


Figure 5.8.: Success metrics for different concentration scores (KLSTD) for list and table – grouping by median

In the group of participants whose concentration was above median, the mean *tppt* was 2.38 for the list variant and 2.02 for the table variant (see Figure 5.8a).

In the group of participants whose concentration was below median, the mean success was a *tppt* of 1.80 for the list variant and 1.82 for the table variant (see Figure 5.8b).

While there was a trend towards the list variant being better in both groups, none of the differences was statistically significant. Wilcoxon's test gave a p-value of 0.68 for the below median group and 0.09 for the above median group, the latter hinting at a possible systematic effect.

Splitting using K-means

The data was split by K-means resulting in centroids at 96.6 and 108.8. This resulted in a split at a KLSTD of 102.5, slightly higher than when splitting by median KLSTD. The graphs are very similar to those from the previous section and are, therefore, omitted for the sake of brevity.

5. Support for Recognition

Wilcoxon's signed rank test was performed again on the data, giving p-values of 0.17 for the high KLSTD cluster and 0.50 for the low KLSTD participants, both of which being far from even a relaxed significance threshold.

Further influence on task success

Gender and success

The success measure "true positives per 120s" differed by 0.07 between the two genders that participants stated: Female participants found 1.81 true positives on average while male participants found 1.74. A Wilcoxon's rank sum test comparing the success of male and female participants gave a p-value of 0.13 for the difference.

Self-reported search experience and success

Participants were asked in the pre-experiment questionnaire to rate their search experience on a partial differential scale between the extremes of 1 ("beginner") and 5 ("expert"). The answers were in the range between 2 and 5, with 9 participants having chosen 3 and 12 having chosen 4.

A Spearman correlation test on the relationship between self-reported search experience and search success gave a p-value of 0.46 for a very mild $\rho = -0.02$, both too far away from the range of values that indicate a meaningful result.

The data gathered

The question remained if it was useful to include open tasks or if they introduced noise. When looking at the false positives it was obvious that participants' notion of the search targets was much less clear in the open tasks than in the closed tasks.⁷ To find out if this was a systematic effect, the difference in false negatives between open and closed tasks was examined using Wilcoxon's signed rank test with the alternative

⁷Adding both table and list conditions, participants classified not a single irrelevant item as relevant in the closed tasks, but 335 items in the open tasks.

5.3. Experiment 1: linear and table-based result presentation

hypothesis that the open tasks have more false negatives than the closed tasks. The mean number of false negatives in the closed tasks was 6.58 ($sd = 2.92$), while in the open tasks it was 8.17 ($sd = 2.79$).

The test reported a p-value of 0.005, indicating that the higher number of false negatives in the open tasks had a systematic reason and was not due to chance. Figure 5.9 shows a graph of the data.

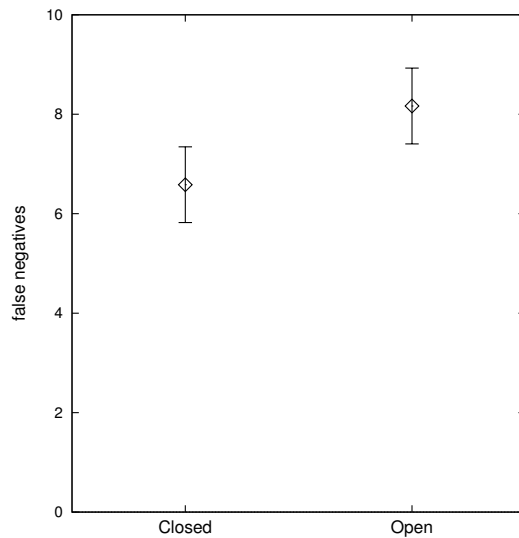


Figure 5.9.: Difference between false negatives in closed and open tasks

5.3.7. Discussion

Despite the attempt to control the experiment tightly, the results are inconclusive.

The analysis of the effect of the kind of task (see Section 5.3.6) shows that it was more difficult for the participants to perform the open tasks than to perform the closed tasks. While this was intended by the experimenter, it might have contributed to the overall inconclusive results. Consequentially, the follow-up study dropped the open tasks and used the freed resources for adding a baseline result list to the experiment.

5. Support for Recognition

1. **Echocardiographic Diagnosis and Prognosis of Fetal Left Ventricular Noncompaction.**
A Arunamata; R Punn; B Cuneo; S Bharati; NH Silverman
2011 (ACM)
2. **An Investigation into Dynamic Customer Requirement using Computational Intelligence.**
Yih Ting Chong; Chun-Hsien Chen
2009 (ACM)
3. **Comparative pathogenesis and systems biology for biodefense virus vaccine development.**
GC Bowick; AD Barrett
2010 (ACM)

Figure 5.10.: Screenshot of the baseline variant

5.4. Experiment 2: baseline, linear and table-based result presentation

Since the previous experiment described in Section 5.3 did not produce significant results, the experiment was repeated in an altered version. The main measure was again the number of relevant documents found per time.

The prime changes were that this time three design variants were tested (the two from the previous experiment plus a baseline) and that only closed tasks were used.

5.4.1. Method and apparatus

As in the previous experiment, detailed in Section 5.3, the experiment was conducted using a modified ezDL system. Since the d2-R test was time-consuming and the collected data gave non-significant results, the test was not used in this rendition of the experiment. Instead, the experiment compared three different result list variants: a baseline variant using the same design as in ezDL 1.6 (shown in Figure 5.10), the result list with highlighting (see Figure 5.11) and the table-based result list (see Figure 5.1 in Section 5.3). The list with highlighting was changed slightly in comparison with the one used in the previous experiment to differ less from the baseline variant.

The experimental session started with a consent form and a short questionnaire to collect demographic data (see Figure C.1 in Appendix C). The experimental tasks were then performed. The session closed with a post-session questionnaire that collected opinions about the experiment and was basically only used as an outroduction.

5.4. Experiment 2: baseline, linear and table-based result presentation

1. **Mittlere Anzahl von Rebalancierungsoperationen in gewichtsbalancierten Bäumen**
N. Blum; K. Mehlhorn
1979 (ACM)
2. **Schätzen von Verfahrenseffekten bei Gruppen von ähnlich strukturierten Experimenten**
W. Berchtold
1973 (ACM)
3. **Praktische Behandlung von Nullwerten - Realisierung im Molekül-Atom-Datenmodell**
Harald Schoningh
1991 (ACM)

Figure 5.11.: Screenshot of the list variant

The participants performed tasks with all three list variants. Each variant was tested using seven closed tasks: one training task to allow the participant to get to know the variant being tested, two author-based tasks, two title-based tasks, and two tasks that were about combinations of authors and titles. The tasks were designed just like in the previous experiment; the only difference were the actual search targets and the fact that the open tasks were dropped to reduce noise in the data.

Participants were allowed two minutes to complete each task and got audio signals (short beeps) in a low volume from front speakers at 60, 90 and 120 seconds to allow for some time orientation and also to give a slight sense of urgency. As an incentive, the participants were told that the five best performing participants were awarded a bonus of an additional 10 euros.

After each design variant, the participants were given a break of two minutes.

At the end of each experimental session, the participants completed a short questionnaire that was very similar to the one used in the previous experiment (see Figure C.2 in Appendix C).

To counter learning and fatigue effects, the order of the design variants was rotated using a Latin square design. Also rotated were the sets of tasks to make sure that all design variants were tested against all tasks. This rotation design resulted in a 3×3 rotation, requiring the number of participants to be an integer multiple of 9.

5.4.2. Sample

In total, 31 participants were recruited using web forums and public bulletin boards of the University of Duisburg-Essen. During the experiment, some participants' data

5. Support for Recognition

was discarded for different reasons.⁸ The sessions of these participants were repeated with new participants recruited using the same method. The data of 27 participants was used for the trial. Of these, 11 were male and 16 were female.

The mean age of the participants was 24.4 years (the youngest participant was 20 and the oldest one 42 years old). The participants reported on average to have used search engines and similar offerings for 10.4 years. Asked to rate their perceived experience with search systems on a scale between 1 (beginner) to 5 (expert), the median answer was 3. Only one participant reported to be working full-time; 26 participants stated they were students. Most students had a computer-science background of various degrees: Of those who stated their course of study, three were computer-science (CS) students, 13 studied a course of study combining psychology and CS.

5.4.3. Results

The number of false positives was 0 over all participants and tasks—not a single participant mistakenly marked even a single document wrongly as relevant. This shows that using only closed tasks indeed reduces the noise.

Main metric

The main success metric was the number of true positives per 120 seconds task time (higher values are better) over all task types.

The mean of the metric for the baseline variant was 1.78 true positives per 120 seconds, for the highlighting variant it was 1.94 and for the table variant it was 1.91. Figure 5.12 shows a graph of the data.

A randomized ANOVA test, testing the success metric against the design variant, yielded a p-value of 0.32 for the difference between all design variants ($F(2, 78) = 1.14$), indicating that the difference is likely due to chance.

⁸One participant served as a test user; two sessions were discarded due to an error in a task description; one participant's data was removed because they admitted to having worked on the wrong task.

5.4. Experiment 2: baseline, linear and table-based result presentation

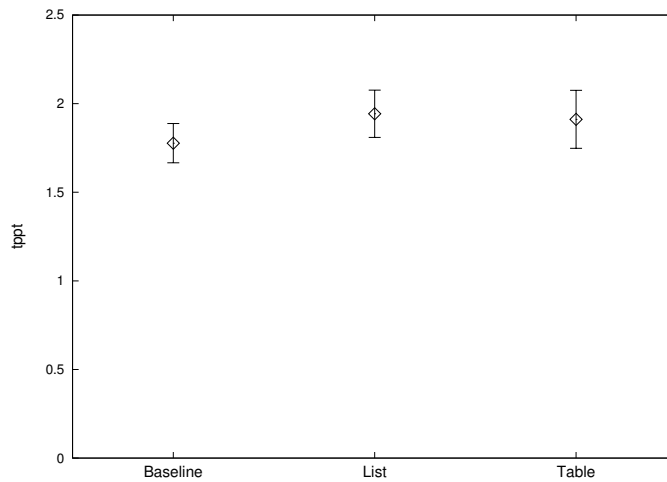


Figure 5.12.: True positives per 120s

To examine if fatigue or learning play a part in the experiments, the main success metric was compared not between the design variants but between the first, second and third set of tasks each participant worked on. Regarding these differences, an ANOVA test measuring the effects of the variable “position” gave a p-value of 0.48 ($F(2, 78) = 0.75$), indicating that there is no significant difference in task metrics over time.

Only author-based tasks

Examining only the author-based tasks, an ANOVA about the effects of the interface variant yields a p-value of 0.01 ($F(2, 78) = 4.95$), meaning a very significant effect of the variant on the success measure. Here, participants found 2.8 targets per 120 seconds using the table, while only finding 2.6 using the list and 2.2 using the baseline variant.

Only title-based tasks

For just the title tasks, the ANOVA reported that neither the effect of the design variant ($F(2, 78) = 0.55$) nor the time effect ($F(2, 78) = 0.16$) is statistically significant (p-

5. Support for Recognition

values 0.58 and 0.86, respectively).

Only combined tasks

Examining only those tasks that dealt with both authors and titles, an ANOVA test gave a p-value of 0.33 for the effect of the design variant ($F(2, 78) = 1.11$).

The time effect for this task type is not significant, either, with $p = 0.07$ determined by an ANOVA test ($F(2, 78) = 2.83$).

5.4.4. Discussion

None of the design variants gave significant differences for either any single task type or all task types in total. This means that the experiment did not gather enough evidence to support the hypothesis that supporting visual search actions using one of the suggested result list variants is more successful than another. For the time being, it has to be assumed that all three variants perform equally well.

There is still a hint at a possible effect when only considering either author-related tasks or combined tasks: the table variant seems to outperform the other variants by 8.8% compared to the highlighting variant, and 29.1% compared to the baseline. Why there is such a difference is not clear, though. This might be because the highlighting really did help with reducing the time needed for finding author names in the surrogates. The difference might also be caused by a tactic that a few participants reported after the experiment: they used a two-tier approach for finding the author name targets by first looking for highlighted boxes of similar length as the target name and only read those names whose boxes fit the target size. How many participants really used this tactic is unclear. An experiment aimed toward examining this question might try to standardize the sizes of the highlighted boxes for author names, possibly after removing longer names.

That both the combined tasks and the author tasks show the same pattern (and a relatively low p-value) is derived from the fact that most participants used a particular tactic in the combined tasks. The tactic was first examining the authors and, if the authors matched, examining the title. Thus, in most cases the time to check a

document in the combined task was about the same as in the author task. This tactic was used by some participants in the previous experiment and the participants in this experiment were encouraged to proceed like this to eliminate some of the variance in success based not on differences in the design variants but in the tactics used by the participants: it could have happened that a participant found out about this tactic only late into the experiment, resulting in an improvement in a particular design variant.

The trouble with the p-values is that the lowest p-values were found in an exploratory analysis looking at multiple metrics. This inevitably causes alpha inflation so the p-values are not only not very low, they also have to be treated with caution since the real probabilities might be even higher.

5.5. Collective analysis

Each of the studies presented so far was unable to show a difference between the experimental conditions. For a collective analysis, the data from both experiments was consolidated into a single data set. The resulting data set included 51 participants and only the list and table variant data restricted to the closed tasks. The effect of the kind of result list on the success in visual search tasks was evaluated as in the first experiment. The randomized Wilcoxon's signed rank test gave a p-value of 0.19 for the main success metric, true positives per time, not allowing to reject the null-hypothesis. The p-value for time-on-task was 0.23. If only the number of true positives is used as the success measure, the two variants differ with 8.3 for the table and 9.1 for the list, being statistically significant with $p = 0.02$. Please note that these tests were not adjusted for alpha inflation, so the few significant results are quite possibly type-I errors.

5.6. Conclusion and outlook

In the first experiment, the list variant seemed to outperform the table variant by some degree when only closed tasks or the combined tasks are considered, even though the difference is not large enough to be statistically significant. In the second experiment, the list variant with highlighting seemed to also outperform both the table variant and

5. Support for Recognition

the baseline. Again, the difference is not large enough to yield statistical significance. In the meta-analysis involving data from both first and second experiment, only one measure showed a statistically significant difference, which was in favor of the list variant.

The data at hand from two experiments with a total of 51 participants points towards a slight advantage of the list with highlighting over the other two design variants.

Since the difference between the tested variants was not as big as desired, further variants should be examined. These variants do not have to look like traditional result lists: Some participants were observed to quickly scroll back and forth in the results. The eyetracking data for some participants showed that they sometimes only glanced over certain surrogates. These types of behavior might make the searcher miss targets. A possible way to overcome the superficial skimming behavior is to show the result documents not in a list, but one-by-one such as in a slide show with forced gaze times for each surrogate. This would be an adaption of the idea presented by Forlines and Balakrishnan [51] and could lead to a better rate of true positives but also to reduced user satisfaction, since users are rarely fond of being patronized. In some situations, the trade-off between these two aspects might still be worth considering.

Another idea to support recognition is to shift the focus from the actual act of recognition, which we have seen is not easy to support, to the filtering query that usually precedes the recognition step. If the power of the filtering query language was increased, the number of items to examine during tasks involving recognition would be reduced, possibly leading to reduced task times.

In a follow-up study, an evaluation could be made on whether it is a better approach to mark the search targets by special characters (e. g. an “x”) instead of specific cultural backgrounds, or other more or less ambiguous descriptions: There is less room for interpretation in the question if “Michaxl” contains an “x” or not than in the question if “Chang” is a Chinese name or not. The participants’ trick of looking for highlights of a specific length could be side-stepped by standardizing the length of the author names.

While little is still known about support mechanisms for recognition, many ideas have been studied for supporting specification activities—and some of these are known to improve the efficiency of searchers.

6. Support for Specification

In IR systems based on the query result list paradigm, specification is the act of typing the search query into a text input affordance of some kind (see Section 2.3.2). It is a frequently used mode of communicating the information need to the system but, despite its popularity, it is not without problems. An important problem with query formulation is the vocabulary problem, as described by Furnas et al. [52]. The problem refers to the amount of overlap between the vocabulary used by authors or indexers to describe facts or texts and that used by searchers to describe the same things. This overlap between vocabularies of different people is so small that single descriptors match in only 10–20% of the searches. Inexperienced searchers may not be aware of this problem; and even experienced searchers may need help if they are new to a certain knowledge domain they are interested in.

While some problems derive from the inherent vagueness of language, other problems arise by users simply making mistakes. The following sections outline a few of these problems and approaches to solve them. First, methods for supporting the typing of plain search queries are introduced, such as query forms. The sections that follow detail support features like spelling corrections and translations that can be used to alleviate problems with queries or to improve queries. The concluding two sections discuss how to present support suggestions to the user. This discussion is related to the levels of system support described by Bates [9] and covers the whole range from having the user explicitly invoke these functions through to the system automatically altering the query without the user even being aware of it.

6.1. Query input

In full text search systems, users specify the desired documents by entering a query into a form, which can be a simple one-textbox form, as seen in many web search

6. Support for Specification

systems, or a more complex one with multiple text fields and other input widgets like drop-down lists. The latter kind is often used in systems that search in structured data, such as OPACs or customer data bases. Stempfhuber collected a number of different query forms that also include experimental interfaces [114, p. 61].

When only considering query text boxes, this leads to two design decisions that have to be made: one about the design of the query form, such as how many and which query fields to use, and one about the grammar of the queries entered into the search boxes.

6.1.1. Query languages

Depending on their application domain, query languages can be very simple. The design space begins with treating the whole input as a single phrase to be searched for. However, if search systems are designed to support complex queries, this complexity has to be represented somewhere in the system. With simple query languages, the complexity is usually placed in the query form (see Figure 6.2), which gets some input affordance for each aspect the user is supposed to specify. Some query form designs shift the complexity from the query form to the query language (see Figure 6.1). These forms are much simpler, while the language understood in the query text fields is complex and can have many operators and syntax elements. These designs can also be mixed: the ezDL desktop client has a query form with multiple input fields. Each field refers to a specific meta-data field but the input can use the full ezDL query syntax. The following sections discuss query languages of differing complexity.

Simple languages

Simple query languages—such as one that accepts only a list of terms—have limited expressive power. They are acceptable if the underlying search engine or IR model is very simple in terms of their queries, as well. In the coordinate level match model, for example, neither document terms nor query terms are weighted. Queries are simple sets of terms, so there is no need for a more elaborate query language than just listing those query terms. In other models, such as the vector space model, terms can be

The screenshot shows the 'Command Search' tab of the IEEE search interface. At the top, there are four tabs: 'Advanced Keyword/Phrases', 'Command Search' (active), 'Citation Search', and 'Preferences'. Below the tabs, the main heading is 'ENTER KEYWORDS, PHRASES, OR A BOOLEAN EXPRESSION'. A note states: 'Note: Use the drop down lists to generate the correct Operator and Data Field Codes. This wizard will NOT build your expression. View examples of how to write a boolean search string'. Below this, there are two radio buttons for 'Search': 'Metadata Only' (selected) and 'Full Text & Metadata'. There are two dropdown menus for 'Data Fields' and 'Operators'. A large text area contains the query: `((\"Abstract\":java) OR \"Publication Title\": \"computer technology\") AND \"Document Title\":rfid`. To the right of the text area, there are 'SEARCH GUIDELINES' which state: 'Operators need to be in all caps - i.e. AND/OR /NOT/NEAR.', 'Asterisk wildcards cannot be used within quotes or with the NEAR/ONEAR operators.', and 'There is a maximum of 15 search terms.'. At the bottom, there are 'Reset All' and 'SEARCH' buttons.

Figure 6.1.: Simple query form, complex language: ieeexplore.ieee.org Command Search

weighted, so the query language has to provide some means to express the weight of query terms. This problem might be solved by accepting queries that contain terms multiple times and by using the multiplicity of the terms as their weight. However, this approach is of limited usability because the queries would get very long and require a lot of typing to formulate them. Thus, a better and more expressive query language would incorporate a means to express term weights. Similar arguments can be made for other features of the search subsystem, like fields, stemming, filter queries, phrases and proximity search.

Complex languages

Systems that include many features usable in a search and which want to expose them in the textual query require more complex query languages.

Carmel et al. [30] described a complex query language for queries on historical information that includes Boolean operators, wildcards and field descriptors.

6. Support for Specification

Advanced Keyword/Phrases | Command Search | Citation Search | Preferences

ENTER KEYWORDS OR PHRASES, SELECT FIELDS, AND SELECT OPERATORS
Note: Refresh page to reflect updated preferences.

Search : Metadata Only Full Text & Metadata

in Metadata Only

AND in Metadata Only

AND in Metadata Only

CONTENT FILTER

All Results
 Open Access

PUBLISHER

Return Results from

<input type="checkbox"/> IEEE(3,300,683)	<input type="checkbox"/> Alcatel-Lucent(6,310)
<input type="checkbox"/> AIP(286,776)	<input type="checkbox"/> IBM(6,236)
<input type="checkbox"/> IET(215,340)	<input type="checkbox"/> BIAI(2,682)
<input type="checkbox"/> AVS(36,860)	<input type="checkbox"/> TUP(2,307)
<input type="checkbox"/> MITP(21,580)	<input type="checkbox"/> Morgan & Claypool(600)
<input type="checkbox"/> VDE(6,497)	

CONTENT TYPES

<input type="checkbox"/> Conference Publications (2,503,223)	<input type="checkbox"/> Early Access Articles (9,924)
<input type="checkbox"/> Journals & Magazines (1,340,199)	<input type="checkbox"/> Standards (5,958)
<input type="checkbox"/> Books & eBooks (26,321)	<input type="checkbox"/> Education & Learning (374)

PUBLICATION YEAR

Search latest content update (12/22/2014)
 Specify Year Range From: All To: Present
 All Available Years

Figure 6.2.: Complex query form, simple language: ieeexplore.ieee.org Advanced Keyword Search

Veale and Hao [131] studied a mood lexicon and query language for specifying terms with different sentiment—e.g. **+crazy** and **-crazy**, which searches for texts with positive mentions of “crazy” and, respectively, negative ones.

ezDL

Another example of a complex query language is the ezDL system. ezDL translates user-provided queries to those of the connected IR systems. For this reason, the ezDL query language incorporates many features found in popular IR systems and digital libraries.

The simplest queries in the ezDL language are just terms separated by space characters. For instance, **information retrieval vagueness** is a grammatically correct query: each term is taken as-is and the space characters are interpreted as implicit AND operators. The same query could be rewritten as **information AND retrieval AND vagueness**. Phrases can be expressed using double quotes, so the former query might be more precise if rewritten as **"information retrieval" vagueness**. Many digital libraries and IR systems implement the concept of fields to support semi-structured documents. To search for “vagueness” in the title of a document, an ezDL user would use the query **Title=vagueness**. Furthermore, the proximity operator **NEAR** can be used to make the query more robust against variations in the word order: **information NEAR/2 retrieval**. To support synonym lists, the **OR** operator binds stronger than the **AND** operator. The operator priority can be changed using parentheses: **a AND b OR c OR d** is the same as **a AND (b OR c OR d)**. Sometimes, concepts with many synonyms are required to occur in specific fields such as the title. The query **Title=a OR Title=b OR Title=c** can be expressed shorter: **Title={a OR b OR c}**. The language also supports masking and truncation using the **\$** and **#** wildcards that mask exactly one and none to many characters, respectively.

Scanning vs. searching

The features in query languages can support different methods of information seeking: Some of these advanced features support the specification of searching queries, while others support specification of scanning queries.

6. Support for Specification

To give an example, Boolean operators used for filtering can be used to support searching queries, queries that intend to locate very few well-known documents. Without filtering, even documents that do not match all terms might be returned by the IR system if their relevance value is high enough. But if the user knows that a given term does appear in the title of the searched-for document, there is no point in listing documents that do not include that term.

The other method is scanning, locating vaguely defined documents of unknown number. In this method, syntax elements that translate this vagueness into a query are useful. Proximity search is useful for handling nominal phrases. For example, when the subject is “information search”, target documents might also use the formulation “search for information”, while irrelevant documents might contain the word “search” at the very beginning (e.g. the search for a solution) and the word “information” at the very end. So a mere AND operator would return too many irrelevant documents. A feature that allows to search for the terms “information” and “search” in a common context¹ would help the user to narrow down the search but keep it wide enough for high recall.

6.1.2. Query forms

A query has to be entered into a text input box. If there is a single text input box for the whole query and the documents are indexed into fields, the user might need a way to express which field certain query terms are about. For example, searching for works of a certain author might produce more fall-out if the author name is also searched for in the document full text, where the author might merely be mentioned as a source of inspiration or a quotation.

One way to achieve this is adding this feature to the query syntax. In ezDL, searching for a term “foo” anywhere in the documents would be expressed by the query `foo`. If `foo` is searched for in the title of the documents, the query would be `Title=foo`. The advantage of this approach is that the user interface may have less clutter—only one text input box is needed—and that very complex queries can be formulated². The disadvantage is that the user has to know this query syntax feature and that it involves

¹In several search grammars, this can be expressed by, for example, `information NEAR/2 search`, meaning “information” and “search” in a maximum distance of two words.

²This might very well be only a theoretical advantage.

more typing. Detailed knowledge of the query syntax might be reasonable to expect from users who frequently use a given system, but casual users are unlikely to have this expert knowledge.

To accommodate casual users and to reduce the amount of typing involved with query formulation, advanced query forms can be provided. These forms would typically have additional text input boxes for the most frequently used document fields, such as title, author and publication year. The query parts entered into these input boxes are connected by a fixed Boolean operator (e.g. AND), reducing the expressiveness of such forms compared to the query syntax element “field” described above. Another way of helping non-expert users are quick lookup facilities like those provided by the IEEE Xplore interface: in Figure 6.1 the two buttons above the query input field, labeled “Data Fields” and “Operators” open drop-down lists with common choices and can insert the chosen item (e.g. a meta-data field) automatically into the query.

Whether a single search box or an “advanced” search form is the better option is still not known for sure. On the one hand, there are advanced query forms on many web search sites and digital libraries. This prevalence is evidence for the usefulness of an advanced search form. On the other hand, studies showed that users tend to be reluctant to use these forms.

The extent to which an advanced form supports the user depends on a cost/benefit analysis. The cost can be estimated using the GOMS model:

Azzopardi et al. [3] examined how different query form designs affected user behavior. They differentiated between three designs, one of which was a grid layout of search text boxes. They argued that, based on the GOMS model, using the grid view is more expensive to users than the single search box. The single search box, in turn, is more expensive than the same single search box extended with query suggestions, also based on the GOMS model. They found that users using the grid layout form issued fewer queries and examined more results. The problem in this case is that the study used a newspaper collection and that newspaper articles have few if any fields, while “advanced” search forms are often all about querying different fields. So the cost associated with the more complex interface is not countered by a gain in the ease with which to query for specific fields.

Tjin-Kam-Jet et al. [118] examined users who used a complex search form and a single query text box for searching a public transport web site for train routes. They found

6. Support for Specification

that users were faster when using the single search box. The findings might be flawed by noise and order effects: The task completion times reported were in the order of seven minutes, which makes it unlikely that this covers only dealing with the search interface. The rest of the interaction might have played a much bigger part, making systematic differences between the query form variants likely to disappear in the noise of the main part of task processing. Additionally, the single query text box was consistently examined as the second stimulus in a repeated-measures design, which is prone to introduce learning effects. It is also unclear whether the findings can be applied to full text search and, if so, to which subdomain of it.

Yuan [147, pp.82] studied fielded query forms, among other things. She found that it took significantly less time to use a fielded query form for searching quotations in electronic books than to use a single query field.

While there is evidence that advanced search forms come with a cost, there is little evidence that there is a benefit from them, apart from the occasional increase in specificity (and therefore, precision), e. g. when a query contains an author name that doubles as natural language word like Baker. In these cases, entering “Baker” into the title field of an advanced form is easier for a casual user than figuring out the correct formulation of the query if no title field exists.

If a search system is required to offer ways of expressing complex queries and a single query box is used instead of an advanced form, the complexity has to be reflected in the query language. This usually involves query operators such as AND or +, which have been examined by some authors.

6.1.3. Query operators

Eastman and Jansen [44] studied the influence of Boolean query operators on the results of web search queries and found that they indeed make a difference, depending on the search engine and the type of operator used. The study was performed by running a set of queries that used advanced operators like AND on a selection of web search engines and comparing the results with those of the same query after removing the operators. The study reports on measures that are more related to system-oriented research than to user-oriented measures. Moreover, the results show that it is hard to come to a general conclusion since the effects of using operators differs between the

search engines. Despite the methodological rigor of the study the question of whether the adept application of operators in search queries by human users has a positive impact on the user is not answered.

Duan et al. [41] examined the effect of automatically inserting operators into user-submitted queries. They started with the observation that queries submitted to web search engines often lack any advanced operators such as the plus sign, which marks a required term, and the double quotes enclosing phrases. Their system inserted such operators using machine learning algorithms. An experimental evaluation indicated that the plus operator improves long queries and that phrases improved short ones. Combining operators yielded the greatest benefits for both short and long queries.

6.1.4. Incremental query building

If query languages offer a rich set of features for expressing complex information needs, users sometimes struggle with their complexity. One approach to assist the user with writing complex queries is to provide a way to incrementally build a query, starting with an easy one.

Demidova et al. [36] introduced FreeQ. FreeQ is a tool for assisting the searcher with incrementally creating a query for FreeBase, a large-scale open ontology. The problem with FreeBase is that the ontology is very large and simple keywords often match many different types and instances, making rankings difficult to interpret. Better queries exploit schema information for greater precision but this requires that users know both the schema and the query language. FreeQ offers interpretations of simple keyword queries to the user and lets the user decide on possible interpretations of this query that can be iteratively narrowed down until the user is satisfied with the result. No study has yet been conducted to evaluate FreeQ or compare it to other query building approaches.

However, users not only have difficulties with the logical structure of complex queries but each individual term can be challenging as well. The section below describes support mechanisms for spelling and translation.

6.2. Spelling correction

Computer users make all sorts of mistakes while typing text. Word processors have taken this into account for a long time by including spelling corrections as a key feature. The standard way of communicating information needs to the search system involves typing text, so at first glance it looks likely that queries also contain spelling errors. Some crucial text processing steps in a search engine (e. g. stemming), but also the search itself, will work only if the query terms are spelled correctly, so spelling corrections appear to be useful. The next section elaborates on how often errors actually occur. It still might be that errors in queries occur frequently but correcting them does not have a huge impact, so the section that follows the next summarizes studies on the effect of spelling errors on search success. As an example, the spelling correction feature in ezDL is also discussed in the third section, along with a study on its merits.

6.2.1. Prevalence of spelling errors

Spelling errors can, in principle, make queries worse to unusable. However, users might rarely misspell their queries so that support of a spell checker provides little help to them. Intuitively, this would be in contradiction to the prevalence of spell checkers in word processors, but empirical data answering this question also exists.

Cucerzan and Brill [33], for example, stated that 10–15% of all queries in web search engines contain spelling errors, but they did not provide any citation or any other basis for that claim. Other studies report a similar prevalence of spelling mistakes.

Li et al. [89] examined spelling corrections using a hidden Markov model approach. They used two datasets in their study, one from TREC with 5892 queries and one from MSN with 4926 queries. In the TREC dataset, 5.3% of queries contain spelling errors, while the MSN dataset contained 13% erroneous queries. Sun et al. [116] examined spelling corrections using Multitask Learning. They used the same TREC dataset as Li et al. and also two additional sets of data. The first was collected from AOL, with 16.7% of the 12000 queries erroneous. The other was an MSN dataset that they reported to have “about 11%” queries with errors.

6.2. Spelling correction

Dalianis [35] reported that 10% of all queries examined in a study contained spelling errors and that 92% of the errors could be corrected.

These latter data points describe queries in web search systems. Thus, it is debatable to which extent these findings can be generalized to all IR system usage scenarios. However, human beings are generally prone to making mistakes, and composing erroneous queries is of serious consequences only in rare cases. So it can be assumed that in many scenarios the percentage of errors in queries is in the range reported by above studies—i. e. between 5% and 17%. In this range, the support of a spell checker seems to make sense. But would spell checking also make a difference in search success?

6.2.2. The influence of hard-to-spell terms

Willson and Given [137] conducted a study on spelling errors. In that study, the participants who dealt with hard-to-spell terms were much more likely to check the spelling. Since the OPAC under scrutiny did not provide spelling correction, the participants had to use external web resources to check their spelling. Willson and Given found that hard-to-spell terms result in less successful searches, so it seems that searchers will profit from spelling corrections if they make mistakes.

6.2.3. Spelling corrections in ezDL

Gustak [59] examined proactive spelling suggestions using ezDL and found no difference in session recall and session precision between users who had spelling corrections available and those who did not. In this study, the participants who had no spelling correction available during the tasks corrected their spelling errors themselves. Gustak noted that the spelling module was actually never used.³ The fact that Gustak was not able to show the utility of spelling corrections might have been caused by the size and selection of the sample, but also by the type of search system used: ezDL is a system aimed at searching in semi-structured document collections; web search systems might have users who behave differently.

³German original: “Die Schreibweise der Suchterme war bei alle [sic!] Probanden korrekt und löste somit das Modul nie aus.”

6. Support for Specification

In fact, Cucerzan and Brill did find errors in queries [33], but those findings are based on an examination of web search queries. So one possible reason for the different observations may be that the error rate differs in the populations observed in the two studies. Another possible reason is that while Cucerzan and Brill report queries “in the wild”, Gustak observed participants in a lab study. This artificial setting might have caused some bias towards correcting queries better than usual.

6.3. Query translations

Spelling corrections are effective because users frequently make mistakes while formulating their queries and these mistakes produce worse queries. This is true even if the users write queries in their native language.

When searchers deal with document collections written in a language other than their native language, new problems occur: if the system does not offer any support for it, the users will have to translate their requests into the documents’ language. In known-item searches, this might not be a problem—especially if bibliographic information of the target document is known. Whereas in exploratory search, which is connected with creative thinking and learning, the user might want to use advanced search tactics such as searching for the antonym. In these situations, translations might be useful.

Lopes and Ribeiro [91] examined query translations in the context of medical search. They found that translating queries for medical tasks performed by non-native speakers of English is indeed useful, and, even more so the less language proficient the user is.

Gustak implemented a translation module for the proactive suggestions framework in ezDL and evaluated it [59]. He found that the translation module (German to English and vice-versa) improved both session precision and session recall.

6.4. Query suggestions

The specification support features described so far work at the term level. An alternative to this is examining whole queries and providing suggestions on this level.

Hughes-Morgan and Wilson [68] found that result-page query suggestions, as offered by web search engines like Bing, result in significantly fewer queries issued than using faceted filtering or hierarchical clustering for simple tasks. In exploratory tasks, these simple query suggestions resulted in a significantly higher number of queries being issued than for filtering, and more time being spent than for both filtering and clustering. These results might have been biased through the fact that web searches are typically short ad-hoc lookups⁴ and, as a result, the popular earlier queries that are suggested might usually be queries for short ad-hoc lookups and not for exploratory search tasks.

Gustak [59] found that offering suggestions of queries by other users improved session precision. An improvement of the session recall was not observed.

Kelly et al. compared term and query suggestions. They found that query suggestions lead to higher satisfaction ratings but not to better performance [83]. This finding could have been a result of the way the term suggestions were presented. The placement of the suggestions was next to the result list, but it is not clear if the suggestions were shown during query formulation or only after results arrived. (The description of the pseudo relevance feedback procedure suggests, after.)

In another study, Kelly et al. [82] examined whether users are influenced by social hints (e. g. star ratings) and the quality of query suggestions. They found that while social hints do not make a difference in preference, users can judge the quality of query suggestions themselves and prefer their own judgments to the social hints.

Kato et al. [81] worked on suggesting queries that move the query to a sibling term (e. g. “canon camera” → “nikon camera”). They compared two ways to present these suggestions: a clustering-based variant called SParQS and a flat list as a comparison baseline. They found that the usefulness of the SParQS design depends on the type of query and the type of measure: in information gathering tasks, the list interface produced more answers and documents than the SParQS interface, but in entity comparison tasks, the SParQS interface produced more answers and documents. The search success rate, among other measures, was greater for the SParQS interface for both of the two task types.

⁴Jansen et al. [75, 76] reported that 53% of examined sessions from `alltheweb.com` consist of only one query and 18% of two, some of which are probably respelled variants. The mean number of terms per session was reported as 2.8. The findings of Broder [27] and Rose and Levinson [103] can be interpreted in a similar manner.

6. Support for Specification

Jain and Mishne [73] examined several ways to present query suggestions to the searcher using clustering. They concluded that searchers prefer the clustered variant.

These findings present evidence in favor of offering query suggestions. If query suggestions are to be shown, clustering them might help the user.

6.5. Proactive support

Sometimes users are unwilling to explicitly invoke interface functions to get help with advancing their search. One reason for this might be the Lake Wobegon effect: In a series of experiments, Dunning and Kruger [87] found that the most incompetent participants overestimated their own competence the most. If the findings of these experiments transfer to the domain of interactive IR, searchers might tend to overstate their own search proficiency and believe that their queries are already pretty good. This would reduce their desire to invoke interface functionality because, from their perspective, it would have a bad cost-return ratio.

As a workaround to this problem, search suggestions have to be offered in such a manner that they appear as low-hanging fruit to the user: it must be obvious that there is a way to improve the search and that it is very easy to do so. One way of achieving this is to offer suggestions proactively: the user does not have to invoke the function because the system does. In Bates' levels of system involvement [9], proactivity would be level 3b.

The next sections summarize research works on proactive functions.

6.5.1. Agent to Improve Information Retrieval Systems

Jansen and Pooch [74] implemented "Agent to Improve Information Retrieval Systems" (AI²RS), a system to improve existing retrieval systems. They integrated AI²RS into Managing Gigabytes (MG) and studied how 30 participants worked with the system to complete two different tasks from a TREC collection. The participants could open the

agent by clicking on a button that appeared when the system had assistance to offer.⁵ Jansen and Pooch found that all participants looked at the assistance feature but not all used the suggestions offered. Still, AI²RS improved the precision of the MG system significantly, but at the cost of increased workload.

6.5.2. Proactivity and reactivity

White and Marchionini [135] examined three modes of providing query suggestions to users. In the baseline system, no query suggestions were made. The real time system provided the user with suggestions in a box next to the query text field while the user was typing the query. The third, retrospective, system offered query suggestions after the user started the search and the system presented the result list. They found that the real time system improved the quality of the initial query significantly when compared to the other two systems tested. Neither system was able to improve precision at 10, regardless of task type.

6.5.3. Proactive suggestions in DAFFODIL and ezDL

Schaefer and Jordan [79, 104, 105] implemented and evaluated proactive suggestions for the DAFFODIL system (see Figure 6.3). In these works, the suggestions opened automatically when the user made a typing break. The system offered help with a spelling correction tool, suggestions of past queries, suggestions for term completions for a given prefix, related terms, synonyms and author-name completion. Some of these functions were only prototypically implemented to return content useful for the evaluation of these functions. Study participants rated the system high on both usability and retrieval quality scales, but the collected performance data about the proactive suggestions was inconclusive and could not show improvements in general.

The subsystem used to retrieve and show the suggestions was the basis of the proactive suggestions in the DAFFODIL successor ezDL. Gustak [59] used ezDL to study the helpfulness of various kinds of term-based and query-based suggestions and found improvements from translations and query suggestions (see Sections 6.3 and 6.4).

⁵In Bates' levels of system involvement [9], this would be level 3a, because the user still has to request the assistance of the system.

6. Support for Specification

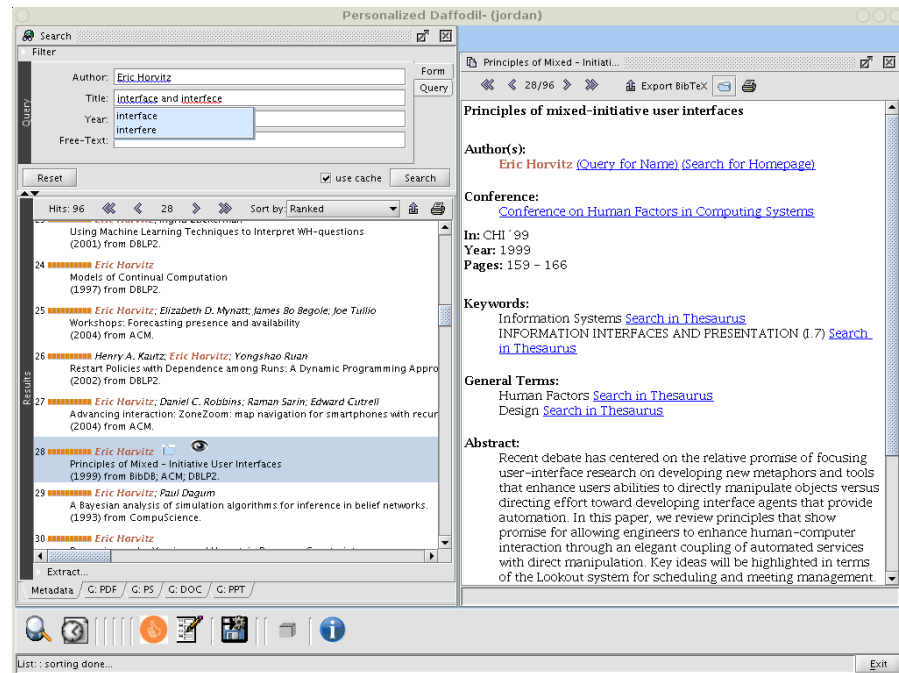


Figure 6.3.: DAFFODIL's proactive suggestions

6.5.4. How to display term suggestions of multiple kinds?

Ignalski, Jordan, and Kriewel [70] examined how a mix of two different kinds of proactive suggestions (synonyms and spelling corrections) should be displayed to the user. The two different list designs were a sorted list in the first condition and a tabbed list in the second condition (see Figure 6.4). The tabbed list included a separate tab for each category of suggestion—i. e. one tab for spelling corrections and one for the synonyms. After 18 participants completed three tasks with one of the design variants, none of the examined variables click rate, learnability, search process rating and number of fixations showed a significant difference, probably due to the small sample size. Joho et al. [78] found that hierarchical presentation of expansion terms provides greater benefit than listing them flatly.

These studies show that the proactive presentation of suggestions is accepted by users and that it helps them with some aspects of their searches.

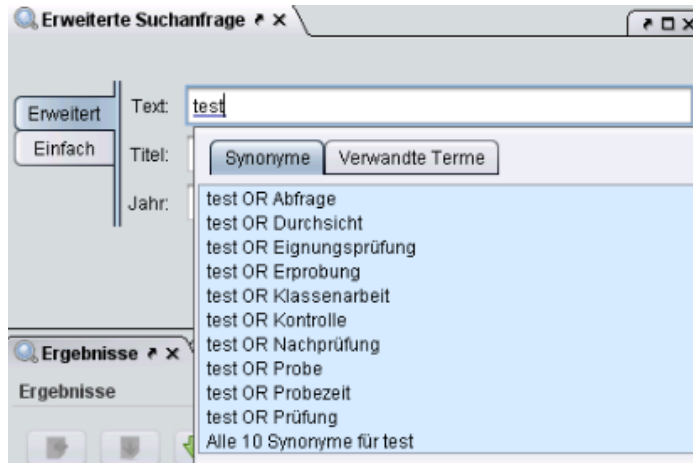


Figure 6.4.: List variants for proactive suggestions

6.6. Conclusion

One important way of specifying an information need is a search query typed into a query text field. Being able to express complex information needs requires complex query languages. This results in many possible difficulties that searchers might encounter: Searchers need to be able to formulate complex queries; they risk making mistakes while typing their queries; they might have a language barrier to overcome and the words they choose match those used by documents' authors only incidentally. There are some means to help the searchers cope with these issues. Some systems, e. g. ezDL, offer rich query languages. Incremental query building helps users develop complex queries. However, whether spelling correction helps is not clear: spelling errors are frequent and correcting them improves queries, but there are environments in which spelling corrections are never used. What will frequently be of help to the user, is a tool for translating words into the language of the document collection. Suggesting whole queries helps users if the context is right. And the correct way of presenting the user suggestions is during query formulation, not afterwards.

Now that approaches for supporting scanning, searching, recognition, and specification have been summarized, the question of how to combine them for giving users, who use different information-seeking strategies during longer search sessions, maximum support can be tackled.

7. Combining Optimal Support Mechanisms

The previous four chapters dealt with support mechanisms for each of the four facet values scanning, searching, recognition and specification. The question was, how can IR systems support users engaged in an information seeking strategy that incorporates the concerning facet value?

This chapter details the experiment that was conducted to examine the main research question: how does an adaptive user interface compare to a static one that provides all the support features at once, and how do these compare with a baseline system without sophisticated support mechanisms?

7.1. Research question and hypotheses

Extending the ideas outlined in Section 1.3, the main research question can be split into two parts, each of which can be reduced to a single hypothesis:

7.1.1. Research question A

Is it better to support searchers with specialized functions customized for each ongoing ISS in an adaptive user interface than with a baseline interface that does not have these functions?

Hypothesis A: Searchers using a system whose interface adapts the set of support features to the current search action are more efficient than those using a baseline system without any support features when performing mixed tasks.

7. Combining Optimal Support Mechanisms

7.1.2. Research question B

Is it necessary to offer support functions adapted to the ISS the user is currently engaged in, or is it feasible to combine these support features into an integrated interface?

Hypothesis B: Searchers using a system that has all support features provided by the adaptive system enabled at the same time are similarly efficient to those using the adaptive system and not more stressed when performing mixed tasks.

The benefit of the integrated system as compared to the adaptive system would therefore be that the system would not have to predict the user's next step, nor would the user have to state which step they intend to perform next.

7.2. Related work

Some researchers examined the benefits of adaptive or adapted systems.

Diriye et al. [38] [39, p.81] examined the relationship between the features of the search interface and the kind of tasks that users performed with them. They used two versions of a system. One version was a baseline system aimed at known-item searches and the other one was a system based on the baseline system, but with added features aimed at exploratory searches. Additional features in the exploratory version were query suggestions, query previews, and the display of concepts related to suggested queries. The 16 participants worked on two tasks with each version of the system: one known-item search and one exploratory task. Task completion time and interactive precision were used as measures along with the number of uses of each feature per task. Additionally, questionnaires about the interfaces and tasks were completed by the participants.

Many analyses were performed with the data gathered in the experiment, but the paper does not mention any correction for alpha inflation, so the findings have to be taken with a grain of salt. While the known-item tasks did not provoke any significant difference between the baseline and exploratory interface, the exploratory tasks showed an advantage of the exploratory system over the baseline concerning interactive precision. An interesting finding was that the known-item tasks were completed faster on

7.2. Related work

the baseline system than on the exploratory system, while the exploratory tasks were not completed faster using the exploratory system. Concerning the questionnaires, the participants reported the exploratory interface to be more distracting than the baseline version.

Yuan and Belkin examined this subject in their 2010 study [147, 148, 149]. They came to the conclusion that users perform better if they have a system available that is specialized in the task they are to perform.

Yuan's dissertation tackled three research problems: The first was to implement and evaluate systems for different ISSs, the second was implementing dialog structures for transitions between ISSs so a search system can adapt to multiple ISSs, and the third and last problem was to evaluate the system that adapts to several ISSs.

In order to find support mechanisms for some ISSs, experimental systems were designed and tested to four kinds of tasks. The tasks that the experimental system had to support were finding the best databases for a given topic (1.1), finding quotations from an electronic book (1.2), finding relevant documents on some topic in a database (2.1), and finding the name of an electronic book that contains certain quotations (2.2).

Some of the systems designed to support these tasks were very specialized systems. For example, the system for the first database summarization task, 1.1, was a system that summarized databases. This system was tested against a baseline system that only searched the given databases and reported for each document in which database the latter was found. The system for task 1.2 presented full document contents along with the Table of Contents. It was compared to a system that treated each document's paragraph as a separate document and offered fulltext view of its contents. The system for task 2.1 provided clustering while the baseline system was a simple one-query-field search system, very similar to the one for task 1.1. The system designed to support task 2.2 used fielded queries, and was compared to a single-field query. The results showed significant and meaningful time savings in two tasks, 1.2 and 2.2 (about 1 standard deviation each), and significantly better user satisfaction in 1.2.

The second research problem examined in the thesis (listed as research problem 3 by Yuan) was the evaluation of a system that adapts to the available ISS. The dialog structure of the system supported two ISS transitions: scanning, then searching and

7. *Combining Optimal Support Mechanisms*

searching, then scanning. It turned out that the experimental system that adapts to the task at hand enabled participants to find more relevant aspects (aspectual recall), with the effect size being about half a standard deviation, while the participants used fewer iterations with the adaptive system.

A problem with the statistics in this work is that the hypothesis of research problem 3 (“An experimental system . . . performs better . . .”) was not properly operationalized into a success metric. Instead, lots of variables were measured and reported individually regarding differences between conditions, but no mention of a correction for alpha inflation could be found.

Another factor that limits the findings of Yuan’s work is that the tasks and support mechanisms were very closely related: it is no surprise that a system which provides a summary of several databases supports obtaining an overview over several databases better than a search system.

Both these works on adaptive search systems have some issues that limit their generalizability, so a new experiment was designed and conducted.

7.3. Selected support mechanisms

The facet value “searching” was supported by the baseline version of the search system because this is a baseline activity and no special features are required for it other than what is usually found in search systems.

For the remaining values of “scanning”, “recognition,” and “specification,” the following techniques were assumed to be potentially supportive and were used in the experiment:

- Scanning
 - Berrypicking tray
 - Markers for documents already found
 - Saved searches

- Faceted result list
- Recognition
 - Query-dependent surrogate highlighting and book cover scaling
- Specification
 - Spell checker
 - Translation
 - Synonym and related term suggestions

The following sections give details on how these support mechanisms were implemented in the experimental search system.

7.3.1. Scanning

Berrypicking tray

The Berrypicking tray is a tray-like tool to collect search results over multiple queries. See Section 3.4.1 for a discussion of the research literature. The tray used in the experiment is the standard implementation of the tray tool in ezDL (Figure 7.1).

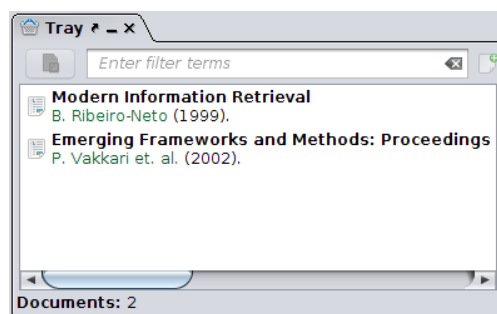


Figure 7.1.: The berrypicking tray

7. Combining Optimal Support Mechanisms

The question could arise if it is reasonable to assume that a tray is a scanning support mechanism rather than a baseline feature, considering that a tray has been a feature in DAFFODIL since about 2008. One argument for counting a tray as a scanning support rather than a baseline feature is that virtually no popular web search engine has offered this in the past. Also, a tray clearly is a feature that goes beyond basic search and is offered to support collecting multiple documents. It can be interpreted as an (intuitive) decision by the developers of a search system to support “Scanning”, even if the decision was not connected at all with the ISS classification.

Markers for documents already found, and retrieval histograms

White and Roth postulated that exploratory search systems should help the user with keeping track of the search and “record what has already been seen” (see section 3.4.2) [136, p. 57].

In order to support this requirement, a new document marker for the result list, called “retrieval histogram,” was introduced. Retrieval histograms are based on the work of Golovchinsky et al [55, 56, 57]. The marker shows the relevance of a document over the course of the past queries within the search session. To that end, an internal data structure stores triples of document ID, a query counter and the corresponding relevance, normalized to the maximum relevance of all documents in the same result list.

The icon is a bar graph that has the most recent relevance bar in full color and wider on the right and the past relevances in decreasingly full color and narrower towards the left. See Figure 7.2a for an example. The user can mark the document as relevant or irrelevant. In this case the bar graph is replaced, respectively, by a “check” symbol (Figure 7.2b), or a “cross” symbol (Figure 7.2c).

In two small-scale formative evaluations based on paper-prototyping [102], several design variants were compared with each other. The consensus among all 10 participants was that the natural reading direction is from left to right, so the bar of the most recent query should be the rightmost one. Some participants favored a left-justified graph, adding new bars to the right and starting to scroll out older bars to the left when the place was used up. Some participants favored a right-justified graph, adding

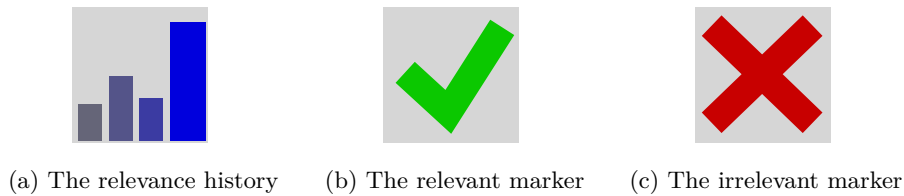


Figure 7.2.: Relevance markers

new bars to the right – just like an EEG graph would be plotted on paper. Assuming that, among the general population, few users would use the EEG-plotter-based model to understand the graph, the former variant was implemented. See Figure 7.3 for a screenshot of the final design.

The document markers can be manipulated by using the corresponding button in the detail view, by selecting the corresponding menu item from the context menu in the result list, or by copying the document into or removing it from the tray. The detail view relevance button can be used by clicking on it or by using the mouse wheel while hovering over it. In the former case, an overlay opens that presents the three choices so the user can select one of them. In the latter case, moving the mouse wheel upwards moves the relevance assessment towards a more relevant value—i. e. from “irrelevant” via “don’t know” to “relevant”—while moving the mouse wheel downwards moves the assessment in the opposite direction. See Figure 7.4 for a screenshot of the button.

Documents marked as “irrelevant” had their surrogate printed in light gray to make them seem to disappear slightly and thus make them less prominent in the result list.

Saved searches

Saved searches can be used to deliberately re-run a previous search or to alter it. Sometimes users have multiple choices to continue searching after inspecting a result list. For instance, if the result list included too few relevant documents, some query terms could be deliberately misspelled or replaced by a translation. If the user is not happy with their choice, they can use a saved search and try to alter it in a different way. In long-term scenarios, saved searches can also be used to see if there is anything

7. Combining Optimal Support Mechanisms

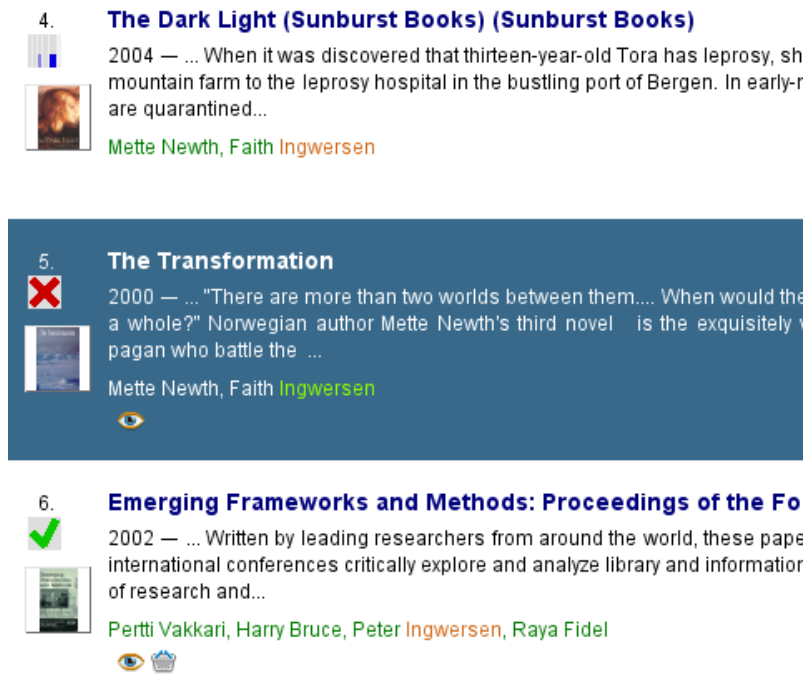


Figure 7.3.: Document markers in the result list

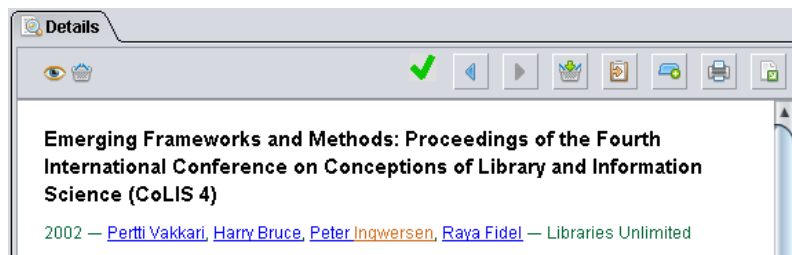


Figure 7.4.: Three-state relevance button

new about a subject that the corpus previously did not include or that went simply unnoticed. This would also fit to “monitor” in Ellis’s model.

The saved searches feature in this experiment was implemented using ezDL’s query history tool (Figure 7.5).

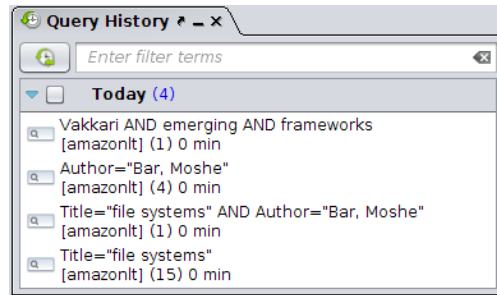


Figure 7.5.: The query history tool

Faceted result list

The faceted result list divides the result list items into subgroups that correspond to the values of certain document facets, such as the publication year or the author names. See Section 3.4.2 for a more detailed discussion.

The faceted list used in the experiment was the basic version from ezDL using the above-mentioned facets (see Figure 7.6).

7.3.2. Recognition

Query-dependent surrogate highlighting

Some aspects of the information need are important but difficult or impossible to specify. Query-dependent surrogate highlighting emphasizes places where these aspects occur in the result list surrogates. One example is searching for female authors. Highlighting all author names was intended to help with visually searching for names that appeared to be female. See Chapter 5 for two experiments that examined this support mechanism.

The implementation used in the experiment worked this way: The system chose the highlighting configuration suitable for the task at hand to abstract from the user's competency to choose the right selection of highlighted fields. The user could override

7. Combining Optimal Support Mechanisms

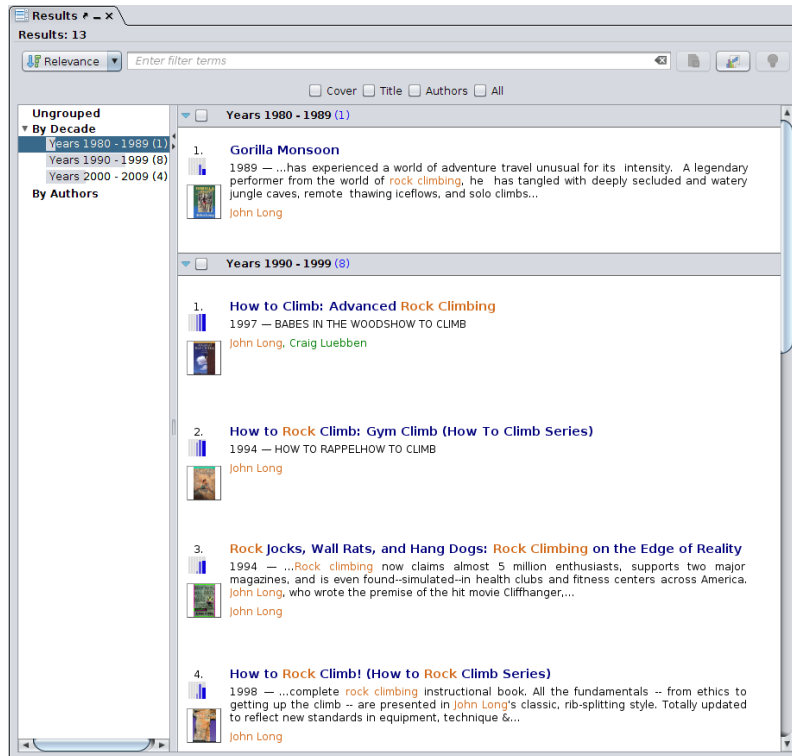


Figure 7.6.: The faceted result list

the configuration manually using a group of checkboxes in the result list. The textual fields “author” and “title” were highlighted using a yellow text background, the cover image was highlighted by visibly enlarging it. Figure 7.7 shows the highlighting checkboxes in their result list context. Figure 7.8 shows a highlighted document title and Figure 7.9 shows highlighted author names. Figure 7.10 shows an example of an enlarged cover.

7.3. Selected support mechanisms



Figure 7.7.: The checkboxes controlling the highlighting

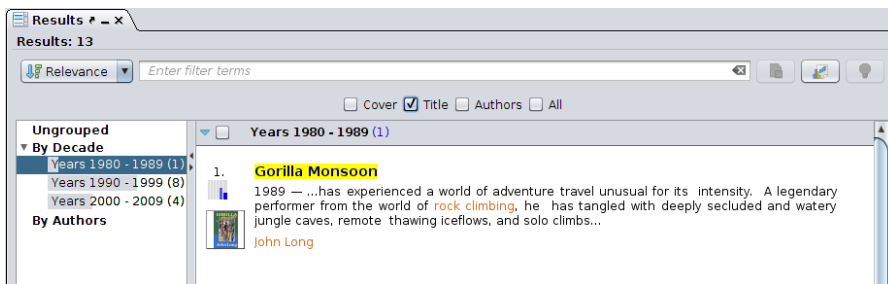


Figure 7.8.: A result item with highlighted title

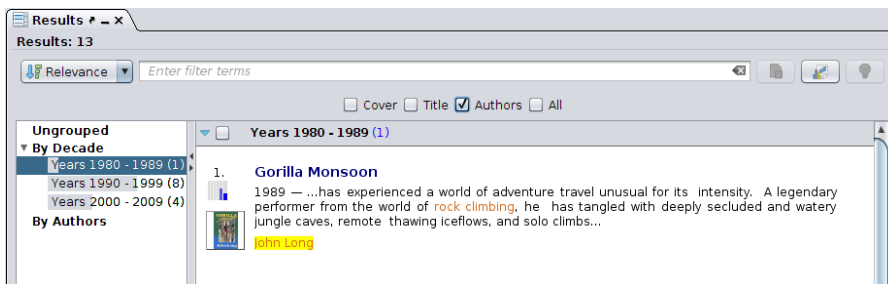


Figure 7.9.: A result item with highlighted authors

7. Combining Optimal Support Mechanisms



Figure 7.10.: A result item with enlarged cover image

7.3.3. Specification

The support features for specification were proactive suggestions of various kinds. The following sections give details of the suggestions offered. See Figure 7.11 for a screenshot of the suggestion pop-up during query formulation.

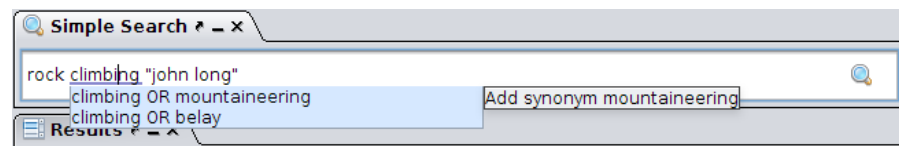


Figure 7.11.: Proactive suggestions

Spell checker

The spell checker was intended to help the user avoid spelling mistakes (see Section 6.2.3). The spell checker was implemented using the basic spell checking tool from ezDL. This in turn is provided by a proactive module that queries the term information agent (TIA) in the backend for spelling corrections. The TIA invokes a local `ispell` process to get its information.

7.4. The three variants and experimental conditions

Translation

The participants were recruited at a German university, so it was assumed that most of them were fluent in German and proficient in English. Because of this, the task descriptions were given in German. Since the corpus used for the experiments comprised English books, users had to translate terms from the German task descriptions into English.

Some tasks contained terminology not assumed to be in the active vocabulary of the typical participant, so the translation feature was offered to help the German-speaking participants formulate queries for English books. The implementation used in the experiment was the one from Gustak's study (see Section 6.3).

Synonym and related term suggestions

In some situations, searchers want to extend a query with additional terms. Synonym and related term suggestions were intended to help users formulate more elaborate queries. The synonym and related term suggestions subsystem was implemented using a proactive module reading a small handcrafted thesaurus from a file to get its information. The file was edited to include terms that were expected to occur in the tasks the participants were asked to perform (see Appendix F for the thesaurus terms).

7.4. The three variants and experimental conditions

Based on the description of the system in the hypotheses in Section 7.1 and using the support mechanisms detailed in the previous Section 7.3, three experimental systems were built: Baseline, Experimental A (the adaptive system), and Experimental B (the integrated system). The following subsections present how the systems used in the experiment were constructed using the ezDL framework.¹

¹Note that the screenshots presented here are translations; the original German screenshots from the software actually used in the experiment can be found in Appendix E.

7. Combining Optimal Support Mechanisms

7.4.1. Baseline

The baseline interface shown in Figure 7.12 did not have any built-in special support mechanisms that exceed normal practice. In the ezDL framework, this meant including the search tool with a basic result list and the detail tool.

Since no support mechanisms were included in the baseline system, the proactive suggestions containing the translation feature were not available to the participants, either. They still had to translate the German terms used in the task descriptions into English. To solve this conflict in the baseline variant, the translation module was replaced by a tool within the ezDL client that used the translation page of bing.com. This choice was made to not violate the *ceteris-paribus* assumption, because the Bing backend was also used for the proactive translations in the other conditions.

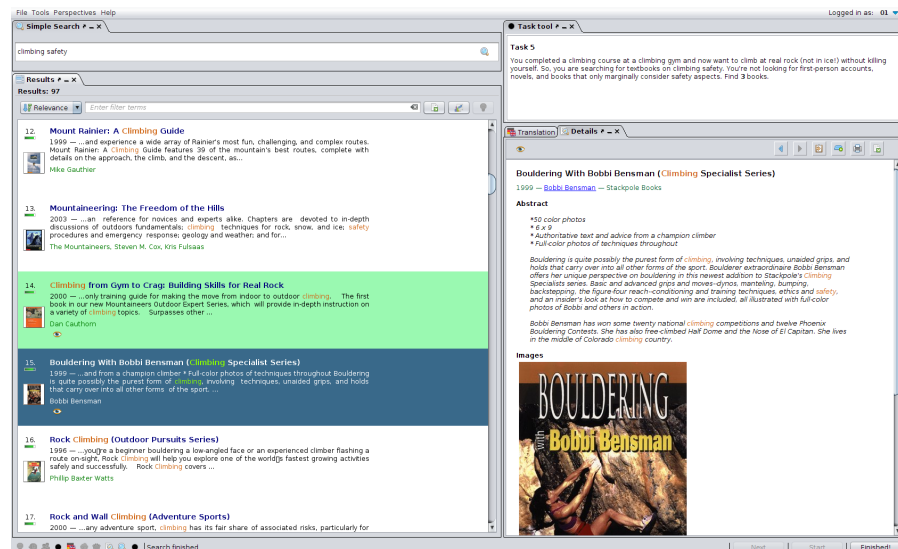


Figure 7.12.: The baseline interface

7.4.2. Experimental A (adaptive)

The adaptive interface shown in Figures 7.13 and 7.14 was adapted automatically to the ISS of the task the user was about to perform. The support mechanisms provided

7.4. The three variants and experimental conditions

were a combination of those mechanisms listed for each relevant facet value in Section 7.3. Some features were implemented as tools in ezDL (e.g. the Berrypicking tray). This meant that enabling or disabling this and other features changed the general appearance (the layout) of the interface with regard to subwindows. This can be seen in the above-mentioned figures.

Initially, proactive suggestions, including translations, were intended to be offered only in specification tasks: these tasks involved formulating queries for searching in an English corpus and the participants, because they were recruited at a German university, were not expected to be perfectly proficient in English. However, even the recognition tasks required the participants to enter an initial filter query. Therefore, the participants had to overcome a language barrier in these tasks as well. To solve this problem, the translation module was enabled in all tasks, specification and recognition. All other modules that were intended for specification support were only enabled in the specification tasks.

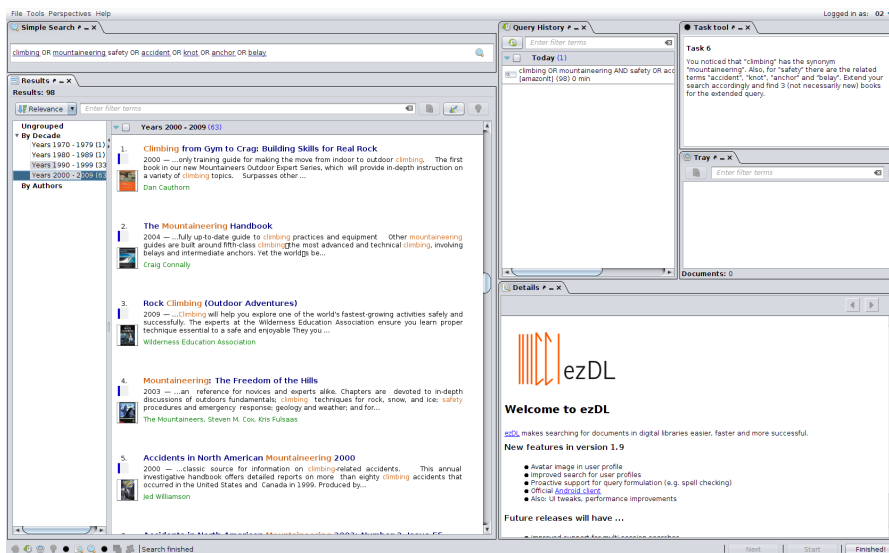


Figure 7.13.: The adaptive interface for scanning/specification

7. Combining Optimal Support Mechanisms

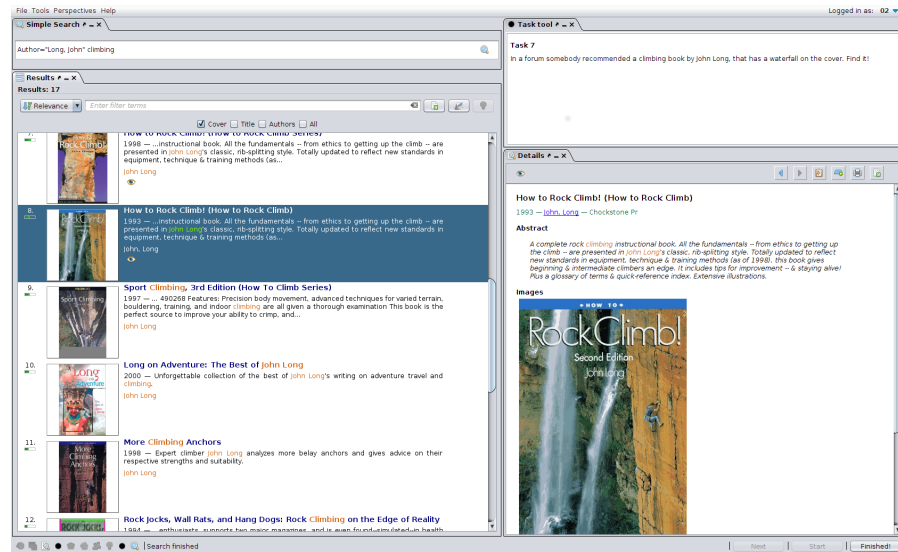


Figure 7.14.: The adaptive interface for searching/recognition

7.4.3. Experimental B (integrated)

The integrated system shown in Figure 7.15 included support mechanisms for all supported ISS classes. All features were available at all times and the surrogate highlighting was configured based on the model of an experienced user.

7.5. Pilot test

The initial study design, including the three interface variants used in the three experimental conditions, was tested with four participants in order to identify problems.

One participant suggested to increase the font size of highlighted surrogate fields. This was not done because the trade-off between the aggressiveness of highlighting and the space available for displaying text seemed to be bad. Also, the highlighting alone seemed to be noticeable enough.

One suggestion was to offer a prototype of the surrogates for highlighting configuration. This was not implemented because of several UI design issues, e.g. available space

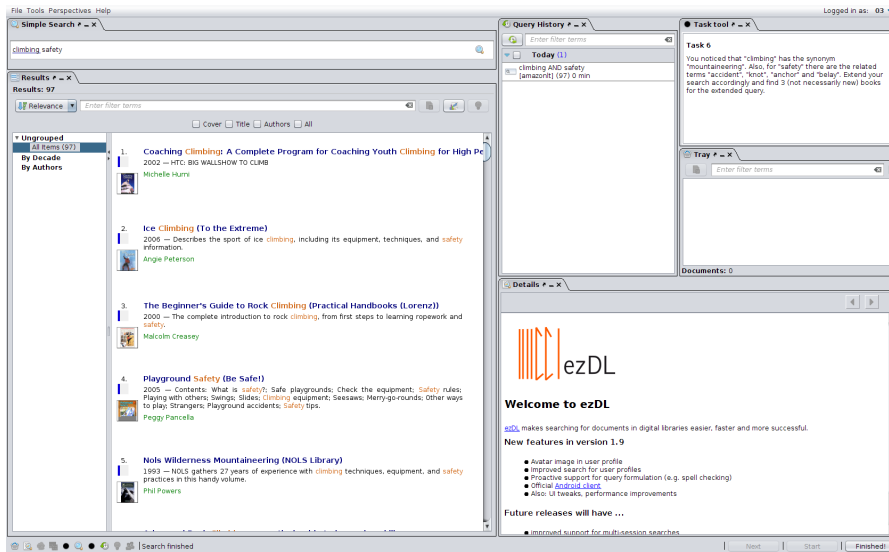


Figure 7.15.: The integrated interface

in the interface and the difficulty to discriminate between the prototype and actual result items. Another consideration was that the system was supposed to configure the surrogate highlighting automatically anyway, so manually overriding the configuration was assumed to occur only rarely.

One suggestion that was implemented was to gray out surrogates whose documents were marked as irrelevant. The idea behind this design was to make the documents seem to move into the background and allow the other documents to stand out more.

The idea of automatically marking documents as relevant or irrelevant when they are, respectively, dragged into or removed from the tray tool, was also implemented. The other direction—adding documents to or removing documents from the tray which were marked as relevant or irrelevant—was not implemented, because it seemed to involve a grave consequence emerging from a rather casual interaction. This caused a possible conflict in the evaluation: theoretically, a user could drag a document into the tray (automatically marking it as relevant) and then mark the same document as irrelevant in the result list, resulting in two conflicting relevance judgments. To avoid this, users were explained that the documents, which were regarded as solutions to the task, were the combination of those documents marked as relevant in the result list and

7. *Combining Optimal Support Mechanisms*

those in the tray. The users were also informed that the difference between not marking a document as relevant and explicitly marking it as irrelevant was only meaningful as a “note to themselves” and that the system only considered these documents equally to be not marked as relevant, and therefore not as solution documents.

One participant remarked that users might like to see more relevance bars in the retrieval histogram and that they might want to see a tooltip explaining what they saw. Thus, tooltips with full relevance histograms were implemented which opened when the user hovered a retrieval histogram in the result list.

Based on the timings of the tasks measured in the pilot study, a limit of four minutes per task was imposed.

The following sections describe the final study design as fixed after the pilot study.

7.6. Operationalization

With the three modified interface variants as between-subjects factors, the study was designed as a single-blind randomized interactive retrieval experiment using relevance-based measures on a large document collection. While the main variables were measured between subjects, one variable—the SUS score—was measured within-subject.

7.6.1. Participants

Between June and October 2014, 53 participants were recruited using flyers handed out in lectures, posters at several locations on campus and posts in three university web forums.

Participants were randomly assigned to one of the three experimental conditions based on the order of the experimental sessions they took part in. For each of the three experimental conditions, baseline, adaptive and integrated, 12 participants were measured and their data used. The data of the other 17 participants was dropped for different reasons as detailed in Section G.1.

7.6.2. User model and tutorials

For all experimental conditions, the user model was that of an experienced user. The concepts used in the experimental systems were rather advanced, so it was assumed that casual users would not immediately be able to work with them sufficiently successfully. Moreover, it was clear from earlier usability studies of another ezDL-based system, that the general learnability of the system was not as good as expected when intended to be used by beginners [11, 12]. Additionally, it was not the intention of the experiment to measure the participants' search competence. For these reasons, suitable support features were configured automatically for the participants with the option to override the features.

Since the participants could not be expected to have prior experience with ezDL, it had to be made sure that they understood the system at least to some extent to be a good fit for the user model. To achieve this, each experimental session included a tutorial part in which the experimenter explained the system to the participant and the participant had a chance to become familiar with the system by executing four training tasks (one for each tested ISS) and by asking questions. See Section 7.6.6 for details of the process.

7.6.3. Collection

The collection used was the Amazon/Library Thing collection also used in the INEX Social Book Search track [21], restricted to those 1.2 million books that have a cover image available. This restriction was imposed to allow book cover related recognition tasks.

7.6.4. Tasks

In the experiment, the usual protocol of interactive retrieval experiments was followed: Participants completed certain search tasks and relevant variables, such as their efficiency, effectiveness, and their satisfaction, were measured.

The objective of the experiment was to study the effect of specific support mechanisms and their combination on specific variables measured in participants completing tasks

7. *Combining Optimal Support Mechanisms*

of different ISS classes. For reasons explained in Section 2.3.2, in this experiment the ISS facets “goal” and “resource used” were considered to have the static, pre-defined values “selecting” and “meta-information” in all tasks. This meant that tasks had to be found for the ISS classes defined by the remaining facets “method” and “mode” with their values “scanning”, “searching”, “recognition”, and “specification.”

Each ISS defined by the combination of these values was considered equally important, so the number of tasks for each ISS was the same. To run less risk of choosing exceptionally difficult or easy tasks, each examined ISS was covered by three experimental tasks. The tasks were grouped into blocks of four tasks that had a common background story to simulate a larger work task [24, 25]. The blocks were constructed, so that each block covered all four ISS classes considered. The subjects covered were rock climbing, the history of Africa, and user interface design and cocktails. See Appendix A for the list of tasks.

Task set permutation was performed to mitigate the risk of learning and fatigue.

7.6.5. Metrics

The research questions were broken down into hypotheses A and B as described in Section 7.1:

Hypothesis A: Searchers using a system whose interface adapts the set of support features to the current search action are more efficient than those using a baseline system without any support features when performing mixed tasks.

Hypothesis B: Searchers using a system that provides all support features in the adaptive system enabled at the same time are comparably efficient to those using the adaptive system,s and not more stressed when performing mixed tasks.

The constructs related to the experimental conditions were already defined in Section 7.4. To be able to examine the hypotheses, the remaining constructs, related to the measures, had to be defined.

Measure candidates

When examining interactive retrieval systems, a host of measures can be used to make statements about their merits.

White and Roth [136, pp. 64ff] listed several metrics for exploratory search systems. They mentioned engagement and enjoyment, information novelty, task success, task time, learning and cognition. They also described criteria for designing tasks for experiments [136, pp. 68f].

Egusa et al. used Concept Maps [46] to measure the differences in the mental representation of concepts participants had before and after an exploratory search.

Vakkari and Huuskonen [130] tested medical students in an essay writing task and found that traditional IR measures such as precision are not good predictors of work task outcomes, i. e. essay grades.

Main measures and metrics

The hypotheses derived from the research questions compare systems according to the efficiency of their users and the stress that the users felt to be subjected to. Consequently, measures for efficiency and stress had to be defined. Efficiency is effectiveness in relationship to cost, so in the next section, effectiveness is introduced, followed by measures for cost and, in combination, for effectiveness. In the sections following the next, measures for stress and user satisfaction are introduced.

Effectiveness

Effectiveness in the context of retrieval experiments is usually defined in terms of precision or recall. Both precision and recall are defined based on sets of documents found during a single query-result list interaction. They are metrics used to describe the quality of an IR engine, abstracted from human users.

When users come into play, the quality of the IR engine is no longer that important, because users can often work around the engine's shortcomings by issuing multiple

7. Combining Optimal Support Mechanisms

queries [127, 128]. Users issue multiple queries during the search regarding one information need also for a different reason: Their information need leaves them in an anomalous state of knowledge [15, 20], which makes it hard to formulate a good query right at the start. Therefore, users experiment with queries and reformulate them until they are finished searching. In some cases, the (perceived) information need shifts if the searcher learned about their situation from preliminary finds. For these reasons, precision and recall are not good metrics for interactive retrieval experiments. It is better to use metrics that take this iterative process into account, such as interactive precision and interactive recall, as introduced by Veerasamy and Heikes [132]. These metrics are calculated like their system-oriented counterparts, but they are adapted to regard the sets of documents that the user collected or marked as relevant (MARKED) during the course of the task:

$$\text{SessionPrecision} = \frac{|\text{MARKED} \cap \text{REL}|}{|\text{MARKED}|}$$

$$\text{SessionRecall} = \frac{|\text{MARKED} \cap \text{REL}|}{|\text{REL}|}$$

So, session precision is the percentage of relevant documents of all documents that the user collected during the session, and session recall is the percentage of collected relevant documents of all relevant documents.

Task effectiveness measure

Since all sets of tasks consisted of both recall- and precision-oriented tasks, the harmonic mean of recall and precision was used to average both measures to get the final effectiveness measure. This resulting measure is similar to the F_1 measure, but for interactive retrieval.

Marking

As usual with Boolean relevance-based measures, two sets of documents had to be determined for each session: the set of actually relevant documents for the task and the set of the documents the user collected.

7.6. Operationalization

Participants had several options to collect documents: In the baseline condition, documents were marked as relevant by using a context-menu in the result list. Removing this relevant marker was also possible using the same context menu. Documents marked as relevant were displayed with a greenish background if not selected or with greenish text color if the list item was selected.

In the integrated condition, marking could be done using the context menu in the result list, which provided three options (“Relevant”, “Irrelevant” and “Reset relevance”), by dragging a document to the tray tool—this would automatically also mark the document as relevant in the result list—and by using the relevance marker button in the detail view (see Section 7.3.1).

The adaptive condition used a combination of these procedures depending on the value of the method facet of the ISS class the task was about. In all tasks, the context menu could be used. In scanning tasks, the tray tool and the relevance marker button was available, as in the integrated condition. In searching tasks, no method-related advantage over the baseline condition was to be given to the user, so marking a document as relevant used the same mechanism as in the baseline condition.

Whatever marking procedure used by the participant, the set of collected documents was logged using the standard logging subsystem of ezDL.

Relevance goldstandard

The set of relevant documents for each task was determined by using the pooling method on the documents that all participants collected.

To support this, a small web site was developed and the document IDs of the pooled documents were imported into its database. The web site worked by selecting a task that had a low ratio of rated to unrated documents. From this task, up to ten documents were randomly selected and shown to the assessor along with the task to which the documents belonged. The summary page for each document contained the title, authors, publication year, the publisher’s summary (abstract), and a large cover shot.

Four assessors supplied relevance ratings using the web site. The assessors were three research assistants with the information systems working group at the University of Duisburg-Essen and this author.

7. Combining Optimal Support Mechanisms

A document was treated as relevant iff more assessors considered it relevant than irrelevant.

Efficiency

Efficiency is effectiveness in comparison to cost. The usual cost measure for interactive retrieval evaluations is time-on-task, i. e. the time needed to complete a task.

This means that the efficiency measure for one task of one user is

$$\text{Efficiency} = \frac{\text{Effectiveness}}{\text{time-on-task}}$$

The time-on-task was measured by the system as the time between the time when the user clicked the “Start” button and the time when the user clicked the “Finished” button. If the user did not click the “Finished” button within the allowed four-minutes time frame, the time-on-task was set to four minutes.

Aggregation

The effectiveness and efficiency measures for each user’s tasks were aggregated using the geometric mean: While the arithmetic mean reflects absolute changes, the geometric mean describes relative changes better, especially if the measures to be aggregated vary wildly in their values.

Post-experiment measurements

Effectiveness and efficiency were measured by observing participants completing tasks. User satisfaction and stress were surveyed afterwards using questionnaires.

NASA-TLX: Workload

The concept of stress, as referred to in the hypotheses, was defined as workload, as measured by the NASA-TLX [62], a widely used workload measure [61]. To keep stress-related impressions fresh, the first measure administered right after the participant completed all experimental tasks was the NASA-TLX, in the paper-and-pencil version². Since the NASA-TLX has originally been published in English and most of the participants were German, the German translation by Niederl [97] was used, which also added brief descriptions of each subscale to the rating sheet.

Participants first completed the rating sheet by marking each subscale. Then they were shown, one by one, 15 workload comparison cards and were asked to mark on each card the subscale that contributed more to the workload than the other.³

The TLX measure was determined by summarizing the data gathered on the comparison cards and the rating sheet according to the instructions in the TLX manual.

SUS: User satisfaction

A System Usability Scale (SUS) questionnaire [29] was completed by all participants to find out whether the trade-off between retrieval effectiveness of each interface and the associated user satisfaction effect (its assumed reduction) is worthwhile.

Despite its name, the System Usability Scale is a user satisfaction scale when the ISO 9241 definition of usability (efficiency, effectiveness, user satisfaction) is considered: none of the items deal with efficiency or effectiveness; these constructs are also unlikely to be measured reliably using self-reported measures.

Tullis and Albert [126, p.149] analyzed the distribution of SUS scores of over 129 different conditions and found that the first, second and third quartiles were at SUS scores of , respectively, 57, 69 and 77. Bangor et al. [5] examined the correlation of SUS scores and the rating participants gave on a 7-level Likert-type item and found a high correlation ($r=0.822$). The quartiles reported by Bangor et al. (62.6, 70.5, 77.8) are consistent with the findings by Tullis and Albert. The two studies come to the

²<http://humansystems.arc.nasa.gov/groups/tlx/paperpencil.html>

³See Figure D.1 for the TLX rating sheet used. The comparison cards were translated accordingly.

7. Combining Optimal Support Mechanisms

conclusion that SUS scores in the range of 80 and above can be considered “pretty good” or “excellent.”

Tullis and Stetson [125] compared the accuracy of five questionnaires for sample sizes between 6 and 14 and found that the SUS questionnaire exceeds all others starting with a sample size of 8. They also found that the accuracy of the questionnaires levels off at a sample size of 12, where SUS reached 100% accuracy in their study.

Although the System Usability Scale is an accurate and widely used user satisfaction scale, whether it is valid for comparison between subjects is not known. To eliminate related threats to the validity of the satisfaction data, users completed two SUS questionnaires: one with a common baseline (see Section 7.6.6) and one with the experimental system. The difference between these measures was used as the satisfaction measure for the experimental system.

Qualitative feedback

Optionally, participants could submit qualitative feedback in a free form field on the post-experiment questionnaire.

Kelly et al. reported that the mode of collecting feedback from the user after an experiment influences the quality of the feedback [84]. They found that, for open questions, pen-and-paper questionnaires result in shorter answers, while being as informative as electronic and interview modes. Because of this, the qualitative feedback was collected on the final post-experiment questionnaire.

Tedesco and Tullis [117] examined several different variations for formulating questionnaire items. They came to the conclusion that, of the examined ways to formulate questionnaire items, the one resulting in the best reliability was an easy sentence that ended in a semantic differential (e. g. “I found this interface boring exciting”). Following this advice, the quantitative questions on the post-experiment questionnaire were formulated in this way.

7.6.6. Procedure

The experiment was conducted in one session per participant.

The agenda of the 90-minute sessions was:

1. Introduction
2. Signing the consent form
3. Questionnaire with demographic data
4. Baseline search task and SUS questionnaire
5. Tutorial section
6. Measurement section
7. Post-experiment measures

Introduction and pre-experiment paperwork

The introduction part began with welcoming the participant and thanking them for taking part in the study. As concerns the goal of the experiment, the participant was led to believe that the experiment was about evaluating a certain program regarding its merits for book search. The participant was told that they could leave at any time during the experiment without adverse consequences and that the experiment did not examine the participant but the program they were about to use.

The introduction part was closed by presenting the agenda of the session, as given in the previous section.

Before proceeding, the participant was handed a standard consent form with the instruction to read it carefully and sign it only if they agreed with the content.

After signing the consent form, the user was handed the demographic data questionnaire and asked to complete it (see Figure D.4 in Appendix D for the questionnaire).

7. Combining Optimal Support Mechanisms

Baseline search task and SUS questionnaire

To eliminate threats to the validity of the user satisfaction data due to the unknown inter-rater validity of the SUS, users completed two SUS questionnaires: one with a common baseline and one with the experimental system.

The system used to establish the baseline was the ACM Digital Library (DL). Participants were shown the front page⁴ of the DL and asked to complete two tasks. The first task was to find a particular document; the second one was to export the document's bibliographical data in BibTeX format.

Given that the experimental tasks also included some difficulties, the ACM DL search task included a crux: it was impossible to find the target document unless a particular link on the upper part of the result list was clicked. Participants were told they had four minutes to complete the task and were offered advice if they could not find the document within three minutes. After giving the advice (clicking the particular link), participants were asked to proceed with a random document in case they still could not locate the target document.

After the participants completed the second task, they were asked to complete the first SUS questionnaire (see Figure D.2 in Appendix D).

Tutorial segment

The tutorial segment began by the experimenter explaining the key features of the given interface variant along a script. Each feature was introduced and the participant was asked to try the various ways of interacting with each feature. The query syntax was introduced in depth, going through many syntax elements (bare terms, phrases, terms with field identifier, Boolean operators, parentheses, author names, and subqueries with fields). After the participant was taught the query syntax, each participant received a cheat sheet containing information about the query language (see Figure D.3 in Appendix D). The content of the cheat sheet was also the subject of the preceding tutorial segment and it was the same in all experimental conditions. The participants were told that the cheat sheet could be used for the rest of the experiment,

⁴<http://dl.acm.org>

but that it would be a good idea to look at it before they started to complete each task, since otherwise the tasks would take them longer to complete, and that would interfere with the measurements.

While the baseline and the integrated version offered all available features at the same time in each task, the adaptive version was introduced using two training tasks that, taken together, made all features available. In all conditions, the participant was talked through completing the first task. In baseline, all features were explained using only the first training task only, so the user was asked to complete the following three training tasks by themselves. In the adaptive condition, the second task was needed to explain the rest of the features, because no task made all features available.⁵ Due to this and because more features needed to be explained in the adaptive and integrated conditions, the explanations in these conditions were split over the first two training tasks. After all features had been introduced, the user was asked to proceed with the rest of the second task, complete it by themselves and then proceed with the next two training tasks.

Measurement section

In the measurement section, participants completed 12 tasks which were divided into groups of four tasks (see Section 7.6.4). The three sets of tasks (A, B and C) were rotated using a Latin square design to eliminate sequence effects such as learning and fatigue.

The experiment did not enforce a specific number of queries per task. In a pilot study, requiring each task to be solved in at most one query resulted in participants constructing very complex queries that were over-constrained and yielded empty result lists. Also, scanning tasks may involve multiple actions of the same ISS to gather the required number of target documents. Furthermore, this was the only way to allow support mechanisms like the relevance histograms and the query history to be of any use.

Before a participant began working on the tasks, they were informed of the time limit of four minutes and the requirement to work quickly but to still find all target

⁵The first training task was in the ISS searching/specification and the second one in ISS scanning/recognition, so together they covered all ISS values and thus all features—see Appendix A.1.

7. Combining Optimal Support Mechanisms

documents. The bonus program, used as a motivator, was explained: the top six participants were awarded an additional sum of 10 euros. The participants were also informed that the bonus program did not require the participant's address data to be stored alongside the experimental data and that the problem of notifying the winners was solved differently.

Participants were also reminded to formulate English language queries and that they had some translation aids they could use.

The last question before the participants started working on the task was whether they agreed to be eyetracked. The participants could choose whether to be eyetracked or not because the eyetracking data was planned to be gathered for exploratory examination but was not required to study the research question. The general procedure was explained to them, as well as the amount and kind of data that was stored and not stored by the eyetracker. All participants agreed to be eyetracked, but for one participant the system could not be calibrated. In this case, the measurement part was completed without recording eyetracking data.

Post-experiment measures

After the participants completed the experimental tasks, they were asked to complete the NASA-TLX, the second SUS questionnaire, and a general questionnaire for quantitative and qualitative feedback. See Figures D.6 and D.2 in Appendix D for the questionnaires and Section 7.6.5 for a description of the NASA-TLX.

Kelly et al. [85] found that giving participants effectiveness feedback during an experiment influences post-experiment satisfaction scores. Because of this, no feedback was given to the participants, neither in the ACM baseline task nor in the training tasks or the experimental tasks.

The last questionnaire contained a shared secret used for the bonus program. The shared secret, a long randomly generated number, was handed over to the participant and the notification procedure explained.

Compensation

At the end of the experimental session, the compensation was given to the participant. There were two kinds of compensation: participants could choose between 15 euros or a certificate for taking part in the experiment, the latter needed by some students of a particular study.

7.6.7. Problems encountered

Despite extensive testing of the system and proactive quality assurance in the form of unit and integration tests, multiple software defects occurred during the experiment. In general, defects were fixed as they appeared. The data from sessions that were rendered unusable due to a defect were discarded and the session was repeated with a new participant. One participant was rejected during an experimental session, but before the tutorial section was finished, owing to language difficulties. See Appendix G for a detailed discussion of the problems and how they were handled.

7.7. Results

7.7.1. Preliminary data analysis

The preliminary analysis of the data showed that one participant in the integrated condition appeared to be extraordinarily successful, having an F_1 per 240 seconds of 18, which is about four times the amount of the second best participant. Careful analysis of the screen capture of the participant's session revealed that the participant learned how to exploit a software bug. The bug allowed to submit queries before the clock was started, resulting in very short task completion times being logged. Since the measurements were both inaccurate and outliers even to stricter standards⁶, the data of this participant was edited in two ways: The timings were updated from the time stamps in the screen capture. The participant also managed to spend 330 seconds on the last task, even though it was planned to take a maximum of 240 seconds. For

⁶The participant's success measure was more than 14 times the interquartile range (IQR) greater than the third quartile—the stricter of standards require a distance of only 2.5 times the IQR.

7. Combining Optimal Support Mechanisms

this reason, the documents selected during this task were chosen to include only those selected during the first 240 seconds.

7.7.2. The participants

The experiment had a between-groups design with 12 participants planned for each of the three groups defined by the three experimental conditions baseline, adaptive and integrated. Thus, 36 participants had to take part in the experiment.

Overall, 53 participants were recruited using the methods described in section 7.6.1.

Of these, 16 were dropped due to problems during the measurement session (see Section G.1) and one participant was rejected (see Section G.2).

The data of the remaining 36 participants was used in the analysis. These participants were between 19 and 38 years of age (median and mean at about 25), 18 were female and 18 were male. They stated to have had between six and 17 years of experience with search systems such as web search engines (mean: 11 years). Asked for their language proficiencies in both German and English on a scale from 1 (beginner) to 5 (expert), all stated to be in the bracket between 3 and 5. While the German proficiencies had a clear trend toward the expert level (median = 5), the English level tended to be lower (median = 4). Twelve participants were studying cognitive and media science, three each studying applied computer science, mechanical engineering, East Asian studies, and economics, and the others were enrolled in less frequent courses, like sociology and biology.

7.7.3. The SUS baseline task

To establish a baseline for the SUS questionnaire, participants had to conduct a search in the ACM Digital Library. The task was completed successfully without assistance by three participants out of 36. All three participants who completed the task used an advanced search stratagem: after finding documents from the assumed target author, they navigated to the details page of the author and searched there for the target document. All other participants failed to locate the target document within the allotted time frame of three minutes.

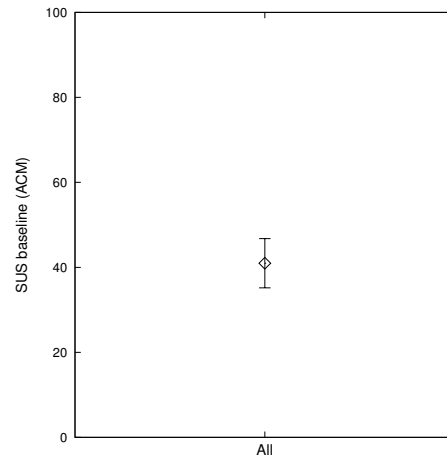


Figure 7.16.: SUS scores for all participants

The SUS questionnaire the participants were asked to complete after this task had a mean score of 41 and a range of scores of 10–75 (see Figure 7.16—again, this and the following graphs show the means and 95% confidence intervals of the respective data).

7.7.4. Main measure

The most interesting measures were the effectiveness and efficiency measures, since these were directly related to the hypotheses under scrutiny. As with the previous experiments detailed in chapter 5, the significance tests in this chapter were ANOVA tests with random permutations, unless stated otherwise.

Effectiveness measured by the F_1 score

The effectiveness of the participants was measured as the F_1 score of the session recall and precision, as described in more detail in section 7.6.5. The participants in the integrated condition achieved the highest effectiveness scores ($\bar{x} = 0.0257$) and were on average about twice as effective as those in the baseline condition (0.0128), who were the second most effective participants. The participants in the adaptive

7. Combining Optimal Support Mechanisms

condition (0.0072) were the least effective. Figure 7.17 shows graphs of the distribution of the effectiveness variable in each condition. The difference between the groups was statistically non-significant ($F(2, 33) = 1.023, p = 0.37$).

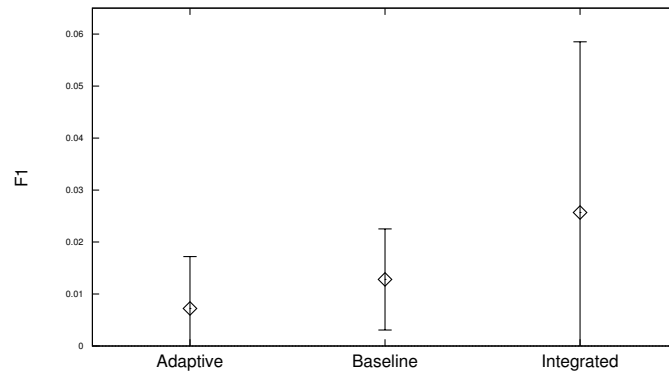


Figure 7.17.: F_1

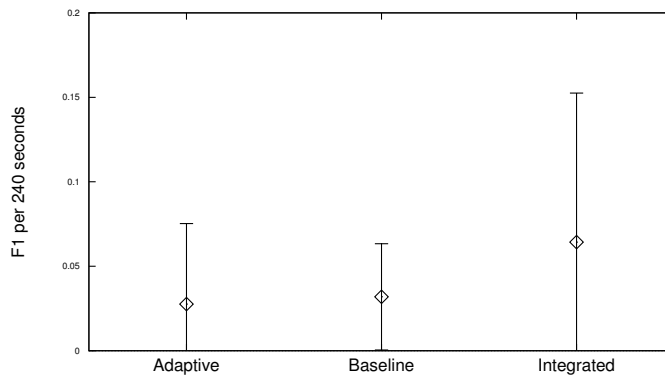
Efficiency as F_1 per time

Efficiency was calculated as F_1 score per time. To get somehow meaningful numbers, the efficiency metrics were normalized to 240 seconds, the time limit of the tasks. The efficiency metric showed a similar pattern as the plain effectiveness metric: On average, the participants in the integrated condition ($\bar{x} = 0.0643$) were about twice as efficient as those in the baseline condition (0.0319). Coming in last on average were again the participants in the adaptive condition (0.0276). Again, the difference between the groups was statistically not significant ($F(2, 33) = 0.5287, p = 0.59$).

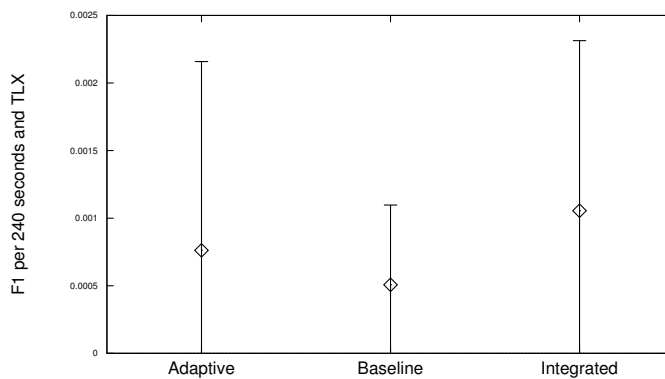
Efficiency as F_1 per time per TLX

An additional cost measure was the NASA-TLX score, which measures perceived stress on a self-reported scale (see Section 7.6.5).

Using this cost metric, an alternative efficiency metric was calculated as F_1 per time per TLX, treating stress as a cost that is expended to increase effectiveness. As with

Figure 7.18.: F_1 per time

the previous measures, the participants in the integrated condition were most efficient. Unlike the previous measures, however, using this metric, participants in the adaptive condition rank second before those in the baseline condition. However, again, the difference observed in the experiments is not significant ($F(2, 33) = 0.2818$, $p = 0.76$).

Figure 7.19.: F_1 per time per TLX

Concerning the main metrics, effectiveness and efficiency, no statistically significant differences could be observed. But other usability-related variables were measured which were possibly affected without also influencing effectiveness and efficiency. The following sections examine these additional measures exploratorily. Because this is only an exploratory examination of the data and most of the values are non-significant

7. Combining Optimal Support Mechanisms

anyway, the p-values in these sections were not corrected for alpha-inflation.

7.7.5. Did conditions differ for single ISSs?

While the global success variables do not vary between the experimental conditions, it was not clear whether this also holds for each individual information-seeking strategy and the tasks related to them. To study this aspect for each ISS, the success metrics of each of the three tasks related to the respective ISS were aggregated using the geometric mean. Statistically significant differences were found in two variables along with one borderline significant variable.

The borderline significant difference was found in the F_1 metric of the scanning/recognition tasks. This variable was increased from 0.04 in baseline to 0.14 in the adaptive condition ($p = 0.05$, see Figure 7.20).

The one significant difference was time on task for the searching/recognition tasks. This variable showed a reduction from 147 seconds in baseline to 111 seconds in the adaptive condition ($p = 0.02$, see Figure 7.21).

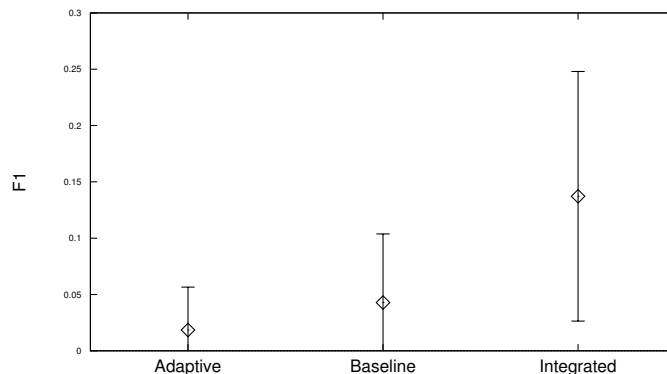


Figure 7.20.: Success for scanning/recognition

An interesting observation was the F_1 score for the searching/specification tasks, the other significant difference ($p = 0.03$): The effect observed was that the participants in the integrated condition were less than 50% as successful as those in the baseline

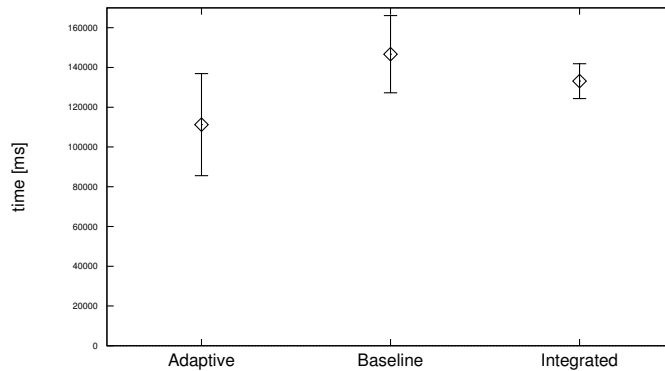


Figure 7.21.: Time on task for searching/recognition

condition: the F_1 metric was reduced from 0.82 in baseline to 0.34 in the integrated condition. Appendix H includes the data table. It can be seen that half of the participants achieved an average F_1 score of 1, which means that they managed to select exactly all target documents, and no other documents, in all tasks. Some participants, however, achieved much lower scores.

This negative effect was not expected and therefore more closely examined. The first assumption was that maybe a language problem resulted in the participants not being able to understand certain tasks. Of the 16 not so successful participants, nine reported a German proficiency level of 5 (the highest possible value); only one reported a proficiency level of 3 and the others a level of 4. So it seemed unlikely that communication problems were the cause. An examination of three participants' screen recordings showed that the problems were partially effects of the participants' personalities and partially effects of the system's and the tasks' shortcomings.

One problem was that the search syntax was logical, but strict: Searching using the query `Author="Edward Tufte"` did not find any document because the author fields of the Tufte books in the corpus read "Edward R. Tufte." The query, however, was formulated as searching for the phrase "Edward Tufte" in the author field, which was too narrow a search. A better formulation would have been `Author="Tufte, Edward"`. This is a fact that the participants were taught about in the tutorial section, but obviously this was too fiddly for some users.⁷

⁷Note that even some of the participants whose screen recordings were examined due to low scores in

7. Combining Optimal Support Mechanisms

Another issue arose from the fact that participants might see a document, still visible from the previous task, that at first glance seemed to be relevant to the current task but in fact was not. This clearly presented a problem with the tasks—tasks which should have been constructed in such a manner as to have different target documents, at least for immediately successive tasks.

Other users made the mistake to search too broadly: one participant used the query `Author="Long, John"`, even though the title of the book was also given in the task description. The result was a long list of result items that the participant could not handle in the allotted time. A different user found two of the three target documents for the Tufte search task and clicked the “Finish” button.

In conclusion, the difference of F_1 scores for the tasks in the ISS searching/specification was indeed likely to be a non-random result. However, closer inspection revealed that the difference was perhaps not caused by the systematic difference in the quality of the systems but by a difference between the participants of the experimental groups, even though their assignment to the conditions was random.

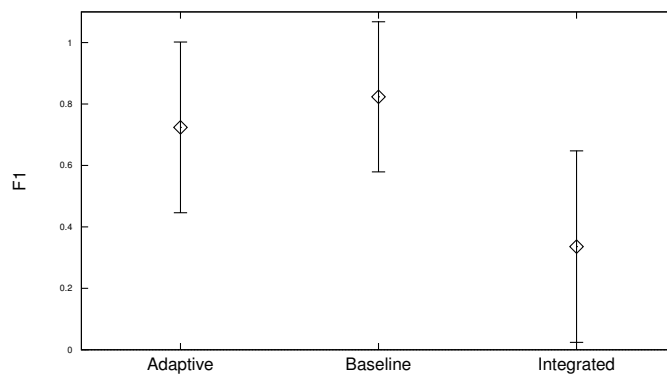


Figure 7.22.: Success for searching/specification

the searching/specification tasks did remember the better form of the author query, so the syntax was not impossible to use—just not very easy.

7.7.6. Did stress vary between groups?

One main measure was the efficiency calculated as F_1 (effectiveness) per time and stress (cost). This measure did not differ significantly between the groups. However, there was the possibility that the perceived stress changed between the groups, but not enough to counter the noise in the other two variables used to calculate the second efficiency metric, F_1 and time-on-task.

Figure 7.23 shows the distribution of the TLX variable between the three groups. The participants who reported the highest levels of stress on average were those in the baseline condition ($\bar{x} = 75$), followed by those in the integrated condition (63) and those in the adaptive condition (61). The difference between all groups is not statistically significant ($p = 0.11$) but the adaptive condition is at the periphery of significance ($p = 0.052$).

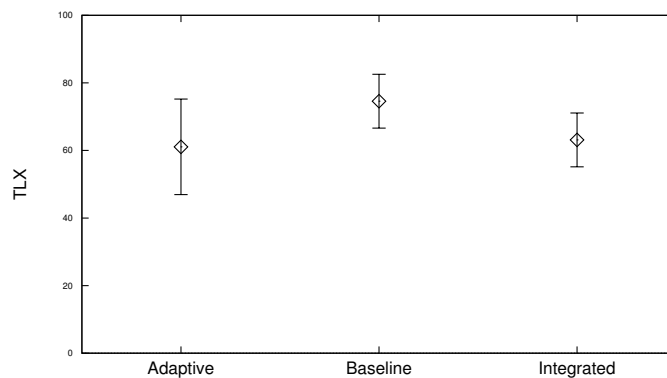


Figure 7.23.: Stress (by TLX)

7.7.7. Were there any differences in user satisfaction?

Another question was: if the performance-oriented variables did not show any difference, was there any difference between the groups concerning satisfaction with the system? As explained in Section 7.6.5, whether the SUS score has good-enough inter-participant reliability has not been ascertained. For this reason, two measurements have been taken: the baseline measurement, describing the satisfaction with the ACM

7. Combining Optimal Support Mechanisms

DL web site, and the treatment measurement, describing the satisfaction with the experimental system. The comparison with the ACM baseline is shown in Figure 7.24 (a positive difference means that the experimental system achieved a higher SUS score than the baseline). It can be seen that the SUS score differences seem to be in favor of those two experimental systems that afforded the users with search support features other than translations ($\bar{x} = +18,9$ for adaptive, $+21.9$ for integrated, $+17.8$ for baseline). However, the ANOVA test reports that these differences are quite probably due to chance ($p = 0.92$).

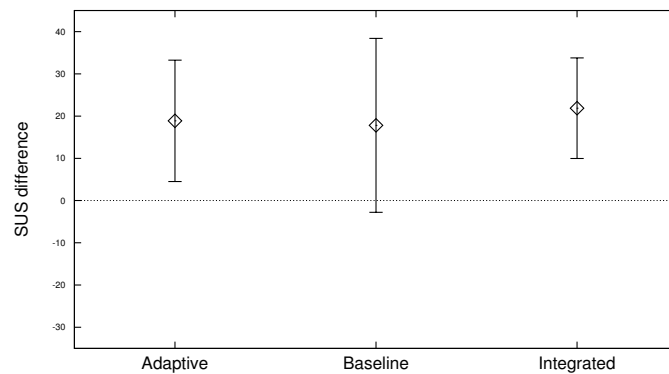


Figure 7.24.: Difference in user satisfaction (SUS) between ACM baseline and the experimental system

An assumption alternative to the one in the discussion above is that the SUS questionnaire is reliable enough to directly compare scores between participants. In this case, only the SUS scores measured after the experimental tasks have to be considered. Figure 7.25 shows the distribution of the SUS scores measured after the experiment in the three experimental conditions. The difference between these is 54.8 (baseline) to 60.1 (adaptive) to 66.0 (integrated). This difference is still not statistically significant ($p = 0.44$), according to an ANOVA test.

The difference of the SUS scores was significantly different between the ACM baseline and the experimental systems in the adaptive and integrated conditions according to paired two-sided t-tests⁸. Participants in the adaptive condition scored their ex-

⁸The two-sided variant was chosen because other observations made during the experiment did not warrant that the experimental system be rated better than the ACM baseline.

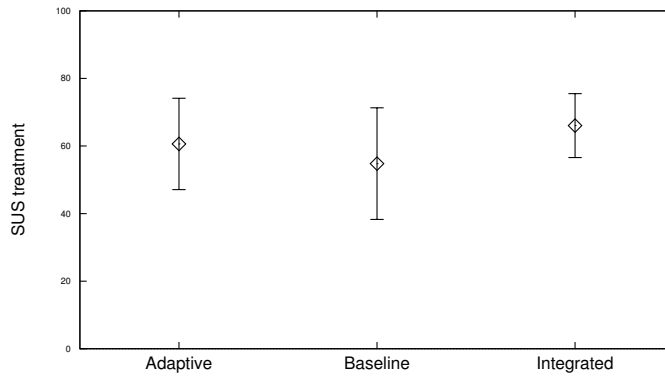


Figure 7.25.: User satisfaction (SUS) for the experimental system

perimental system on average 19 points higher than the ACM baseline ($p = 0.01$). Those participants tasked with the integrated system, rated their system, on average, 22 points higher than the baseline ($p = 0.002$). Globally, all participants rated their respective system on average 20 points higher on the SUS scale than the ACM DL ($p < 0.0001$). Only the experimental baseline was not significantly higher ($p = 0.08$, average difference 18 points).

7.7.8. Were the support features of any help?

With differences between the groups so low across virtually all metrics, the question arose whether the features intended to support the users did actually do so. To answer this question, four new success scores were calculated: one per ISS facet value examined in the experiment (scanning, searching, recognition, specification). Each new success variable was the geometric mean of the success metrics of all tasks in which the given facet value was involved. For example, to calculate the success metric for “scanning”, the success metrics for all tasks in the ISS scanning/recognition and for all tasks in the ISS scanning/specification were aggregated.

The only ISS facet value that came close to a statistically significant difference between the experimental conditions was the specification facet value. For this value, the difference in F_1 was bordering on significance ($p = 0.068$), but the difference was to

7. Combining Optimal Support Mechanisms

the disadvantage of both adaptive (-0.18) and integrated (-0.11) conditions, while the baseline users achieved an F_1 of 0.20.

7.7.9. Demographics and search

In interactive information retrieval studies, demographics are often described in terms of age, gender, course of study, search experience and other variables. One of the problems of gathering demographic data about the sample is that the more data is gathered, the easier it is to de-anonymize the data: there are probably not so many 42-year old female post-docs in the mechanical engineering department. On the one hand, data that could compromise the participants' privacy should be avoided. On the other hand, some statistical description of the participant sample is needed to allow readers to assess whether the findings in a study might also apply to another group of users. To find out which variables can be safely omitted, the influence of two variables on the search result was examined: gender and age.

Was there any difference between genders?

No difference in the main variables was observed between the genders: neither the effectiveness metric F_1 ($p = 0.88$) nor the efficiency metrics F_1 per time ($p = 0.65$) and F_1 per time per TLX ($p = 0.66$) showed statistically significant differences. See Figure 7.26 for graphs of the F_1 measure between genders.⁹

Was age a significant factor?

When plotting success against age, the regression line shows a slightly negative slope (see Figure 7.27) but the relationship is not statistically significant for any of the metrics F_1 , F_1 per time and, respectively, F_1 per time per TLX ($p = 0.68$, $p = 0.48$ and $p = 0.24$).

⁹In these graphs the most successful male participant was omitted because it was the most extreme outlier

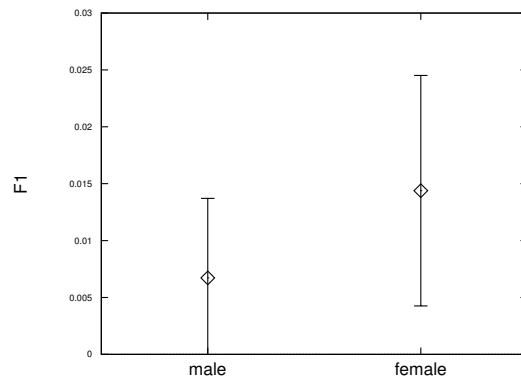


Figure 7.26.: Success by gender

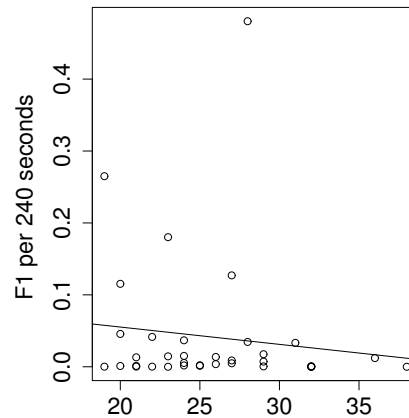
The above shown progression in the p -values indicates that the TLX measure might introduce some sort of signal into the data, giving the metric “success per time per TLX” a far smaller value than that of the same metric without TLX. However, the raw stress measure TLX shows a slight negative slope but it is not statistically significant ($p = 0.86$).

7.7.10. Does Heap’s law hold for the pool?

The question arose whether Heap’s law can also be applied to the number of documents in a pool, depending on the number of participants who contributed to the pool.

To examine this question, the set of documents each participant collected over all 16 tasks (including training tasks) was determined. The idea was to accumulate all participants’ documents into a pool to determine how many unique documents are in it after n participants. Then 1000 random permutations of the participants were drawn. For each permutation the pool size for the first n participants for all n between 1 and the number of all participants was determined and saved. In the last step, the pool sizes for each of the values of n was averaged and plotted (see Figure 7.28). The dots represent the actual mean number of documents determined for each number of participants and the line represents the fitted curve of the anticipated Heap’s law for constants $C = 31$ and $s = 0.6375$. The fit is quite close and shows that this is another situation in which Heap’s law applies.

7. Combining Optimal Support Mechanisms



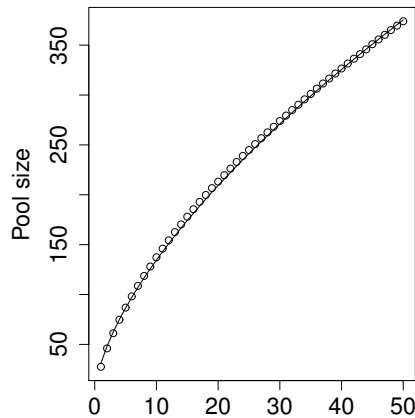


Figure 7.28.: Pool size by number of participants

so the standard deviation plays a part in calculating the effect size. This makes power a function of raw effect difference, its standard deviation, α and sample size. Each of these is a suspect in searching for a reason as to why no effect could be observed even though the assumption is that there is one. The chosen α of 0.05 complies with the tradition in user-oriented research, which is also frequently used in interactive information retrieval studies. Of course, the α could have been relaxed, but with $p = 0.37$ as the lowest p-value for the main metrics, this would have had to be a *very* relaxed value. While this part of the power calculation is difficult to debate, the other parts can be given more space for discussion.

The next suspect is the raw effect. Assuming it was too small, this could have been caused by multiple reasons. One possible reason is that the effect of each individual support feature was not large enough. This would not come as too much of a surprise since hard data on effect sizes of individual user interface mechanisms are rare. Even if individual mechanisms are studied (as in Tombros's and Sanderson's paper on query-biased summaries [120]), these studies are seldom replicated. If studies are not replicated, it is still possible that the findings have been due to chance—even for low p-values.

Another hypothetical reason for an effect too small to measure could have been that the effects of each individual mechanism were not additive; they might even have cancelled

7. Combining Optimal Support Mechanisms

each other out. Moreover, the tasks could have been designed in a way unsuitable for evoking the behavior that the user interface mechanisms were supposed to support (see Section 7.8.3).

The standard deviation is another suspected reason. In the data gathered in this experiment, the standard deviation of the sample data was higher than the mean in all three conditions (baseline $sd = 0.015$, $\bar{x} = 0.013$, adaptive $sd = 0.016$, $\bar{x} = 0.007$, integrated $sd = 0.051$, $\bar{x} = 0.026$). This was probably caused by the sample not being homogeneous enough. Due to very slow enrollment in the study, participants from a diverse range of backgrounds were accepted: students and unemployed, students from many different fields of study, some in their thirties, some barely graduated from secondary school. Due to the other constraints¹⁰ of the study design, this could not be accounted for by within-subject measurements.

The last suspect is the sample size. The central limit theorem posits that the larger the sample size the better the sample mean estimates the population mean. Maybe the estimate established in the experiment is off because the sample size was too small. In fact, while the standard deviation of the population is difficult to influence, the effect size is to be measured, and α is relatively fixed, the sample size is the only effective variable in the equation that describes statistical power. Assuming there is an effect that had been missed in the experiment, the sample size might have been too small. In the case of this study, the sample size was 12 participants per condition. It was dictated by time and monetary constraints, as well as enrolling speed. Other interactive IR studies with much smaller samples were more successful. For example, a study of Diriyeh et al. [37] involved 18 participants and very low p-values of 0.001 and lower but, admittedly, did not measure search success.

A last possible reason as to why no differences have been observed between the experimental conditions is that the wrong metrics were chosen. It might be that there are differences between the systems, but they do not influence effectiveness and efficiency, but other aspects such as learning or joy of use. However, the research question of this experiment was one of efficiency, and the chosen metrics reflect this aspect.

If exploratory searches also contain stratagems, and stratagems cannot be completely

¹⁰Other constraints were that the experimental sessions should not take too much time—90 minutes was already considered very long, that multiple tasks of each type had to be completed to smooth the data measured and that users had to be tutored about the interface to avoid also measuring learnability.

expressed by ISS classes, it follows that some activities that happen during an exploratory search cannot be expressed in Belkin's and Cool's ISS classification. The question that follows from this is, to what extent do the other activities in an exploratory search that cannot be expressed as an ISS play a part in the success of an exploratory search? And how important is it to support them? Is merely supporting all ISS classes really enough in order to support everything the user could want during a search?

7.8.2. Differences in the SUS scores

The SUS scores of the ACM baseline and the experimental systems were found to differ. This might be due to a systematic difference between these systems. This assumption is not entirely unfounded, since the systems actually differ in a variety of ways, including technology (web-site vs. desktop application), navigation, underlying search system, and surrogate representation. The difference observed might have also been caused by the experiment design: a repeated measures design, as used for this variable, always involves the risk that the participants try to find out what is being measured. Due to the social desirability bias, some participants are then trying to do what they think the experimenter would like to see, e. g. give favorable, but insincere, answers on a SUS questionnaire. For this reason, the differences observed have to be taken with a grain of salt.

Even if there were significant differences in any of the main metrics, this study is limited in other ways.

7.8.3. Tasks, ISSs, and participants

A problem was that each task was intended to provoke a specific ISS but users could decide to proceed in a different way. Each ISS class in this study was a combination of two facets: one facet describing the number and type of documents to search for, and the other describing how the user performed the task. The number of documents to search for—one or few known documents in the “searching” value of the facet and many unknown documents in the “scanning” value—was easy to control. However, the actions employed by the participants were not easy to control. Consider, for example,

7. Combining Optimal Support Mechanisms

the task of finding Tufte's books on information visualization. The task description mentioned Tufte's full name and the full titles of the three books. The participant could now decide that enough information was available to formulate a comprehensive query¹¹ to find all specified books. The participant could also assume that Tufte had not written a large number of books and intentionally under-specify the query¹² to sift through the result list, picking out the books mentioned in the task description. The first approach, emphasizing the query formulation, would be specification, while the latter approach, sifting through result lists, would be recognition. An additional problem was that participants could split each task into several subtasks, e. g. finding each of Tufte's books using a different query and even switch between specification and recognition while doing so.

One way to work around this problem, suggested by Käkki and Aula, is to provide the participant with initial queries to gently push them in the right direction [80]. This method was not applied because providing initial queries would have made it impossible to measure the effect of query specification support. Providing initial queries only in some of the tasks could have confused the participants and would have resulted in a less realistic scenario.

7.8.4. Limitations

Even experiments that basically find nothing can be limited by their study designs. One such limitation is that the tutorial sections were delivered in person by the experimenter instead of using standardized video tutorials. This decision was made due to resource constraints and because it was assumed that different participants have different learning speeds and prior knowledge. The consequence of delivering the tutorials in person is a possible expectation bias: the experimenter might tutor the baseline participants less enthusiastically than those of the other two conditions, leading to worse performance of the participants of the baseline condition. A counter argument is that other study designs would have had similar problems. Delivering the tutorial using a video recording of a tutor other than the experimenter would have allowed to double-blind the experiment, but it would have introduced a new problem: The original intention was not to analyze the question of how participants instructed by video

¹¹For example, `Author="Tufte,Edward" AND Title="Visual display ..." OR Title="Visual Explanations" OR Title="Envisioning ..."`

¹²For instance, `Author="Tufte,Edward"` or even `Title=visual`, hoping for proper stemming

would perform, but how participants who model experienced users would perform. How much each participant matched that model after the video tutorial would have depended on the participant and/or the quality of the video tutorial. Delivering the tutorial in person allowed to make sure that each participant entered the experimental section after they understood the tutorial content.

7.9. Conclusion

To examine the question if searchers are best supported by giving them support mechanisms for each of the search tasks they perform, an experiment with a between-groups design using three experimental conditions was conducted. A total of 36 participants completed three sets of tasks related to four ISS classes. The participants using the baseline interface only had a translation tool available to them, while the participants in the adaptive condition were supported by a search user interface that adapted the set of features to the task at hand. The participants in the integrated condition had all support features available at all times.

Despite extensive planning, the experiment described in this chapter was not able to show a statistically significant difference in the main measures for effectiveness and efficiency between the three experimental conditions.

Numerous other variables were examined, few of which were significantly different between the conditions. The participants in the integrated conditions seemed to suffer less from stress. User satisfaction was not significantly different, presumably because even the experimental baseline system was not really bad: The difference between the ACM baseline and the experimental baseline system was slightly in favor of the experimental system, and the ACM baseline is, after all, a production system widely used by academic searchers.

Since no correction for alpha-inflation was performed for any of the significance tests, even the few significant outcomes might be false positives. It seems difficult to show an improvement in either relevance-based effectiveness or efficiency measures in an experiment with few participants, if it is not so tightly controlled.

8. Conclusion

8.1. Summary

In the general case and from the user's point of view, search is a complex process that involves many variables, methods and aspects. If only a specific document is to be found, things are simple, particularly when the user has found the document before and knows what they are trying to find. If the goal of a search is less well-defined, things become complex. Not only might the user not know the terms needed to describe the information need, they might not even know her real information need in the first place. They might not know where to search for documents, and cannot predict which and how many documents they need for their work task.

Search tasks can get so complex that users even give up entirely, trying to work around their problem or resorting to other information sources. These users would benefit from search systems that support their searches as they progress. The problem, from the system designer's point of view, is that users in long search sessions tend to try many different search actions, each of which is a candidate for being supported by the search system. Should all of these possible search actions be supported in any way by the search system? If so, how?

The research question of this thesis was if users should be provided specific support features for the search action they are currently performing, or if it is enough to put enough features into a search system so that all potential actions are supported and then leave the burden of choice to the user. This research question contained two subordinate questions, the first of these is, Which search actions exist? and the second one, How are each of these actions properly supported?

The first question about the set of potential actions was answered by choosing the classification of information seeking strategies (ISS) by Cool, Marchetti, and Belkin,

8. Conclusion

a faceted classification that, by way of definition, covers all search actions that are conceivable.

The ISS classification uses four facets to describe each search action: method, goal, mode, and resource used. The latter facet involves some issues concerning its application to realistic scenarios. The second facet is hard to operationalize. This left two facets to classify every search action in the context of this study.

The two facets of the ISS classification, method and mode, are binary facets, leading to a total of four classes of search actions to support in a search user interface. The method facet has the two values “scanning” and “searching.” The value “scanning” describes an open-ended, vaguely defined search for an unknown set of documents, while the value “searching” refers to the act of looking for one or very few well-defined documents with a clear finish line. The mode facet has the two values “recognition” and “specification.” “Recognition” is the act of finding documents by visual stimulation, and is called “visual search” in the field of cognitive psychology: the searcher does not give a full specification of the target documents, for whatever reasons, but relies on finding them when they see them. “Specification” is the act of giving enough information about the target documents to help the search system narrow down the result set to reduce the time needed to visually search through the list as much as possible. The borderline between specification and recognition is blurred because even users in a search action involving recognition usually enter some search terms to reduce the result list they have to work through.

The second of the questions following from the research question was which support mechanism is needed for which class of search actions.

The IR literature can boast relatively few papers about studies examining single, well-defined search interface features in terms of relevance-based efficiency metrics. Yet some support features were identified in a literature search, mainly on the basis of hope instead of empirical evidence, due to a lack of the latter. Since the literature on visual search applied to interactive IR was particularly scant, two studies were conducted.

The first study examined two search result list design variants for text-based search each of which was supposed to be better than the traditional baseline design (i. e. showing title, snippet, and other meta-information fields, sometimes in a summarized

version). The support mechanisms compared in this experiment were the highlighting of surrogate parts and a table-based layout of the result list. Having found no real difference between these designs, a second study was conducted to compare the already examined designs with a baseline that did not have any support mechanism. This study did not find any significant difference, either.

The last study compared three search user interfaces with each other: a baseline system that had no support features beyond a basic translation tool; an adaptive system that offered support features specific to the task the participant was completing; and an integrated system that combined all support features of the adaptive system but offered access to all of them at the same time.

The questions in this experiment were: Which of these different systems allows users to search more effectively and/or more efficiently? Which one is less stressful? And which one is accepted the most by the users?

36 participants completed a tutorial section, four training tasks and 12 experimental tasks with one of the three systems (between-groups design). The time to perform the tasks was limited to four minutes and the documents rated as relevant to the task were logged by the system. The relevance ratings by the participants were compared with relevance assessments collected from external assessors using a specially crafted web site. The differences between the system, as observed in the experiment, tend toward the integrated system (see Section 7.7.4), but not significantly so. Nor did the stress reported by the participants, using a well-known standard test, and the satisfaction, measured using the SUS questionnaire, differ significantly.

8.2. Outlook

The main questions studied in this work could not be answered due to the lack of significant differences between the experimental conditions. Section 7.8.1 discussed possible reasons for this outcome.

The question now is: How should the experiment design be changed to actually see differences between the conditions?

8. Conclusion

A possible explanation for the fact that no differences could be observed is that the experimental conditions did not differ with respect to the variables measured in the experiment. This explanation is possible because neither the literature nor preparatory experiments offered reliable evidence that any of the support mechanisms could be expected to really support the user. If the building blocks used to assemble the systems were ineffective, the systems could not be expected to show any differences to another relatively ineffective system.

To overcome this, more experiments are needed that study the effect of single isolated support mechanisms on efficiency, effectiveness and satisfaction metrics. When effective support mechanisms are known, their combination can be compared with a baseline system without those mechanisms. It might still be that these systems do not show any differences in the experiments, but this would then be caused by the combination of the mechanisms. After that, other combinations could be tested, with the knowledge that the individual mechanisms are effective.

Some of the tasks should be improved in a follow-up study. In particular, the difference between searching and scanning tasks could be more noticeable. Asking the user to find as many documents as possible in the scanning tasks instead of a given (larger) number might help the support mechanisms for scanning to play their strengths.

An aspect of the experiment design resulting from time constraints was the tutorial section. To better fit to the assumed user model of an experienced user, the participants had to complete a tutorial section right before the measurement section in the same experimental session. This might have resulted in fatigue since the tutorial covered a wide range of features. At the same time, the tutorial, compressed to about 30 minutes, might have been too fast for the user to remember all details and be able to apply the material to the experimental tasks. A different method of getting the participants up to speed with the system would be either a longer and separate session, preferably on a different day. The tutorial timing could be slower, with more tasks and breaks, thus increasing the learning success. The downside would be increased costs from compensation payments and probably slower recruiting rates if participants took part in the tutorial but did not make it to the experimental session. A trade-off could be to run the tutorials on the same day as the experiment is conducted. Alternatively, the participants could install an experimental system on their own computer and practice in their own time, and be allowed to take part in the experiment only after passing a test.

8.2. Outlook

Last, but not least, the experiment could be repeated with a larger sample of a more homogeneous group of participants. As discussed in Section 7.8.1, the sample of participants used here was heterogeneous, so participants who were extreme to some respect were possibly overrepresented in the sample. Taking the possibility into account that the actual difference between the experimental conditions was small, a larger sample size might help detect the difference.

One of the few differences that were statistically significant was the level of stress between the adaptive and the baseline condition. This is an interesting finding because it affects the balance between possible advantages of adaptive interfaces and their costs. A future experiment could try to reproduce this finding. If the effect is systematic, there should be a study on what causes this increased stress and at what level of adaptivity it begins to show.

To sum up, the studies detailed in this thesis still leave open the research questions, but they offer leads doing the groundwork for future experiments.

A. Tasks

A.1. Set Training

1. *Searching/Specification* – Find a book whose title contains “file systems”, that was written by Moshe Bar and published by McGraw-Hill! ¹
2. *Scanning/Recognition* – Find books by female authors that cover file systems or in which, besides other things, file systems are covered. Books on paper-based filing systems are not relevant. Find 2 books! ²
3. *Scanning/Specification* – Some authors write “filesystem” instead of “file system”. Find 3 books of that kind. ³
4. *Searching/Recognition* – Find the one book on operating systems whose cover features an airport in bird’s eye view (the view is directed toward the ground, not into the sky). ⁴

A.2. Set A

1. *Scanning/Recognition* – You completed a climbing course at a climbing gym and now want to climb at real rock (not in ice!) without killing yourself. So, you are

¹German original: Finde ein Buch, dessen Titel „file systems“ enthält, das von Moshe Bar geschrieben wurde und bei McGraw-Hill erschien!

²German original: Finde Bücher von weiblichen (Co-)Autoren, die Dateisysteme („file systems“) behandeln oder in denen, neben anderem, auch Dateisysteme behandelt werden. Bücher über Papier-basierte Ablagesysteme sind nicht gesucht. Finde 2 Bücher!

³German original: Manche Autoren schreiben statt „file system“ auch „filesystem“, ... Finde 3 passende Bücher!

⁴German original: Finde das eine Buch über Betriebssysteme, auf dessen Cover ein Flughafen aus der Vogelperspektive abgebildet ist (die Perspektive ist also auf den Boden gerichtet, nicht in die Luft).

A. Tasks

searching for textbooks on climbing safety. You're not looking for first-person accounts, novels, and books that only marginally consider safety aspects. Find 3 books. ⁵

2. *Scanning/Specification* – You noticed that “climbing” has the synonym “mountaineering”. Also, for “safety” there are the related terms “accident”, “knot”, “anchor” and “belay”. Extend your search accordingly and find 3 (not necessarily new) books for the extended query. ⁶
3. *Searching/Recognition* – In a forum somebody recommended a climbing book by John Long that has a waterfall on the cover. Find it! ⁷
4. *Searching/Specification* – In a forum somebody recommended a book by John Long with the title “How to Rock Climb!”. Locate the 2003 edition. ⁸

A.3. Set B

1. *Searching/Recognition* – You'd like to recommend a book on “interface design” that you've read to a colleague. Unfortunately, you don't have the book at hand and cannot remember the exact title. It was a book by the publisher “O'Reilly” with the typical cover design (the illustration of an animal on a white background). The book had a female author and her first name was Jennifer, Jessica, or so—probably with a “J”, anyway. Find the book! ⁹

⁵German original: Du hast einen Kletterkurs in einer Kletterhalle gemacht und möchtest nun an echten Felsen (nicht im Eis!) klettern, ohne Dich direkt umzubringen. Dazu suchst Du Sachbücher über Sicherheit beim Klettern. Nicht gesucht sind Erlebnisberichte und Romane, sowie Bücher, in denen Sicherheitsaspekte nur am Rande vorkommen. Finde 3 Bücher!

⁶German original: Du hast gesehen, dass es für „climbing“ das Synonym „mountaineering“ gibt. Außerdem gibt es für „safety“ die verwandten Begriffe „accident“, „knot“, „anchor“ und „belay“. Erweitere Deine Suche entsprechend. Finde 3 (nicht unbedingt neue) Bücher zu dieser erweiterten Anfrage.

⁷German original: In einem Forum wurde Dir ein Buch von John Long über Klettern empfohlen, das einen Wasserfall auf dem Cover hat. Finde es!

⁸German original: In einem Forum wurde Dir ein Buch von John Long mit dem Titel „How to Rock Climb!“ empfohlen. Finde die Ausgabe von 2003!

⁹German original: Du möchtest einem Kollegen ein Buch über „Interface Design“ empfehlen, das Du gelesen hast. Leider hast Du das Buch nicht zur Hand und kannst Dich nicht an den exakten Titel erinnern. Es war ein Buch vom Verlag „O'Reilly“ mit dem typischen Cover (eine Tierzeichnung auf weißem Hintergrund). Das Buch hatte einen weiblichen Autor und ihr Vorname war Jennifer, Jessica, oder so - jedenfalls wohl mit „J“. Finde das Buch!

2. *Searching/Specification* – While compiling the list of recommendations, you remember the three-part series of books on information visualization by Edward Tufte: “Visual Display of Quantitative Information”, “Visual Explanations”, and “Envisioning Information”. Some of these are available in a new edition; you would like to have the newest one of each. Find all 3 books! ¹⁰
3. *Scanning/Recognition* – Now you are looking for other books on modern graphical user interface design (published after 2000). The books should be (programming-)language-independent and not application-specific, so no books on user interface design in Java or for games. Specific platforms (e. g. mobile) are okay. Find 3 books! ¹¹
4. *Scanning/Specification* – You are collecting reprints of classical cocktail books (before 1940). You know that many old cocktail books did not have the word “cocktail” in their title but used words like “mixed drinks”, “ice drinks” oder “bartender”. Reprints have a new introduction or are described as “reproduction” or “reprint”. Some works are available in multiple reprints but you want to find only one version per book. Find 10 different books! ¹²

A.4. Set C

1. *Searching/Specification* – In a TV show, you heard about the book “The History of Africa” (exact title) of Molefi Asante. The documentary was very interesting and you want to read the book. Find the book! ¹³

¹⁰German original: Beim Zusammenstellen der Empfehlungsliste fällt Dir noch die dreiteilige Buchreihe von Edward Tufte über Informationsvisualisierung ein: „Visual Display of Quantitative Information“, „Visual Explanations“ und „Envisioning Information“. Einige davon sind bereits in neuer Auflage erschienen, Du hättest gerne jeweils das neueste. Finde alle 3 Bücher!

¹¹German original: Du suchst nun nach anderen Büchern über modernes Graphical User Interface Design (neuer als 2000). Die Bücher sollten aber (Programmier-)Sprachen-unabhängig und nicht Anwendungs-spezifisch sein, also keine Bücher über User Interface Design in Java etc. oder User Interface Design für Spiele etc. Spezifische Plattformen (z.B. Mobil) sind okay. Finde 3 Bücher!

¹²German original: Du sammelst Neuauflagen klassischer Cocktailbücher (von vor 1940). Du weißt, dass viele alte Cocktailbücher nicht das Wort „Cocktail“ im Titel hatten, sondern von „mixed drinks“, „ice drinks“ oder „bartender“ sprachen. Neuauflagen sind z.B. dadurch zu erkennen, dass sie eine neue Einleitung bekommen oder als „reproduction“ oder „reprint“ beschrieben sind. Einige Titel sind in verschiedenen Reproduktionen erschienen, aber Du möchtest nur eine Version pro Buch. Finde 10 verschiedene Bücher.

¹³German original: Du hast im Fernsehen vom Buch „The History of Africa“ (exakter Titel) von Molefi Asante gehört. Die Reportage war sehr interessant und Du möchtest das Buch lesen. Finde das Buch!

A. Tasks

2. *Scanning/Recognition* – Your interest in history was inspired by the book “The History of Africa” and now you are searching recent history text books (not novels) on Carthage, the famous rival of the Roman Empire. The books don’t have to cover Carthage exclusively but can also cover other ancient countries in north Africa or the Mediterranean. Books on the Roman Empire and its conflict with Carthage are too peripheral. The treatise should be by a contemporary author, not by a Roman or ancient Greek historian. Find 3 books! ¹⁴
3. *Scanning/Specification* – Besides Carthage, in the book you also found the pre-colonial Africa interesting. Now you are interested in this pre-colonial era in the empires of Ghana, Mali and Songhai, or generally western Africa. As a result, you’d like to have books that cover ancient history as well as the middle ages and tell about at least three empires. Find 5 books! ¹⁵
4. *Searching/Recognition* – During your search for Carthage you came across an interesting novel taking place in an alternative Carthage. Unfortunately, you forgot to memorize the title. You do know that the book cover was a color-illustration of a warrior (not black-and-white and not abstract), that the title had a subtitle, AND that the book was published in 2 parts. Find the two parts, i. e. 2 books! ¹⁶

¹⁴German original: Dein Geschichtsinteresse wurde durch das Buch „The History of Africa“ geweckt und nun suchst Du aktuelle Geschichtsbücher (nicht historische Romane) über Karthago, den berühmten Rivalen des Römischen Reichs. Die Bücher müssen nicht nur über Karthago handeln, sondern können auch andere Länder des Altertums in Nordafrika oder dem Mittelmeerraum berühren. Bücher über das Römische Reich und seinen Konflikt mit Karthago sind aber zu randständig. Die Abhandlung sollte von einem modernen Autoren sein, nicht von einem römischen oder altgriechischen Geschichtsschreiber. Finde 3 Bücher!

¹⁵German original: Neben Karthago fandest Du in dem Buch auch das präkoloniale Westafrika interessant. Nun interessieren Dich Bücher über diese präkoloniale Epoche in den Reichen Ghana, Mali und Songhai oder allgemein in Westafrika. Als Ergebnis sollten Bücher herauskommen, die sowohl das Altertum als auch das Mittelalter abdecken und dabei jeweils mindestens drei Reiche behandeln. Finde 5 Bücher!

¹⁶German original: Während Deiner Suche nach Karthago hattest Du noch einen interessanten Roman gesehen, der in einem alternativen Karthago spielte. Leider hast Du vergessen, Dir den Titel zu merken. Du weißt aber noch, dass das Buchcover eine bunte Illustration eines Kriegers war, (nicht schwarz-weiss oder abstrakt), der Titel einen oder mehrere Untertitel hatte UND das Buch in 2 Teilen erschienen ist. Finde beide Teile, also 2 Bücher!

B. Handouts used in experiment 1

Teilnehmernummer

1. Ich bin Jahre alt

2. Mein Beruf ist

3. Bei Studenten: meine Studienrichtung ist

Informatik (Bachelor)
Informatik (Diplom)
Informatik (Master)
Komedia (Bachelor)
Komedia (Master)
Keine Angabe

4. Ich benutze Suchmaschinen bzw. Suchsysteme seit Jahren

5. Wie schätzt Du Deine Sucherfahrung ein?

	1	2	3	4	5	
Ich bin Anfänger						Ich bin Experte

Figure B.1.: Pre-experiment questionnaire

B. Handouts used in experiment 1

Teilnehmernummer

1. Ich fand die Aufgaben

	1	2	3	4	5	
langweilig						interessant
schwierig						einfach
ermüdend						nicht ermüdend
sehr gut verständlich						unverständlich

Figure B.2.: Post-experiment questionnaire

C. Handouts used in experiment 2

Vom Versuchsleiter auszufüllen

Teilnehmernummer

Vom Teilnehmenden auszufüllen

1. Mein Beruf ist

2. Bei Studenten: meine Studienrichtung ist

Informatik (Bachelor)

Informatik (Diplom)

Informatik (Master)

Komedia (Bachelor)

Komedia (Master)

Keine Angabe

3. Ich benutze Suchmaschinen bzw. Suchsysteme seit

Jahren

4. Wie schätzt Du Deine Sucherfahrung ein?

	1	2	3	4	5	
Ich bin Anfänger						Ich bin Experte

Figure C.1.: Pre-experiment questionnaire

C. Handouts used in experiment 2

Vom Versuchsleiter auszufüllen

Teilnehmernummer

Vom Teilnehmenden auszufüllen

1. Ich fand die Aufgaben

	1	2	3	4	5	
langweilig						interessant
schwierig						einfach
ermüdend						nicht ermüdend
sehr gut verständlich						unverständlich

2. Ich glaube, ich habe ...

	1	2	3	4	5	
alle Suchziele gefunden						keine Suchziele gefunden

3. Ich glaube, an Suchzielen habe ich ... gefunden


	1	2	3	4	5	
überdurchschnittlich viele						unterdurchschnittlich viele

4. Ich glaube, ich habe ... gearbeitet

	1	2	3	4	5	
überdurchschnittlich schnell						unterdurchschnittlich schnell

Figure C.2.: Post-experiment questionnaire

D. Handouts used in the final experiment



UNIVERSITÄT
DUISBURG
ESSEN

Universität Duisburg-Essen
Fakultät für Ingenieurwissenschaften
Abteilung Informatik und angewandte
Kognitionswissenschaft
Fachgebiet Informationssysteme

Fragebogen zur Usability

Benutzbarkeit der Software (Usability of the Software)

	Starke Ablehnung			Starke Zustimmung	
	1	2	3	4	5
Ich denke, ich würde dieses System häufig benutzen wollen. <i>I think that I would like to use this system frequently.</i>					
Ich fand das System unnötig komplex. <i>I found the system unnecessarily complex.</i>					
Ich fand, das System war einfach zu benutzen. <i>I thought the system was easy to use.</i>					
Ich denke, dass ich die Hilfe eines Technikers brauchen würde, um dieses System benutzen zu können. <i>I think that I would need the support of a technical person to be able to use this system.</i>					
Ich finde die unterschiedlichen Funktionen des Systems sinnvoll integriert. <i>I found the various functions in this system were well integrated</i>					
Ich denke, das System enthielt zu viele Inkonsistenzen. <i>I thought there was too much inconsistency in this system.</i>					
Ich glaube, dass die meisten Leute die Verwendung des Systems schnell lernen könnten. <i>I would imagine that most people would learn to use this system very quickly.</i>					
Ich fand, dass das System sehr umständlich zu benutzen ist. <i>I found the system very cumbersome to use.</i>					
Ich fühle mich sicher im Umgang mit dem System. <i>I felt very confident using the system.</i>					
Ich musste einiges lernen, bevor ich das System wirklich nutzen konnte. <i>I needed to learn a lot of things before I could get going with this system</i>					

Figure D.2.: SUS questionnaire

<code>term</code>	Ein Term, irgendwo im Text
<code>a couple of terms</code>	Mehrere Terme (AND)
<code>"phrase search"</code>	Phrasen: diese Wörter in dieser Reihenfolge
<code>AND, OR, NOT</code>	logische Verknüpfungen
<code>(term OR word) AND NOT summer</code>	Klammerung
<code>Felder: Author, Title, Year, Text</code>	
<code>Title=term</code>	Feld-bezogene Terme
<code>Title="a title phrase"</code>	Feld-bezogene Phrasen
<code>Author="Nachname, Vorname"</code>	Autoren-Namen
<code>Title=test OR Title=text</code>	kompilziert
<code>Title={test OR text}</code>	genau wie <code>Title=test OR Title=text</code>

John Doe some terms from the text
 Author="Doe, John" AND (Title=test OR Title=text)
 Author="Doe, John" AND Title={test OR text}

1

Figure D.3.: Query language cheat sheet

D. Handouts used in the final experiment

Teilnehmernummer

Vom Teilnehmenden auszufüllen

1. Statistische Daten

Alter: Jahre

Geschlecht:

männlich

weiblich

keine Angabe

siehe unten

2. Mein Beruf ist

3. Bei Studenten: meine Studienrichtung ist

Informatik (Bachelor)

Informatik (Master)

Komedia (Bachelor)

Komedia (Master)

siehe unten

Keine Angabe

4. Ich benutze Suchmaschinen bzw. Suchsysteme seit Jahren

5. Meine Sucherfahrung würde ich so beschreiben:

	1	2	3	4	5	
Ich bin Anfänger						Ich bin Experte

(Bitte umblättern)

Figure D.4.: Pre-experiment questionnaire

6. Meine Deutschkenntnisse sind:

	1	2	3	4	5	
Ich bin Anfänger						Ich bin Experte (z.B. Muttersprachler)

7. Meine Englischkenntnisse sind:

	1	2	3	4	5	
Ich bin Anfänger						Ich bin Experte (z.B. Muttersprachler)

Figure D.5.: Pre-experiment questionnaire, page 2

D. Handouts used in the final experiment

Teilnehmernummer

Vom Teilnehmenden auszufüllen

1. Ich fand die Aufgaben

	1	2	3	4	5	
langweilig						interessant
schwierig						einfach
ermüdend						nicht ermüdend
sehr gut verständlich						unverständlich

2. Ich glaube, ich habe die Aufgaben ...

	1	2	3	4	5	
sehr gut gelöst						sehr schlecht gelöst

3. Wenn ich Literatur für die Uni suche (Hausarbeit, Abschlussarbeit), benutze ich ...:

4. Um meine Literatur zu verwalten (Papers, bibliographische Angaben, Zitate), benutze ich ...:

5. Sonstige Dinge, die mir aufgefallen sind:

Figure D.6.: Post-experiment questionnaire

E. Original screenshots from the final experiment

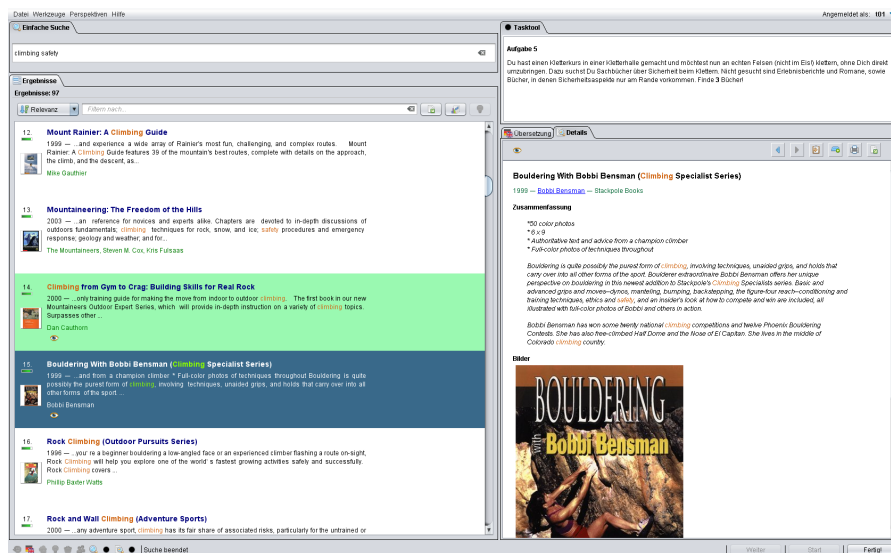


Figure E.1.: The baseline interface (German original)

E. Original screenshots from the final experiment

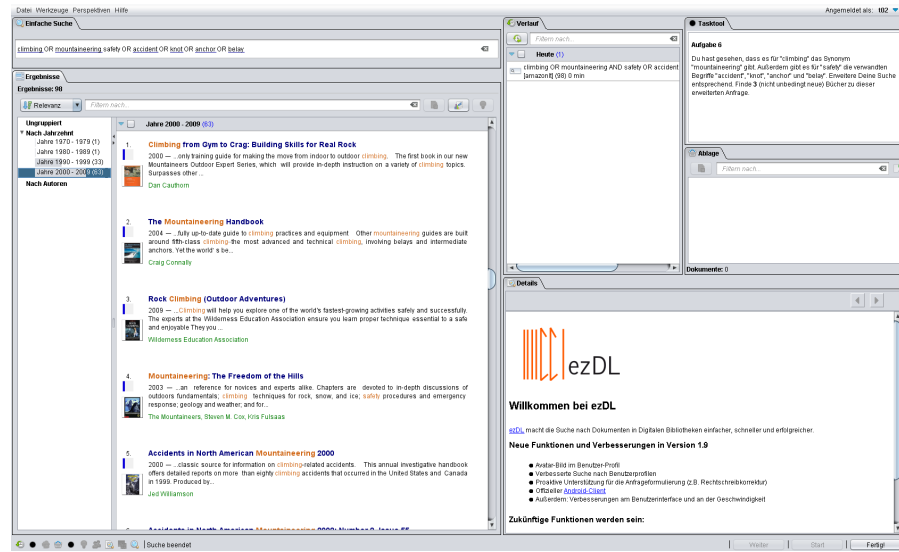


Figure E.2.: The adaptive interface for scanning/specification (German original)

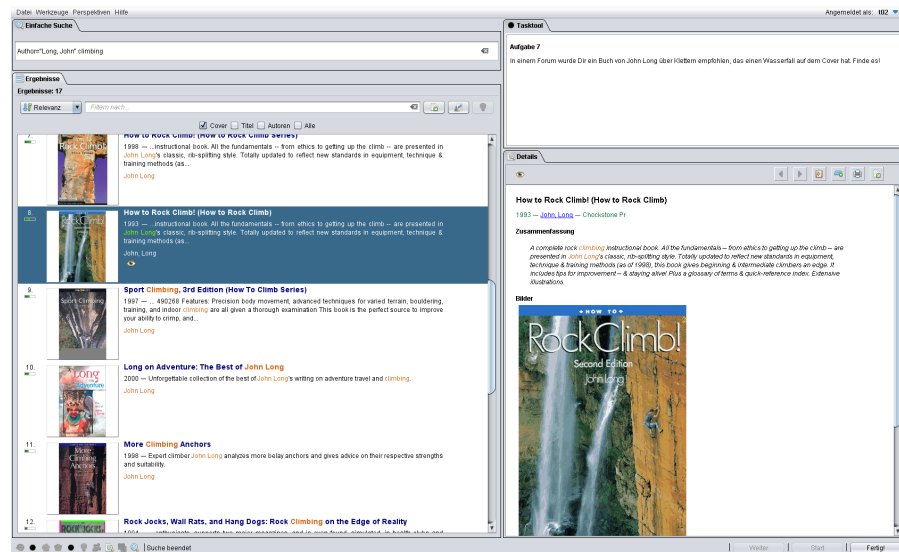


Figure E.3.: The adaptive interface for searching/recognition (German original)

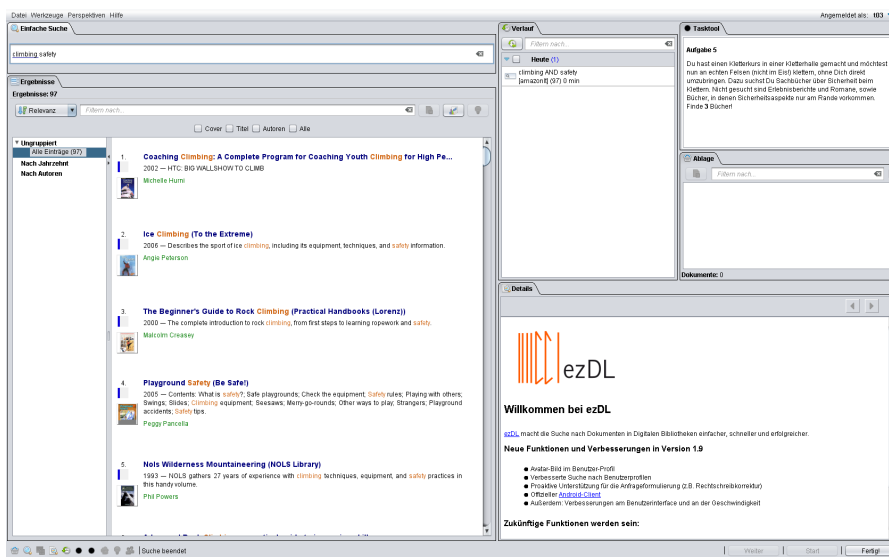


Figure E.4.: The integrated interface (German original)

F. The custom thesaurus

In the final experiment, a custom thesaurus was used for query expansion. The following shows the list of terms (bold) and their synonyms (prefixed with “Syn”) and related terms (prefixed with “Rel”). The last term, “Fenster” was used in the tutorial section.

cocktail *Syn:* mixed drink, *Syn:* iced drink, *Rel:* bartender

reproduction *Syn:* reprint

climbing *Syn:* mountaineering, *Rel:* belay

belay *Rel:* knot, *Rel:* accident, *Rel:* anchor, *Rel:* harness, *Rel:* carabiner

ancient *Rel:* medieval

carthage *Rel:* phoenicians, *Rel:* mediterranean

Fenster *Syn:* Luke, *Rel:* Glas, *Rel:* Fensterbank

G. Problems encountered in the final experiment

G.1. Client replacements and bug fixes

Despite extensive testing of the system and proactive quality assurance in the form of unit and integration tests, multiple software defects occurred during the experiment. In general, defects were fixed as they appeared. The data of sessions that were rendered unusable because of a defect were discarded and the session repeated with a new participant.

Two types of bugs were fixed during the course of the experiment: a) bugs that negatively affected support features and b) bugs that rendered experimental sessions unusable (e. g. a backend crash, that was leading to wrong timings and would have had the consequence of SUS- and TLX scores being badly influenced by the crash)

The rationale behind the decision to fix type a) bugs was that the experiment was not about testing a real system (in which case bug fixing would be detrimental to the experiment's objective) but the question, whether a theoretical, ideal system can support the user better. Of course, an ideal system wouldn't have bugs. Bugs of this kind usually affected adaptive and integrated conditions, resulting in a misalignment with the baseline condition, leading to bad validity of the data. For this reason, fixing the bugs was imperative. Bugs that affected either adaptive or integrated conditions were fixed in order to get a more valid differentiation between these two conditions.

Type b) bugs were fixed in order to increase the number of participants whose data could be used for answering the research questions. Bugs of this kind were fixed for all conditions, keeping the *ceteris paribus* criterion intact. The unusable sessions were repeated with new participants.

G. Problems encountered in the final experiment

In the course of the experiment, multiple software defects were found that made it necessary to abort the respective running session, fix the defect and rerun the session with a new participant. This was the reason to drop the data of 16 participants.

Three logins were removed at the very start of the experiment due to multiple software errors.

The wording of one particular task description was found to be misleading. This led to dropping the data of one participant.

One participant was dropped because of a bug that led to the relevance marker in the detail view not showing up under certain conditions. Another participant was dropped since the software change that was supposed to fix the bug did not actually do so and another change had to be applied.

One bug appeared twice and led to searches not being served anymore. The service that was responsible for interfacing with the Solr server containing the index sporadically ceased being able to resolve the Solr server's domain name for reasons yet unknown. The only remedy for this situation was restarting the search wrapper service. Since the occurrence of the bug was diagnosed only after a few search queries did not return results and to find the underlying issue, experimental sessions with this bug were aborted, the participant paid and the data recorded up to this point dropped from the experiment.

Another bug that appeared in the experiment was a failure to highlight search query terms that begin with a plus sign (“+”). In this case, a regex pattern compiler failed to compile a pattern, which led to the client not working anymore. The experimental session was aborted, the participant paid and the client was fixed. Subsequent participants worked with a new, fixed client.

One session failed because the search agent stopped working due to a memory leak.

After measuring a full task rotation (three participants) in each of the three conditions, it was noticed that the surrogate highlighting of one task (the UI task in the searching/recognition ISS) highlighted the title field instead of the author field. This was fixed in revision `experiment_final_6` for the remaining 9 participants in each condition. The rationale behind this decision was that the support features obviously didn't support the task in the right way and that the probability, that participants did not notice this

G.1. Client replacements and bug fixes

but relied on the automatic selection, was greater than 0. By highlighting the right fields the measurement of this task should be more accurate. So the first users, using the wrong highlighting, were basically instances of an imperfect user model, while those later participants complied more with the perfect user model. For this reason, and for reasons of cost for repeating nine sessions, the data for these sessions was kept.

Way too late into the experiment it was noticed that participants in the adaptive condition did not have any support for translation available in the recognition tasks.

In the baseline condition, translations were always available using the translation tool. In the integrated condition, translations were offered by the proactive modules, that were available in all tasks. In the adaptive condition, though, the initial design assumed that users would not need specification support in recognition tasks. For this reason, the proactive modules were turned off in these tasks. The problem was that translations, too, were afforded by the proactive modules and even in recognition tasks, the users had to translate the German tasks into English queries. So while baseline and integrated users got translation always, adaptive users got them only for some tasks but were in dire need of them in others. This was particularly true for the scanning/recognition tasks in the climbing set and in the history set, since these were tasks where the user could not have gotten a translation of the critical terms in an earlier task.

The searching/recognition tasks in the UI design set and in the history set did not have the problem to this extent, despite being recognition tasks: In the UI design task description the critical terms were given as needed and in the history task, the critical term could be translated in the preceding task.

The solution to the problem was that in the adaptive version, for the duration of the recognition tasks, only the translation module was enabled, achieving parity with the other conditions.

The session in which the problem was noticed, was dropped and the problem fixed so the following sessions could be ran properly. Because the problem affected only 2 out of 12 tasks in the sessions of 6 participants, the issue was estimated to be of rather low severeness. Since the recruiting was slow by that time (lectures were off) it was decided to keep on measuring following the usual scheme with the fixed software version until all required participants were measured and, if additional participants were available,

G. Problems encountered in the final experiment

to repeat the affected sessions, replacing the entire data sets to make sure that actual search experience and self-reported measures (SUS, TLX) were aligned.

G.2. Rejected participants

The session of one participant was aborted during the tutorial section because the participant was Chinese, communication was incredibly difficult due to the language barrier and the participant indicated only intermediate level of German proficiency in the demographic data questionnaire, making it likely that task description would not have been understood correctly.

H. Strange results

In the following table, the column “condp” is the condition the participant was assigned to: “b” is baseline, “a” is adaptive, and “i” is integrated. The column “e_SeSp_Succ” lists the success variable.

```
> d[c("condp", "e_SeSp_Succ")]
condp e_SeSp_Succ
  b 1.000000e+00
  i 1.000000e+00
  b 1.000000e+00
  i 1.000000e-06
  b 1.000000e+00
  i 1.000000e+00
  b 1.000000e+00
  i 1.000000e-02
  b 1.000000e+00
  i 1.000000e-02
  b 1.000000e+00
  i 1.000000e-04
  b 1.000000e+00
  a 1.000000e-02
  i 1.000000e+00
  b 8.735808e-05
  a 1.000000e-02
  i 1.000000e-04
  b 7.937011e-03
  a 8.735808e-01
```

H. Strange results

i 1.000000e-06
b 1.000000e+00
a 1.000000e+00
i 7.368068e-05
b 8.735808e-01
a 7.937011e-01
i 1.000000e-02
i 1.000000e+00
b 1.000000e+00
a 1.000000e+00
a 1.000000e+00
a 1.000000e+00
a 1.000000e+00
a 1.000000e+00
a 1.000000e+00
a 1.000000e+00
a 1.000000e-04

List of Figures

1.1. The search form on <code>www.amazon.com</code>	2
1.2. The results on <code>www.amazon.com</code>	3
3.1. The mSpace browser	29
3.2. Flamenco interface later on in a search [146]	30
3.3. HyperGrid with a detached browser view [77]	31
3.4. MedioVis main screen [58]	32
5.1. Screenshot of the table variant	48
5.2. Screenshot of the list variant	49
5.3. Screenshot of the ezDL system used in the experiment	50
5.4. True positives per 120s in both open and closed tasks	56
5.5. True positives per 120s in closed tasks	57
5.6. True positives per 120s in open tasks	58
5.7. Correlation between ability to concentrate and true positives found per time	60
5.8. Success metrics for different concentration scores (KLSTD) for list and table – grouping by median	61
5.9. Difference between false negatives in closed and open tasks	63
5.10. Screenshot of the baseline variant	64
5.11. Screenshot of the list variant	65
5.12. True positives per 120s	67
6.1. Simple query form, complex language: <code>ieeexplore.ieee.org</code> Command Search	73
6.2. Complex query form, simple language: <code>ieeexplore.ieee.org</code> Advanced Keyword Search	74
6.3. DAFFODIL’s proactive suggestions	86

List of Figures

6.4. List variants for proactive suggestions	87
7.1. The berrypicking tray	93
7.2. Relevance markers	95
7.3. Document markers in the result list	96
7.4. Three-state relevance button	96
7.5. The query history tool	97
7.6. The faceted result list	98
7.7. The checkboxes controlling the highlighting	99
7.8. A result item with highlighted title	99
7.9. A result item with highlighted authors	99
7.10. A result item with enlarged cover image	100
7.11. Proactive suggestions	100
7.12. The baseline interface	102
7.13. The adaptive interface for scanning/specification	103
7.14. The adaptive interface for searching/recognition	104
7.15. The integrated interface	105
7.16. SUS scores for all participants	121
7.17. F_1	122
7.18. F_1 per time	123
7.19. F_1 per time per TLX	123
7.20. Success for scanning/recognition	124
7.21. Time on task for searching/recognition	125
7.22. Success for searching/specification	126
7.23. Stress (by TLX)	127
7.24. Difference in user satisfaction (SUS) between ACM baseline and the experimental system	128
7.25. User satisfaction (SUS) for the experimental system	129
7.26. Success by gender	131
7.27. Success by age	132
7.28. Pool size by number of participants	133
B.1. Pre-experiment questionnaire	149
B.2. Post-experiment questionnaire	150
C.1. Pre-experiment questionnaire	151
C.2. Post-experiment questionnaire	152

List of Figures

D.1. TLX rating sheet 153
D.2. SUS questionnaire 154
D.3. Query language cheat sheet 155
D.4. Pre-experiment questionnaire 156
D.5. Pre-experiment questionnaire, page 2 157
D.6. Post-experiment questionnaire 158

E.1. The baseline interface (German original) 159
E.2. The adaptive interface for scanning/specification (German original) . . . 160
E.3. The adaptive interface for searching/recognition (German original) . . . 160
E.4. The integrated interface (German original) 161

Bibliography

- [1] Phillip L. Ackerman and Ruth Kanfer. Test length and cognitive fatigue: An empirical examination of effects on performance and test-taker reactions. *Journal of Experimental Psychology: Applied*, 15(2):163–181, 2009.
- [2] Jae-wook Ahn and Peter Brusilovsky. Adaptive visualization for exploratory information retrieval. *Information Processing & Management*, 49(5):1139–1164, 2013.
- [3] Leif Azzopardi, Diane Kelly, and Kathy Brennan. How query cost affects search behavior. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '13*, pages 23–32, New York, NY, USA, 2013. ACM.
- [4] Leif Azzopardi, Wim Vanderbauwhede, and Hideo Joho. Search system requirements of patent analysts. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, SIGIR '10*, pages 775–776, New York, NY, USA, 2010. ACM.
- [5] Aaron Bangor, Philip Kortum, and James Miller. Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of Usability Studies*, 4(3):114–123, May 2009.
- [6] M. J. Bates. Idea tactics. *Journal of the American Society for Information Science*, 30(5):280–289, 1979.
- [7] M. J. Bates. Information search tactics. *Journal of the American Society for Information Science*, 30(4):205–214, 1979.
- [8] M. J. Bates. The design of browsing and berrypicking techniques for the online search interface. *Online Review*, 13(5):407–424, 1989. <http://www.gseis.ucla.edu/faculty/bates/berrypicking.html>.

Bibliography

- [9] Marcia J. Bates. Where should the person stop and the information search interface start? *Information Processing and Management*, 26(5):575–591, 1990.
- [10] Melissa R. Beck, Maura C. Lohrenz, and J. Gregory Trafton. Measuring search efficiency in complex visual search tasks: Global and local clutter. *Journal of Experimental Psychology: Applied*, 16(3):238–250, 2010.
- [11] Thomas Beckers, Tina Bannert, Sebastian Dungs, Matthias Jordan, Noel Kamda, Sascha Kriewel, and Andreas Tacke. Report on results of the WP3 second evaluation phase. Technical Report D3.7, KHRESMOI, July 2014.
- [12] Thomas Beckers, Sebastian Dungs, Norbert Fuhr, Lorraine Goeuriot, Jessica Ignalski, Matthias Jordan, Liadh Kelly, and Sascha Kriewel. Report on results of the WP3 first evaluation phase. Technical Report D3.2, KHRESMOI, August 2012.
- [13] Thomas Beckers, Sebastian Dungs, Norbert Fuhr, Matthias Jordan, Georgios Kontokotsios, Sascha Kriewel, Yiannis Paraskeuopoulos, and Michail Salampasis. *ezDL: An Interactive IR Framework, Search Tool, and Evaluation System*, pages 118–146. Springer, 2014.
- [14] Thomas Beckers, Sebastian Dungs, Norbert Fuhr, Matthias Jordan, and Sascha Kriewel. ezDL: An interactive search and evaluation system. In *Proceedings of the SIGIR 2012 Workshop on Open Source Information Retrieval*, pages 9–16, Dunedin, New Zealand, August 2012. Department of Computer Science, University of Otago.
- [15] N. J. Belkin. Anomalous states of knowledge as a basis for information retrieval. *Canadian Journal of Information Science*, 5:133–143, May 1980.
- [16] Nicholas J. Belkin. Interaction with texts: Information retrieval as information seeking behavior. In G. Knorz, J. Krause, and C. Womser-Hacker, editors, *Information Retrieval '93. Von der Modellierung zur Anwendung. Proc. d. 1. Tagung Information Retrieval*, pages 55–66, Konstanz, 1993.
- [17] Nicholas J. Belkin, Colleen Cool, Adelheit Stein, and Ulrich Thiel. Cases, scripts, and information-seeking strategies: On the design of interactive information retrieval systems. *Expert Systems with Applications*, 9(3):379–395, 1995. <http://www.scils.rutgers.edu/~belkin/articles/eswa.pdf>.

- [18] Nicholas J. Belkin, P.G. Marchetti, and Colleen Cool. BRAQUE: Design of an interface to support user interaction in information retrieval. *Information Processing and Management*, 29(3):325–344, 1993.
- [19] N.J Belkin. Intelligent information retrieval: Whose intelligence? In Jürgen Krause, Matthias Herfurth, and Jutta Marx, editors, *Herausforderungen an die Informationswirtschaft. Informationsverdichtung, Informationsbewertung und Datenvisualisierung. Proceedings des 5. Internationalen Symposiums für Informationswissenschaft (ISI '96)*, volume 27 of *Schriften zur Informationswissenschaft*, pages 25–31, Konstanz, 1996. Universitätsverlag Konstanz.
- [20] N.J Belkin, R.N. Oddy, and H.M. Brooks. Ask for information retrieval: Part i. background and theory. *The Journal of Documentation*, 38(2):pp. 61–71, 1982.
- [21] Patrice Bellot, Antoine Doucet, Shlomo Geva, Sairam Gurajada, Jaap Kamps, Gabriella Kazai, Marijn Koolen, Arunav Mishra, Véronique Moriceau, Josiane Mothe, Michael Preminger, Eric SanJuan, Ralf Schenkel, Xavier Tannier, Martin Theobald, Matthew Trappett, and Qiuyue Wang. Overview of INEX 2013. In Pamela Forner, Henning Müller, Roberto Paredes, Paolo Rosso, and Benno Stein, editors, *Information Access Evaluation. Multilinguality, Multimodality, and Visualization*, volume 8138 of *Lecture Notes in Computer Science*, pages 269–281. Springer Berlin Heidelberg, 2013.
- [22] Ulises Cerviño Beresi, Dawei Kim, Yunhyong Song, and Ian Ruthven. Why did you pick that? Visualising relevance criteria in exploratory search. *International Journal on Digital Libraries*, 2010(11):59–74, 2010.
- [23] Krishna Bharat. Searchpad: explicit capture of search context to support web search. *Comput. Netw.*, 33(1-6):493–501, June 2000.
- [24] Pia Borlund. The IIR evaluation model: a framework for evaluation of interactive information retrieval systems. *Information Research: an international electronic journal*, 8(3):1–38, April 2003. <http://informationr.net/ir/8-3/paper152.html>.
- [25] Pia Borlund and Jesper W. Schneider. Reconsideration of the simulated work task situation: a context instrument for evaluation of information retrieval interaction. In *Proceedings of the third symposium on Information interaction in context, IiX '10*, pages 155–164, New York, NY, USA, 2010. ACM.

Bibliography

- [26] Rolf Brickenkamp, Lothar Schmidt-Atzert, and Detlev Liepmann. *Test d2-R*. Hogrefe, Göttingen, 2010.
- [27] Andrei Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, September 2002.
- [28] Marc Bron, Jasmijn van Gorp, Frank Nack, Maarten de Rijke, Andrei Vishneuski, and Sonja de Leeuw. A subjunctive exploratory search interface to support media studies researchers. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '12, pages 425–434, New York, NY, USA, 2012. ACM.
- [29] John Brooke. Sus — a quick and dirty usability scale. Technical report, Redhatch Consulting Ltd., 1991.
- [30] David Carmel, Naama Zwerdling, and Sivan Yogev. Entity oriented search and exploration for cultural heritage collections: The EU cultura project. In *Proceedings of the 21st International Conference Companion on World Wide Web, WWW '12 Companion*, pages 227–230, New York, NY, USA, 2012. ACM.
- [31] Jacob Cohen. *Statistical power analysis for the behavioral sciences*. Lawrence Erlbaum Associates, 2nd edition edition, 1988.
- [32] Colleen Cool and Nicholas J. Belkin. A classification of interactions with information. In H. Bruce, R. Fidel, P. Ingwersen, and P. Vakkari, editors, *Emerging frameworks and methods. Proceedings of the Fourth International Conference on Conceptions of Library and Information Science (COLIS4)*, pages 1–15, Greenwood Village, 2002. Libraries Unlimited.
- [33] Silviu Cucerzan and Eric Brill. Spelling correction as an iterative process that exploits the collective knowledge of web users. In *Proceedings Of Conference On Empirical Methods In Natural Language Processing, EMNLP '04*, pages 293–300, 2004.
- [34] Edward Cutrell and Susan T. Dumais. Exploring personal information. *Communications of the ACM*, 49(4):50–51, 2006.
- [35] Hercules Dalianis. Evaluating a spelling support in a search engine. In Birger Andersson, Maria Bergholtz, and Paul Johannesson, editors, *Natural Language Processing and Information Systems*, volume 2553 of *Lecture Notes in Computer Science*, pages 183–190. Springer Berlin Heidelberg, 2002.

- [36] Elena Demidova, Xuan Zhou, and Wolfgang Nejdl. Freeq: an interactive query interface for freebase. In *Proceedings of the 21st international conference companion on World Wide Web, WWW '12 Companion*, pages 325–328, New York, NY, USA, 2012. ACM.
- [37] Abdigani Diriye, Ann Blandford, and Anastasios Tombros. Exploring the impact of search interface features on search tasks. In Mounia Lalmas, Joemon Jose, Andreas Rauber, Fabrizio Sebastiani, and Ingo Frommholz, editors, *Research and Advanced Technology for Digital Libraries*, volume 6273 of *Lecture Notes in Computer Science*, pages 184–195. Springer Berlin Heidelberg, 2010.
- [38] Abdigani Diriye, Ann Blandford, and Anastasios Tombros. When is system support effective? In *Proceedings of the third symposium on Information interaction in context, IiX '10*, pages 55–64, New York, NY, USA, 2010. ACM.
- [39] Abdigani Mohamed Diriye. *Search interfaces for known-item and exploratory search tasks*. PhD thesis, University College London, 2012.
- [40] Mira Dontcheva, Steven M. Drucker, Geraldine Wade, David Salesin, and Michael F. Cohen. Summarizing personal web browsing sessions. In *Proceedings of the 19th annual ACM symposium on User interface software and technology, UIST '06*, pages 115–124, New York, NY, USA, 2006. ACM.
- [41] Huizhong Duan, Rui Li, and ChengXiang Zhai. Automatic query reformulation with syntactic operators to alleviate search difficulty. In *Proceedings of the 20th ACM international conference on Information and knowledge management, CIKM '11*, pages 2037–2040, New York, NY, USA, 2011. ACM.
- [42] Geoffrey B. Duggan and Stephen J. Payne. Text skimming: The process and effectiveness of foraging through text under time pressure. *Journal of Experimental Psychology: Applied*, 15(3):228–242, 2009.
- [43] John Duncan and Glyn W. Humphreys. Visual search and stimulus similarity. *Psychological Review*, 96(3):433–458, 1989.
- [44] Caroline M. Eastman and Bernard J. Jansen. Coverage, relevance, and ranking: The impact of query operators on web search engine results. *ACM Transactions on Information Systems*, 21(4):383–411, October 2003.
- [45] E. N. Efthimiadis. Query expansion. *Annual Review of Information Science and Technology*, 31:121–187, 1996.

Bibliography

- [46] Yuka Egusa, Hitomi Saito, Masao Takaku, Hitoshi Terai, Makiko Miwa, and Noriko Kando. Using a concept map to evaluate exploratory search. In *Proceedings of the third symposium on Information interaction in context*, IIX '10, pages 175–184, New York, NY, USA, 2010. ACM.
- [47] D. Ellis. A behavioural approach to information retrieval system design. *Journal of Documentation*, 45(3):171–212, 1989.
- [48] David Ellis, D. Cox, and K. Hall. A comparison of the information seeking patterns of researchers in the physical and social sciences. *Journal of Documentation*, 49(4):356–369, 1993.
- [49] David Ellis and Merete Haugan. Modelling the information seeking patterns of engineers and research scientists in an industrial environment. *Journal of Documentation*, 53(4):384–403, 1997.
- [50] Mathias S. Fleck and Stephen R. Mitroff. Rare targets are rarely missed in correctable search. *Psychological Science*, 18(11):943–947, 2007.
- [51] Clifton Forlines and Ravin Balakrishnan. Improving visual search with image segmentation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1093–1102, New York, NY, USA, 2009. ACM.
- [52] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. The vocabulary problem in human-system communication. *Communications of the ACM*, 30(11):964–971, November 1987.
- [53] Jens Gerken, Mischa Demarmels, Stefan Dierdorf, and Harald Reiterer. Hyper-Scatter - Modellierungs- und Zoomtechniken für Punktdiagramme. In Herczeg and Kindsmüller, editors, *Mensch & Computer 2008: Viel mehr Interaktion*, 8. Konferenz für interaktive und kooperative Medien, 2008.
- [54] Jens Gerken, Mathias Heilig, Hans-Christian Jetter, Sebastian Rexhausen, Mischa Demarmels, Werner A. König, and Harald Reiterer. Lessons learned from the design and evaluation of visual information-seeking systems. *International Journal on Digital Libraries (IJDL)*, 10(2–3):49–66, 2009.
- [55] Gene Golovchinsky. Queries? Links? Is there a difference? In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '97, pages 407–414, New York, NY, USA, 1997. ACM.

- [56] Gene Golovchinsky, Abdigani Diriye, and Tony Dunnigan. The future is in the past: designing for exploratory search. In *Proceedings of the 4th Information Interaction in Context Symposium, IIX '12*, pages 52–61, New York, NY, USA, 2012. ACM.
- [57] Gene Golovchinsky, Anthony Dunnigan, and Abdigani Diriye. Designing a tool for exploratory information seeking. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems, CHI EA '12*, pages 1799–1804, New York, NY, USA, 2012. ACM.
- [58] Christian Grün, Jens Gerken, Hans-Christian Jetter, Werner König, and Harald Reiterer. Mediovis – a user-centred library metadata browser. In Andreas Rauber, Stavros Christodoulakis, and A Min Tjoa, editors, *Research and Advanced Technology for Digital Libraries*, volume 3652 of *Lecture Notes in Computer Science*, pages 174–185. Springer Berlin Heidelberg, 2005.
- [59] Jaroslaw Gustak. Proaktive Suchunterstützung in ezDL. Diplomarbeit, Universität Duisburg-Essen, 2013.
- [60] Craig Harris, Alisdair Owens, Alistair Russell, and Daniel Alexander Smith. mspace: Exploring the semantic web. A technical report in support of the mspace software framework. Technical report, University of Southampton, December 2004.
- [61] Sandra G. Hart. NASA-Task Load Index (NASA-TLX); 20 years later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 50(9):904–908, 2006.
- [62] Sandra G. Hart and Lowell E. Staveland. Development of NASA-TLX (task load index): Results of empirical and theoretical research. In Peter A. Hancock and Najmedin Meshkati, editors, *Human Mental Workload*, volume 52 of *Advances in Psychology*, pages 139–183. North-Holland, 1988.
- [63] Joyce M. Hawkins and Robert Allen, editors. *The Oxford Encyclopedic English Dictionary*. Clarendon Press, Oxford, 1991.
- [64] Marti A. Hearst. Clustering versus faceted categories for information exploration. *Communications of the ACM*, 49(4):59–61, April 2006.

Bibliography

- [65] Marti A. Hearst and Emilia Stoica. NLP support for faceted navigation in scholarly collections. In *Proceedings of the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries*, NLP4DL '09, pages 62–70, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.
- [66] Mathias Heilig, Mischa Demarmels, Werner A. König, Jens Gerken, Sebastian Rexhausen, Hans-Christian Jetter, and Harald Reiterer. Medioviz: Visual information seeking in digital libraries. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '08, pages 490–491, New York, NY, USA, 2008. ACM.
- [67] William Hersh, Andrew Turpin, Susan Price, Benjamin Chan, Dale Kraemer, Lynetta Sacherek, and Daniel Olson. Do batch and user evaluations give the same results? In Emmanuel Yannakoudakis, Nicholas J. Belkin, Mun-Kew Leong, and Peter Ingwersen, editors, *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 17–24, New York, 2000. ACM.
- [68] Kingsley Hughes-Morgan and Max L. Wilson. Information vs interaction: examining different interaction models over consistent metadata. In *Proceedings of the 4th Information Interaction in Context Symposium*, IIX '12, pages 72–81, New York, NY, USA, 2012. ACM.
- [69] Isto Huvila. The Cool and Belkin faceted classification of information interactions revisited. *Information Research*, 15(4), 2010.
- [70] Jessica Ignalski, Matthias Jordan, and Sascha Kriewel. Evaluierung von Darstellungsvarianten für Anfragevorschläge bei der Informationssuche. In *Proceedings of the IR Workshop at LWA 2012, Dortmund, Germany*, September 2012.
- [71] P. Ingwersen. Polyrepresentation of information needs and semantic entities, elements of a cognitive theory for information retrieval interaction. In Bruce W. Croft and C. J. van Rijsbergen, editors, *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 101–111, London, 1994. Springer-Verlag.
- [72] Peter Ingwersen. Cognitive perspectives of information retrieval. *The Journal of Documentation*, 52(1):3–50, 1996.

- [73] Alpa Jain and Gilad Mishne. Organizing query completions for web search. In *Proceedings of the 19th ACM international conference on Information and knowledge management, CIKM '10*, pages 1169–1178, New York, NY, USA, 2010. ACM.
- [74] Bernard J. Jansen and Udo Pooch. Assisting the searcher: utilizing software agents for web search systems. *Internet Research: Electronic Networking Applications and Policy*, 14(1):19–33, 2004.
- [75] Bernard J. Jansen and Amanda Spink. An analysis of web searching by European alltheweb.com users. *Information Processing & Management*, 41(2):361–381, 2005.
- [76] Bernard J. Jansen, Amanda Spink, and Jan Pedersen. A temporal comparison of AltaVista web searching. *Journal of the American Society for Information Science and Technology*, 56(6):559–570, 2005.
- [77] Hans-Christian Jetter, Jens Gerken, Werner König, Christian Grün, and Harald Reiterer. Hypergrid — accessing complex information spaces. In *People and Computers XIX — The Bigger Picture, Proceedings of HCI 2005*, 2005.
- [78] Hideo Joho, Claire Coverson, Mark Sanderson, and Micheline Beaulieu. Hierarchical presentation of expansion terms. In *Proceedings of the 2002 ACM Symposium on Applied Computing, SAC '02*, pages 645–649, New York, NY, USA, 2002. ACM.
- [79] Matthias Jordan. DAFFODIL: Proaktive Vorlagefunktionen. Diplomarbeit, Universität Dortmund, FB Informatik, 2005.
- [80] Mika Käki and Anne Aula. Controlling the complexity in comparing search user interfaces via user studies. *Information Processing & Management*, 44(1):82–91, 2008.
- [81] Makoto P. Kato, Tetsuya Sakai, and Katsumi Tanaka. Structured query suggestion for specialization and parallel movement: effect on search behaviors. In *Proceedings of the 21st international conference on World Wide Web, WWW '12*, pages 389–398, New York, NY, USA, 2012. ACM.
- [82] Diane Kelly, Amber Cushing, Maureen Dostert, Xi Niu, and Karl Gyllstrom. Effects of popularity and quality on the usage of query suggestions during infor-

Bibliography

- mation search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 45–54, New York, NY, USA, 2010. ACM.
- [83] Diane Kelly, Karl Gyllstrom, and Earl W. Bailey. A comparison of query and term suggestion features for interactive searching. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 371–378, New York, NY, USA, 2009. ACM.
- [84] Diane Kelly, David J. Harper, and Brian Landau. Questionnaire mode effects in interactive information retrieval experiments. *Information Processing and Management*, 44(1):122–141, 2008.
- [85] Diane Kelly, Chirag Shah, Cassidy R. Sugimoto, Earl W. Bailey, Rachael A. Clemens, Ann K. Irvine, Nicholas A. Johnson, Weimao Ke, Sanghee Oh, Anezka Poljakova, Marcos A. Rodriguez, Megan G. van Noord, and Yan Zhang. Effects of performance feedback on users' evaluations of an interactive ir system. In *Proceedings of the second international symposium on Information interaction in context*, IiiX '08, pages 75–82, New York, NY, USA, 2008. ACM.
- [86] Andruid Kerne, Eunyee Koh, Steven M. Smith, Andrew Webb, and Blake Dworaczyk. combinFormation: Mixed-initiative composition of image and text surrogates promotes information discovery. *ACM Transactions on Information Systems*, 27(1):5:1–5:45, December 2008.
- [87] Justin Kruger and David Dunning. Unskilled and unaware of it: How difficulties in recognizing one's own incompetence lead to inflated self-assessments. *Journal of Personality and Social Psychology*, 77(6):1121–1134, 1999.
- [88] C. C. Kuhlthau. Inside the search process: Information seeking from the user's perspective. *Journal of the American Society for Information Science*, 42(5):361–371, 1991.
- [89] Yanen Li, Huizhong Duan, and ChengXiang Zhai. CloudSpeller: query spelling correction by using a unified hidden markov model with web-scale resources. In *Proceedings of the 21st international conference companion on World Wide Web*, WWW '12 Companion, pages 561–562, New York, NY, USA, 2012. ACM.
- [90] Fernando Loizides and George R. Buchanan. What patrons want: supporting interaction for novice information seeking scholars. In *Proceedings of the 9th*

- ACM/IEEE-CS joint conference on Digital libraries*, JCDL '09, pages 427–428, New York, NY, USA, 2009. ACM.
- [91] Carla Teixeira Lopes and Cristina Ribeiro. Measuring the value of health query translation: An analysis by user language proficiency. *Journal of the American Society for Information Science and Technology*, 64(5):951–963, 2013.
- [92] Aran Lunzer and Kasper Hornbæk. Subjunctive interfaces: Extending applications to support parallel setup, viewing and control of alternative scenarios. *ACM Transactions on Computer-Human Interaction*, 14(4):17:1–17:44, January 2008.
- [93] Saadia Malik. Interactive retrieval with XML documents. Dissertation, Universität Duisburg Essen, Fakultät Ingenieurwissenschaften, 2009.
- [94] Gary Marchionini. Exploratory search: from finding to understanding. *Communications of the ACM*, 49(4):41–46, 2006.
- [95] Tamaryn Menneer, Nick Donnelly, Hayward J. Godwin, and Kyle R. Cave. High or low target prevalence increases the dual-target cost in visual search. *Journal of Experimental Psychology: Applied*, 16(2):133–144, 2010.
- [96] Stefano Mizzaro. How many relevances in information retrieval? *Interacting With Computers*, 10(3):303–320, 1998.
- [97] Tanja Niederl. *Untersuchungen zu kumulativen psychischen und physiologischen Effekten des fliegenden Personals auf der Kurzstrecke*. PhD thesis, Universität Kassel, January 2008.
- [98] Marcus Nitsche and Andreas Nürnberger. Trailblazing information: An exploratory search user interface. In Sakae Yamamoto, editor, *Human Interface and the Management of Information. Information and Interaction Design*, volume 8016 of *Lecture Notes in Computer Science*, pages 230–239. Springer Berlin Heidelberg, 2013.
- [99] Vicki L. O’Day and Robin Jeffries. Orienteering in an information landscape: how information seekers get from here to there. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, pages 438–445, New York, NY, USA, 1993. ACM.

Bibliography

- [100] Jane E. Raymond, Kimron L. Shapiro, and Karen M. Arnell. Temporary suppression of visual processing in an RSVP task: An attentional blink? *Journal of Experimental Psychology: Human Perception and Performance*, 18(3):849–860, 1992.
- [101] Harald Reiterer, Christian Jetter, Werner A. König, Jens Gerken, and Christian Grün. Zoomtechniken zur Exploration komplexer Informationsräume am Beispiel HyperGrid. In *Mensch und Computer 2005; Kunst und Wissenschaft - Grenzüberschreitung der interaktiven ART*, Mensch und Computer 2005, pages 143–153, 2005.
- [102] Marc Rettig. Prototyping for tiny fingers. *Communications of the ACM*, 37(4):21–27, April 1994.
- [103] Daniel E. Rose and Danny Levinson. Understanding user goals in web search. In *Proceedings of the 13th International Conference on World Wide Web, WWW '04*, pages 13–19, New York, NY, USA, 2004. ACM.
- [104] André Schaefer. *Arbeitsteilung zwischen direkter Manipulation und proaktiven Software-Agenten in Informationssystemen*. PhD thesis, University of Duisburg-Essen, 2008.
- [105] André Schaefer, Matthias Jordan, Claus-Peter Klas, and Norbert Fuhr. Active support for query formulation in virtual digital libraries: A case study with DAFFODIL. In A. Rauber, C. Christodoulakis, and A. M. Tjoa, editors, *Research and Advanced Technology for Digital Libraries. Proc. European Conference on Digital Libraries (ECDL 2005)*, Lecture Notes in Computer Science, Heidelberg et al., 2005. Springer.
- [106] m. c. schraefel, Yuxiang Zhu, David Modjeska, Daniel Wigdor, and Shengdong Zhao. Hunter gatherer: interaction support for the creation and management of within-web-page collections. In *Proceedings of the 11th international conference on World Wide Web, WWW '02*, pages 172–181, New York, NY, USA, 2002. ACM.
- [107] m.c. schraefel, Max Wilson, Alistair Russell, and Daniel A. Smith. mSpace: improving information access to multimedia domains with multimodal exploratory search. *Communications of the ACM*, 49(4):47–49, 2006.

- [108] Johann Schrammel, Michael Leitner, and Manfred Tscheligi. Semantically structured tag clouds: an empirical evaluation of clustered presentation approaches. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 2037–2040, New York, NY, USA, 2009. ACM.
- [109] Guy Shani and Noam Tractinsky. Displaying relevance scores for search results. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '13, pages 901–904, New York, NY, USA, 2013. ACM.
- [110] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *IEEE Symposium on Visual Languages*, pages 336–343, Sep 1996.
- [111] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. Technical Report CS-TR-3665, University of Maryland, Department of Computer Science, July 1996.
- [112] Sidney L. Smith. Color coding and visual separability in information displays. *Journal of Applied Psychology*, 47(6):358–364, 1963.
- [113] A. Spink. Study of interactive feedback during mediated information retrieval. *Journal of the American Society for Information Science*, 48:382–394, 1997.
- [114] Maximilian Stempfhuber. Objektorientierte dynamische Benutzungsoberflächen - ODIN. Forschungsberichte 6, Informationszentrum Sozialwissenschaften, 2003.
- [115] Emilia Stoica, Marti Hearst, and Megan Richardson. Automating creation of hierarchical faceted metadata structures. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 244–251, Rochester, New York, April 2007. Association for Computational Linguistics.
- [116] Xu Sun, Anshumali Shrivastava, and Ping Li. Query spelling correction using multi-task learning. In *Proceedings of the 21st international conference companion on World Wide Web*, WWW '12 Companion, pages 613–614, New York, NY, USA, 2012. ACM.
- [117] Donna P. Tedesco and Thomas S. Tullis. A comparison of methods for eliciting post-task subjective ratings in usability testing. In *UPA 2006 Conference*, UPA 2006, Bloomingdale, IL, USA, 2006. User Experience Professionals Association.

Bibliography

- [118] Kien Tjin-Kam-Jet, Dolf Trieschnigg, and Djoerd Hiemstra. Free-text search versus complex web forms. In Paul Clough, Colum Foley, Cathal Gurrin, GarethJ.F. Jones, Wessel Kraaij, Hyowon Lee, and Vanessa Mudoch, editors, *Advances in Information Retrieval*, volume 6611 of *Lecture Notes in Computer Science*, pages 670–674. Springer Berlin Heidelberg, 2011.
- [119] Anastasios Tombros and Mark Sanderson. Advantages of query biased summaries in information retrieval. In W. Bruce Croft, Alistair Moffat, Cornelis J. van Rijsbergen, Ross Wilkinson, and Justin Zobel, editors, *SIGIR*, pages 2–10, New York, 1998. ACM.
- [120] Anastasios Tombros and Mark Sanderson. Advantages of query biased summaries in information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 2–10. ACM, 1998.
- [121] James T. Townsend. Serial vs. parallel processing: Sometimes they look like tweedledum and tweedledee but they can (and should) be distinguished. *Psychological Science*, 1(1):46–54, 1990.
- [122] Vu T. Tran and Norbert Fuhr. Quantitative analysis of search sessions enhanced by gaze tracking with dynamic areas of interest. In *The International Conference on Theory and Practice of Digital Libraries 2012*, pages 468–473. Springer, September 2012.
- [123] Vu T. Tran and Norbert Fuhr. Markov modeling for user interaction in retrieval. In *SIGIR 2013 Workshop on Modeling User Behavior for Information Retrieval Evaluation (MUBE 2013)*, August 2013.
- [124] Anne M. Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12(1):97 – 136, 1980.
- [125] Thomas S. Tullis and Jacqueline N. Stetson. A comparison of questionnaires for assessing website usability. Technical report, Usability Professionals Association (UPA) 2004 Conference, June 2004.
- [126] Tom Tullis and Bill Albert. *Measuring the User Experience*. Morgan-Kaufman, 2008.

- [127] Andrew Turpin and Falk Scholer. User performance versus precision measures for simple search tasks. In Efthimis N. Efthimiadis, Susan T. Dumais, David Hawking, and Kalervo Järvelin, editors, *SIGIR*, pages 11–18. ACM, 2006.
- [128] Andrew H. Turpin and William Hersh. Why batch and user evaluations do not give the same results. In W. B. Croft, D. Harper, D. H. Kraft, and J. Zobel, editors, *Proceedings of the 24th Annual International Conference on Research and development in Information Retrieval*, pages 225–231, New York, 2001. ACM Press.
- [129] Pertti Vakkari. Exploratory searching as conceptual exploration. In *HCIR 2010: Proceedings of the Fourth Workshop on Human-Computer Interaction and Information Retrieval.*, HCIR '10, 2010.
- [130] Pertti Vakkari and Saira Huuskonen. Search effort degrades search output but improves task outcome. *Journal of the American Society for Information Science and Technology*, 63(4):657–670, 2012.
- [131] Tony Veale and Yanfen Hao. In the mood for affective search with web stereotypes. In *Proceedings of the 21st international conference companion on World Wide Web*, WWW '12 Companion, pages 429–432, New York, NY, USA, 2012. ACM.
- [132] Aravindan Veerasamy and Russell Heikes. Effectiveness of a graphical display of retrieval results. In *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '97, pages 236–245, New York, NY, USA, 1997. ACM.
- [133] Robert Villa, Iván Cantador, Hideo Joho, and Joemon M. Jose. An aspectual interface for supporting complex search tasks. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 379–386, New York, NY, USA, 2009. ACM.
- [134] Ryen W. White, Bill Kules, Steven M. Drucker, and m.c. Schraefel. Introduction. *Communications of the ACM*, 49(4):36–39, 2006.
- [135] Ryen W. White and Gary Marchionini. Examining the effectiveness of real-time query expansion. *Information Processing & Management*, 43(3):685–704, 2007. Special Issue on Heterogeneous and Distributed IR.

Bibliography

- [136] Ryen W. White and Resa A. Roth. Exploratory search: Beyond the query-response paradigm. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 1(1):1–98, 2009.
- [137] Rebekah Willson and Lisa M. Given. The effect of spelling and retrieval system familiarity on search behavior in online public access catalogs: A mixed methods study. *Journal of the American Society for Information Science and Technology*, 61(12):2461–2476, 2010.
- [138] Max L. Wilson and m.c. schraefel. Bridging the gap: Using IR models for evaluating exploratory search interfaces. In *SIGCHI 2007 Workshop on Exploratory Search and HCI*, 2007.
- [139] Max L Wilson, m.c. schraefel, and Ryen W. White. Evaluating advanced search interfaces using established information-seeking models. *Journal of the American Society for Information Science and Technology*, July 2009.
- [140] T. D. Wilson. On user studies and information needs. *Journal of Documentation*, 37(1):3–15, March 1981.
- [141] T. D. Wilson. Models in information behaviour research. *Journal of Documentation*, 55(3):249–270, June 1999. <http://informationr.net/tdw/publ/papers/1999JDoc.html>.
- [142] Jeremy M. Wolfe. What can 1 million trials tell us about visual search? *Psychological Science*, 9(1):33–39, 1998.
- [143] Jeremy M. Wolfe, Todd S. Horowitz, and Naomi M. Kenner. Rare items often missed in visual searches. *Nature*, 435(7041):439–440, May 2005.
- [144] Jeremy M. Wolfe, Todd S. Horowitz, Michael J. Van Wert, Naomi M. Kenner, Skyler S Place, and Nour Kibbi. Low target prevalence is a stubborn source of errors in visual search tasks. *Journal of Experimental Psychology: General*, 136(4):623–638, 2007.
- [145] Iris Xie and Soohyung Joo. Factors affecting the selection of search tactics: Tasks, knowledge, process, and systems. *Information Processing & Management*, 48(2):254–270, 2012.
- [146] Ka-Ping Yee, Kirsten Swearingen, Kevin Li, and Marti Hearst. Faceted metadata for image search and browsing. In *Proceedings of the SIGCHI Conference on*

Human Factors in Computing Systems, CHI '03, pages 401–408, New York, NY, USA, 2003. ACM.

- [147] Xiaojun Yuan. *Supporting Multiple Information-Seeking Strategies in a Single System Framework*. PhD thesis, Rutgers, The State University of New Jersey, October 2007.
- [148] Xiaojun Yuan and Nicholas J. Belkin. Supporting multiple information-seeking strategies in a single system framework. In Wessel Kraaij, Arjen P. de Vries, Charles L. A. Clarke, Norbert Fuhr, and Noriko Kando, editors, *SIGIR*, pages 247–254, New York, NY, USA, 2007. ACM.
- [149] Xiaojun Yuan and Nicholas J. Belkin. Investigating information retrieval support techniques for different information-seeking strategies. *Journal of the American Society for Information Science and Technology*, 61(8):1543–1563, 2010.