Personalized Web Exploration with Task Models

Jae-wook Ahn Peter Brusilovsky Daqing He Jonathan Grady Qi Li
School of Information Sciences, University of Pittsburgh
135 N. Bellefield Ave., Pittsburgh, PA 15256, USA
+1 412 6249404
{jaa38,peterb,dah44,jpg14,qil14}@pitt.edu

ABSTRACT
Personalized Web search has emerged as one of the hottest topics for both the Web industry and academic researchers. However, the majority of studies on personalized search focused on a rather simple type of search, which leaves an important research topic – the personalization in exploratory searches – as an under-studied area. In this paper, we present a study of personalization in task-based information exploration using a system called TaskSieve. TaskSieve is a Web search system that utilizes a relevance feedback based profile, called a "task model", for personalization. Its innovations include flexible and user controlled integration of queries and task models, task-infused text snippet generation, and on-screen visualization of task models. Through an empirical study using human subjects conducting task-based exploration searches, we demonstrate that TaskSieve pushes significantly more relevant documents to the top of search result lists as compared to a traditional search system. TaskSieve helps users select significantly more accurate information for their tasks, allows the users to do so with higher productivity, and is viewed more favorably by subjects under several usability related characteristics.

Categories and Subject Descriptors
H.3.1 [**Content Analysis and Indexing**]: *Indexing method;* H.3.3 [**Information Search and Retrieval**]: *Information filtering; Relevance feedback;* H.3.5 [**Online Information Services**]: *Web-based services.*

General Terms
Experimentation, human factors, performance.

Keywords: Personalization, task-based information exploration, adaptive search, user profile, task model, empirical study

1. INTRODUCTION
Personalized Web search emerged as one of the hottest topics for both the Web industry and academic researchers [21]. Unlike traditional "one-size-fits-all" search engines, personalized search systems attempt to take into account interests, goals, and preferences of individual users in order to improve the relevance of search results and the overall retrieval experience. In the context of a tight competition between search engines and technologies, personalization is frequently considered as one of the technologies that can deliver a competitive advantage.

WWW 2008, April 21–25, 2008, Beijing, China.
ACM 978-1-60558-085-2/08/04.

While personalized search was a focus of many papers and projects over at least the last 10 years, the absolute majority of these projects aimed to support a rather simple type of search, which is known as *lookup search* [20]. The assumption behind this type of search is that the answer to the user's information need is located on one or few Web pages. Correspondingly, the goal of search personalization is to insure that one or more of the target pages is retrieved and is pushed to the top of the results list despite such known problems as short queries, synonymy, and polysemy. To achieve this goal, personalized search systems build *profiles of user interests* and use them for personalized query expansion and result ranking.

It is commonly accepted that lookup search is just one of several types of searches performed by Web users. Marchionini [20] calls searches "beyond lookup" as *exploratory searches*, which can be further distinguished as *search to learn* and *search to investigate*. Exploratory search assumes that the user have some broader *information need*, which can't be simply solved by a "relevant" Web page, but requires multiple searches interleaved with browsing and analyzing the retrieved information. For example, an academic researcher plans her visit to a foreign city while attending a conference, and wants to investigate the most appropriate means of transportation, places to stay, and nearby sights to visit.

As long as the Web is getting closer and closer to becoming a primary source for all kinds of information, more and more users of Web search engines run exploratory searches to solve their everyday needs. In addition, a growing proportion of users, such as information analysts, are engaged in Web based exploratory search professionally by the nature of their jobs. It makes exploratory Web search both attractive and an important focus for research on search personalization. Yet, only a very small fraction of projects devoted to search personalization seek to support exploratory search.

This paper presents our efforts to create a personalized system for *task-based information exploration* where the information needs and the corresponding search processes are defined by the task assigned to the user. This kind of exploratory search is typical for a range of professional users, such as information analysts. Capitalizing on the nature of task-based information exploration, we transformed the traditional profile of user interests to a more focused *task model* and introduced several innovative techniques for both constructing and utilizing this model. To explore the value of these techniques, we developed *TaskSieve*, a platform for task-based information exploration. A recent study performed with TaskSieve demonstrated that our personalization techniques can significantly improve the user's and system's performance in the task-based exploratory search context. The following sections present an account of our work. We start with an analysis of similar work. Then we present TaskSieve and a study comparing it to a traditional search engine. At the end we summarize the obtained results and our plans for future work.

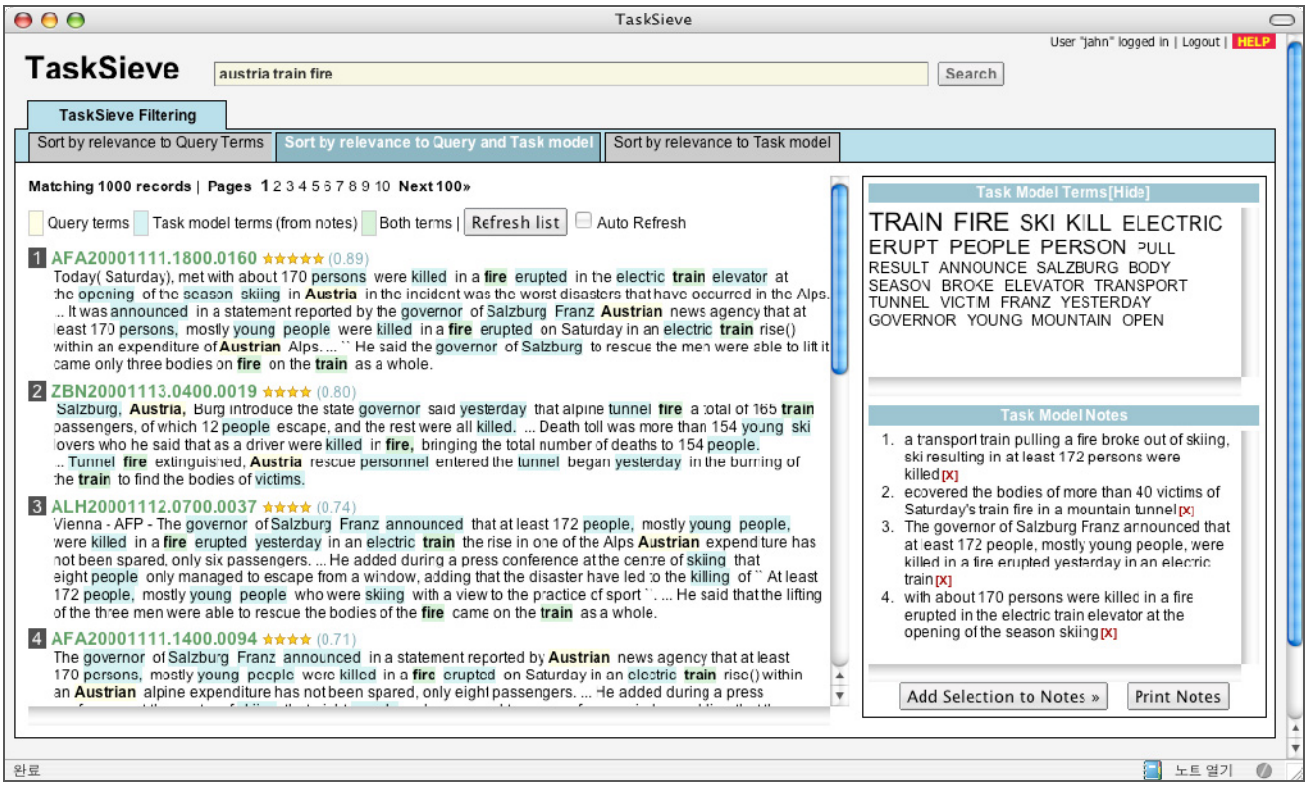

Figure 1 TaskSieve interface

2. PERSONALIZED SEARCH: THE STATE OF THE ART

The work on search personalization has deep roots in the area of information retrieval and filtering, which can be traced to the use of keyword-level user profiles in selective dissemination of information [14] and early research on relevance feedback [24].

Relevance feedback, which obtains extra information beyond users' initial requests, has been shown to be an effective technique for improving retrieval performance in experiments [9; 25]. However, there are still questions about the true effectiveness of traditional relevance feedback, because users of Web search systems seldom want to make extra effort for providing relevance judgments [17]. A range of projects attempted to replace explicit relevance feedback with various kinds of implicit feedback indicators [7; 17]: from time spent [4] to eye movements [26]. Yet, the trade-off between reliable, but hard to collect explicit feedback and less reliable implicit feedback still exists. It is still a research challenge to design feedback techniques, which can combine the precision of explicit feedback with the unobtrusiveness of implicit feedback.

Relevance feedback has long been used for short-term search personalization through a sequence of query refinements. However, it was the integration of relevance feedback with user profiles (originally applied for selective dissemination of information) that launched the research on personalized information access. Utilizing relevance feedback as the mechanism for collecting information about the user, personalized information access systems construct a dynamic profile of user needs and interests and apply it to improve the quality of search, information filtering, and recommendation.

User profiles can be distinguished by the timeframe of their construction and usage. Two of the most typical cases are short-term profiles, which model immediate information need, and long-term profiles, which attempt to model user general interests and preferences. Surprisingly enough, we were not able to find any work on intermediate-level profiles, which, for example, can model user's larger-scale information exploration tasks pursued over a longer period of time, yet not equal to the general interests. Several works on collaborative information retrieval use an idea of a search task or *quest* [15; 16], however these works do not build task profiles.

The classic form of user profile, which can even be found in a textbook [19], is a weighed vector of keywords. This form of the user profile is applied in the majority of personalized systems, including personalized search systems [21] and content-based recommendation systems [22]. A number of systems attempted to build more complicated profiles, mostly by integrating several keyword vectors within a single profile. For example, WebMate [3] used several keyword vectors for each user *interest*. YourNews [1] also separates long-term and short-term profiles for each interest. Finally a number of projects explore more innovative approaches to long-term user profiling such as networked profiles or ontology-based profiles; however, there is still no reliable evidence that advanced profiles are superior to simpler keyword profiles. A good review of major types of user profiles for personalized search is provided in [8].

The most typical usage of the profile is to rank information items. Filtering and recommendation systems simply rank the presented items by their similarity to the user profile. In personalized search systems, the profile is fused with the query and applied for filtering and re-ranking of initial search results. Referring to any specific paper is difficult, since there are dozens of reported systems using this approach: see [21; 22] for a review.

However, ranking is not the only aspect information retrieval or filtering systems attempt to improve to better assist their users. User access to information can also be improved by generating better document surrogates, such as search snippets and summaries.

The summaries are important for indicating the content of the documents, and to provide clues about the potential relevance to the users' searches. Many different ideas have been used to generate summaries, including Keyword in Context approach (see [12] or Google search results), automatic text summarization [6] and passage generation [13]. Most of these methods either are query independent (like text summarization), or related to query only; however, some pioneering information filtering projects attempted to apply user profiles to generate personalized document summaries [5].

Nearly all personalized search systems hide user profiles from their users, so the user can neither view nor edit the profiles. A few projects studied whether the profiles and relevant information should be presented to users in their searches. [18] found that it is useful to make this part of information available to the users. However, studies that attempted to make user profiles both visible and editable to users [1; 27] found that the ability to edit user profiles may negatively affect system performance.

In summary, personalized search and user profiling are popular and well-explored areas. Yet there are research challenges and space for improvement in nearly all reviewed sub-areas. Task-based information exploration provide a unique and unstudied context for examining some less explored approaches such as intermediate-level profiling, adaptive snippets, or user profile visualization.

3. TaskSieve: A PLATFORM FOR TASK-BASED INFORMATION EXPLORATION

3.1 Task Model

Unlike the majority of known personalized search systems, TaskSieve aims to support the task-based exploratory search process. In place of a traditional model of user interests, TaskSieve applies a more focused *task model*, which attempts to accumulate information about the task explored by the user. A task model is a relatively short-term model in comparison with a long-term model of user interests, yet it can support the user over a lengthy sequence of queries (frequently spread over several sessions) as long as the user is focused on a specific task. The model is constructed unobtrusively while the users are interacting with the system. There is no task description to enter, as in AntWorld [16] or SERF [15]. The user simply starts working on a new task by entering the first query and processing the list of initial, but not yet adapted, search results. Standard stemming and stopword removal procedures are applied to these task model vectors. Among the hundreds of terms from the user notes, the top 300 important terms are selected according to their TF-IDF weights in the document corpus.

TaskSieve was designed to assist users who perform exploratory searches reasonably often, i.e., it focuses on relatively experienced searchers up to the level of professional information analysts. These users appreciate more powerful and sophisticated information access tools; but as we learned from our earlier work on adaptive filtering [1], they also want to be in control of the system's work and highly value the transparency of the system mechanisms. This requirement contradicts the traditional approach taken by personalized search systems, which tend to make personalization decisions without user consent and hide the underlying personalization mechanism. Unlike these systems, TaskSieve attempts to make the personalization transparent. It starts with using a relatively simple, but easy to understand task model form: weighted term vectors. In addition, it makes the task model visible to the user through the model viewer (upper right in Figure 1). The viewer shows terms, which form the task model, sorted by their importance (weight). A larger font size is used for more important terms. The model visualization is kept up-to-date according to the task model changes. This visible task model is expected to help users to understand the task-based engine of TaskSieve; however, users who consider the model less useful or need more space for other parts of the interface can hide the viewer at any time.

3.2 Personalized Ranking

As in many other personalized search systems, TaskSieve uses the post-filtering approach to personalized search results, using the task model to re-rank the plain search results retrieved by a search engine (Figure 2). The idea of re-ranking is to promote documents, which are more relevant to the user task as measured by their similarity to the task model. For transparency reasons, TaskSieve uses the traditional linear approach to combine query relevance and task relevance:

(1) Retrieve documents along with their relevance scores by submitting the user query to a search engine.

(2) Calculate similarity scores between retrieved documents and the model.

(3) Calculate combined score of each document by equation (1).

$$alpha * Task\_Model\_Score + (1 - alpha) * Search\_Score \quad (1)$$

(4) Re-rank the initial list by the combined score from step 3.

TaskSieve uses Indri[1] as a search engine and normalizes its scores, dividing by the maximum score (score of the rank 1 item) of the corresponding list (step 1). Task model scores are calculated by measuring the similarity between each document vector and the task model vector. We use BM25 [23] for this task (step 2) and the scores are also normalized.

In equation (1), *alpha* controls the power of the task model. It can vary freely from 0 to 1. The traditional approach is to fix *alpha* either ad-hoc, or by learning the "optimal" value and using this value to fuse all search results. We believe this approach contradicts the desire of our target users to be "in control", and instead give the control over the fusion to users. TaskSieve allows the users to alternate among three preset ranking options: "Sort by relevance to Query Terms", "Sort by relevance to Query and Task model", and "Sort by Relevance to Task Model" (which correspond to *alpha* values 0, 0.5, and 1.0 respectively). If *alpha* is 0, the ranking is the same as plain search. If *alpha* is 1.0, then the search rank is completely ignored. If *alpha* is 0.5, which is the default, the system considers equally the importance of query and task.

Figure 1 shows an example of the task-based ranked list (lower left in the screen). A user enters a query "austria train fire". Important task model terms such as "TRAIN", "FIRE", "SKI", and "KILL" were extracted from the user notes in order to re-rank the original search result to the query "austria train fire" generated from the baseline search engine. Just above the ranked list, there are three tabs labeled with three ranking options explained above. Users can

[1] http://www.lemurproject.org/indri

explore different query terms and control the task-based post-filtering engine in order to complete their tasks.

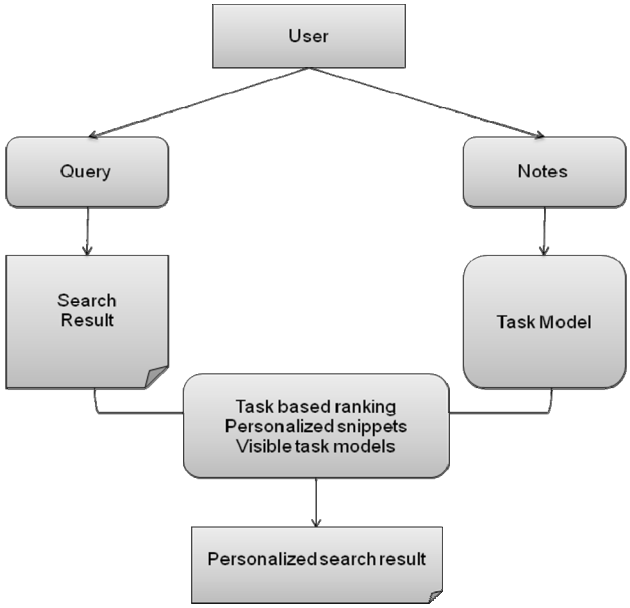

Figure 2 TaskSieve system diagram

3.3 Task-Infused Snippets

As mentioned in the introduction, the task model is more focused and, as a result, more precise and reliable than a typical model of interests. It allows extending the use of the task model beyond the traditional re-ranking. A promising innovation explored in TaskSieve is *task-infused snippets* (Figure 3). Snippets are document surrogates displayed by search engines below document titles to help the user in selecting the relevant results. Most modern search engines construct snippets from sentences that include query terms, and even highlight these terms to stress the relevance of the document to the query. In task-based exploratory search, however, the document relevance to the task could be equally or more important than its relevance to the query. To help users in selecting task-relevant documents, TaskSieve constructs personalized task-based snippets by extracting top 3 most relevant sentences given users' query terms and task model terms. The following briefly describes the procedure:

(1) Divide the document contents into sentences.

(2) Calculate sentence-to-query relevance score for each sentence.

(3) Calculate sentence-to-task-model score for each sentence.

(4) Linearly combine the two scores from step 2 and 3 for each sentence, as described in the previous section.

(5) Sort the sentences by the combined score (step 4) and select top 3 sentences.

(6) Display the selected sentences in the order of their appearance in the document.

The *alpha* preset used for re-ranking the document list is equally applied to this process so users can see different sentences according to their task model contents and the preset selection.

TaskSieve also emphasizes the query and the task model terms appearing in all surrogate sentences. As shown in Figure 3, query terms ("Austria") are highlighted in yellow and shown in bold face; task model terms ("kill", "tunnel", "Salzburg", etc.) are highlighted in blue; and the terms belonging to both the query and the model ("train" and "fire") are highlighted in green and shown in bold face. The highlighting and text formatting make the terms pop out, helping users to see how each surrogate sentence is relevant to the query or task model.

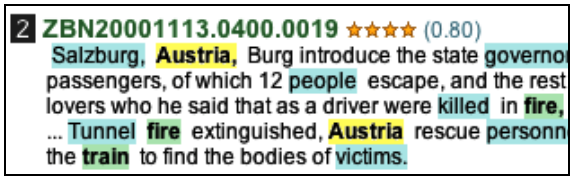

Figure 3 Task infused document surrogates and highlights

3.4 Using Notebook for Task Model Update

In addition to the innovative ways of using the task model, TaskSieve explores a new approach to updating this model. This approach is based on the idea of a *notebook*. A notebook is a collection of document fragments (which we call *notes*) extracted and saved by the user. From one side, the notebook supports the user's need to collect the most important information for further processing. A note collection tool is frequently used in the process of information exploration (analysts call it a "shoebox"). From the other side, the content of the collected notes represents the task much better than the documents from which they are extracted. It allows TaskSieve to use the content of the saved notes to increase the quality of modeling in comparison with existing personalized search systems.

TaskSieve encourages the user to take notes and make this process very simple. The users can highlight any text from the search snippets or whole document and add it to the notebook by a single button click. When a new note is saved, it is displayed in the notebook (lower right in Figure 1). Each note can be removed by clicking on the "X" beside it if the user doesn't think she needs it anymore.

Every action in the notebook (adding and removing) instantly affects the task model – the weights of the task model terms found in the added or removed note are increased or decreased correspondingly. The important task model terms in the task model viewer are immediately updated to reflect the new set of weights. The ranking of the current search result list can also be updated immediately after each task model change if *Auto Refresh* is checked. However, this option is switched off by default, because our previous studies in a similar context of information filtering demonstrated that automatic update of ranking confuses users and causes performance decreases [10]. Therefore, TaskSieve offers a "Refresh list" button, allowing the user to re-rank the search results according to the current state of the task model whenever it is most convenient for her.

4. THE STUDY DESIGN

To assess the value of TaskSieve's task modeling features, an experimental study was performed using a full-fledged version of TaskSieve (i.e. with the task model) as the experimental system. For comparison, the baseline system takes a simplified version of TaskSieve, which does not have the task model functionalities (i.e., query-based ranking only, query-term highlighting only, no model visualization, no model-extended snippets). Thus the baseline system replicated a modern search engine interface.

We attempted to examine two groups of hypotheses in the study:

H1: At the operational level, the experimental system – by having the task model and task-based features – performs better.

H1-1: The experimental system will generate results with higher precision in terms of result ordering.

H1-2: Users of the experimental system will demonstrate higher productivity measured by numbers of selected notes and higher task performance in terms of precision on selected notes.

H1-3: Users of the experimental system will actively use the innovative features of TaskSieve, such as the ability to view the user model and to vary the influence of the query and the model on document ranking.

H2: At the subjective level, users prefer the experimental system over the baseline system.

H2-1: Users are more satisfied with the experimental system.

H2-2: Users appreciate the ability to change the query-profile weighting of search results.

H2-3: Users appreciate the ability to view the task model

H2-4: Users appreciate the document surrogates generated from the task model and the query.

The document collection used in this experiment is an expanded TDT4 English test corpus, in which there are 28,390 English documents published between October 2000 to January 2001 [11]. The expansion happened at the topic and ground truth aspects. 18 of the original TDT4 topics are enriched into so- called GALE topics to resemble the tasks performed by intelligence analysts. Each GALE topic contains an overarching task theme and up to 10 different but related sub-tasks (Figure 4). The search outcomes of these topics are a group of selected useful passages that can be used to answer the questions raised in these tasks/subtasks. The relevance criteria of the selected passages are based on utility, which can be seen as a function between topical relevance and material novelty.

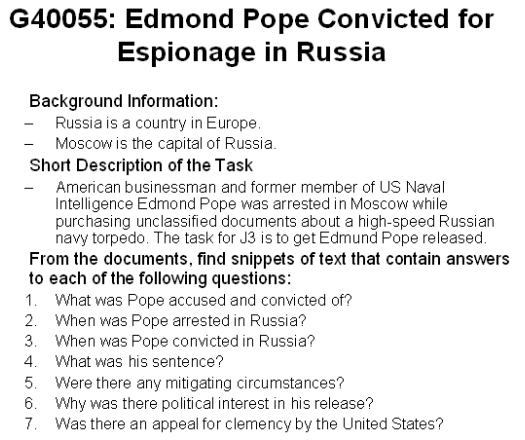

Figure 4 An example of the GALE topics

For this study, we selected the whole expanded TDT4 corpus. Six GALE topics were selected (see Table 1), but each subject performed search tasks only on four of them. The experiment was split into two 1-hour sessions. During each session, subjects completed one search task on the baseline system and one on the experimental system. The orders and the combination of the systems and the topics were randomized according to Latin square design to control for any possible learning and fatigue effects.

Table 1 The six selected GALE topics

TDT4 Topic ID	Title
40001	Galapagos Oil Spill
40038	Earthquake hits India's Gujarat state
40055	Edmond Pope Convicted of Espionage in Russia
41011	Turkish Prison Riots
41012	Trouble in the Ivory Coast
41019	Iliescu wins Romanian elections

Subjects were tested independently, using the following procedure:

1. Prior to the first session, subjects read a one-page introductory statement to the experiment, and completed a demographic questionnaire focusing on their search experience and consumption of news.
2. Prior to the first search task, subjects were given a 15-minute training that included a walkthrough of the TaskSieve system and instructions on completing a search task.
3. For each search task:
   a. Subjects were given a one-page task description providing a brief background to the task scenario and a list of questions to answer (Figure 4).
   b. Subjects were given 20 minutes to search for and collect useful notes that provided answers to the questions in the task scenario. The required minimum length of a note was one full sentence (i.e. to provide enough context for a superior to determine why the selected information was useful.) This part simulated information foraging stage of analyst work.
   c. At the end of the allotted 20 minutes, subjects were asked to process their notebook by annotating each note with the number(s) of the question(s) from the task scenario answered by the note. This part simulated simple sense-making and report preparation.
   d. Subjects completed a post-task questionnaire.
4. At the end of both sessions, subjects took part in a 10 –minute exit interview and were compensated.

The evaluation metrics used in our study include system performance and user performance measures, which are based on document and passage level precision (see the further discussion in the next paragraph), and the usability measures regarding the systems' support in task-based exploration processes, especially those examining the interactions between the users and the systems. Samples of these measures are the productivity of selecting useful information, usage data of different integration modes of the query and the task model, and users' subjective comments on the systems.

Ten subjects recruited from the School of Information Sciences at the University of Pittsburgh participated in the experiment between October 2, 2007 and October 19, 2007. To ensure they best fit the profile of an information analyst, participants were required to be native English speakers and have been graduate students trained in search (i.e. a graduate course in information retrieval.) Five of the

ten subjects were female, and the age range of all subjects was 20-56. On a ten-point scale (10 being the highest), the participants mean rating of their search abilities was 8.6 with a mode of 10.

5. PERFORMANCE ANALYSIS

5.1 System Performance

The job of a personalized search system is to "push" the most relevant items to the top of the ranked list. Having relevant items at the very top of the list (top 5 or top 10) is especially important since Web users are known to pay most attention to the first screen of results. To measure the performance of the proposed experimental system we calculated *precision at rank 5 and 10*. TaskSieve displays 10 articles per page so we can calculate the precision by counting the number of relevant documents among top 5 or top 10 documents in the ranked lists and then divide it by 5 or 10 accordingly. For example, if we found 4 relevant documents from top 5 documents from a retrieved list, the precision at rank 5 here is 0.8. We did this document level precision for evaluating the system performance because the system returns *documents* as a response to the user queries. First two rows of Table 2 and Figure 5 show the averages of the precisions at rank 5 and 10. The experimental system shows higher precision at rank 5, 0.93, which means more than 4 articles (0.93 * 5 = 4.65) are relevant to the topic on average. The baseline system showed poorer performance (0.87 and 0.86 for both rank 5 and 10). The experimental system outperforms the baseline system overall and the differences between them are statistically significant (paired sample Wilcoxon signed rank test).

Table 2 Document-level performance

Precision	Baseline	Experimental	*p*
System at Rank 5	0.87	0.93	0.014
System at Rank 10	0.86	0.92	0.011
User document access	0.88	0.96	0.027

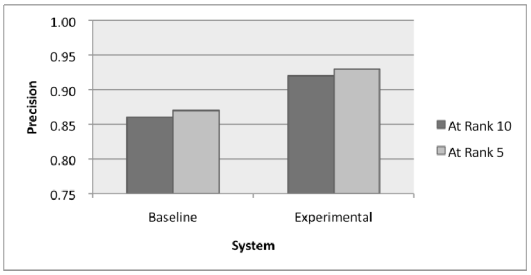

Figure 5 System performance

5.2 User Document Access Precision

The user performance parameter, which we expected to be most affected by the improved system performance, is *document access precision*, i.e., the ratio of relevant opened documents to the total number of documents clicked by the subjects. While the ultimate measure of user performance is a collection of highly relevant passages assembled into a high quality report, the first step to it is the ability to notice and open relevant documents. Document access precision is exactly what an adaptive search system is able to improve directly guiding users to good documents by applying task-influenced ranking and task-infused snippets.

To calculate the document access precision, we recorded all documents the subjects clicked and then compared them with document relevance assessment in our ground truth information. The data confirmed our expectation. As shown in Table 2, subjects using the experimental system opened and examined more relevant documents than those with the baseline system (0.96 vs. 0.88) and this different was statistically significant (paired sample Wilcoxon signed rank test). This table makes it also easy to observe that the user document access precision in the baseline system closely matched the system precision, i.e., the users performed exactly as well as the system allowed. However, in the experimental system, document access precision was noticeably better than the system precision making it nearly perfect 0.96. I.e., the users of the experimental system were able to perform better than the system ranking allowed them. We hypothesize that it could be an effect of task-infused snippets, which provided users with an additional help in selecting relevant documents.

While the match between system and document opening precision provides some implicit evidence that the users of both systems followed systems' recommendation expressed in the form of ordering, it is interesting to check an evidence that users take advantage of the system's ability to push relevant documents to the top of the ranked list. Otherwise, the performance growth in the experimental system could be attributed to other factors like task-infused snippets, rather than system performance. To uncover this, we observed the rank information of the documents that were examined or annotated by the users. If these documents are mostly highly ranked items in the system's results, we can hypothesize that the users trusted the systems' ranking and there is a correlation between user and system performance. Table 3 shows that the average ranks of the documents examined and annotated by the subjects are very high (smaller number means higher ranks on top of the list). We believe that the subjects had strong beliefs on the system performance so that they concentrated mostly on highly ranked articles for their task reports.

While Table 3 does not allow us to notice any significant difference between user trust in two systems, a deeper analysis presented in Table 4 uncovers the difference. As we can see, the subjects using the experimental system did not proceed beyond page 2 (rank 20) at all before they opened full documents. Even the frequency of checking the second page for subjects using the experimental system is minimal compared to that of the baseline. The baseline system users checked the second page and the deeper pages at the rate of 9% and 3% respectively. The difference for the first page is significant (chi-square test, $p < 0.01$), which suggests that subjects trusted the experimental system more and were more satisfied with its highly ranked items.

Table 3 Average ranks of open and add note actions

User action	Baseline	Experimental
Open document	3.79	3.66

Table 4 Page navigation

Page	1	2	3	4	>= 5	Total
Baseline	244 (88%)	25 (9%)	3 (1%)	1 (0%)	5 (2%)	278
Experi-mental	258 (98%)	6 (2%)	0 (0%)	0 (0%)	0 (0%)	264

5.3 User Performance

During the study, the subjects were asked to collect useful notes for their notebooks (i.e. task reports). An interesting question is whether the experimental system helped subjects to be more effective not only in terms of finding relevant documents more accurately, but also to collect higher numbers of relevant notes. Therefore, we examined the number of notes the subjects had taken. Figure 6

shows the results for each half of the 20 min. session. Overall, the users of the experimental system were much more productive. They took more notes for the task report (329 vs. 292) despite of opening slightly fewer documents (264 vs. 278) and were saving notes at much higher rate, especially within the first 10 minutes. This is important evidence in favor of experimental system's better ability to help users in finding relevant information, especially in time-critical contexts. It is also interesting to observe that only 9 notes in the baseline system were saved directly from regular snippets (i.e., without document opening) while 40 notes were saved from task-infused snippets in the experimental system. It can be considered as another evidence for the value of task-infused snippets.

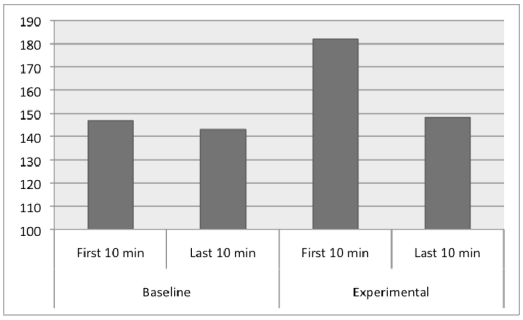

Figure 6 User productivity

Of course, taken alone, the volume of collected information is not sufficient to compare two systems reliably. It could be that the users of the experimental systems collected more notes, sacrificing the quality. However, both the quality and the quantity of collected notes are important for preparing a good report. The quality of user notes, which we call passage precision, is the hardest performance parameter to evaluate.

Our calculation of passage precision takes advantage of the fact that two human annotators generated the ground truth. The formula (2), which is derived from [2], calculates the precision of a passage against the ground truth, where *overlap_length* is the character length of the common text chunk between a user's selection and the ground truth; *weight* is the weight of the ground truth combining the two annotators mark-ups, where the weight can be one of five levels: 0, 0.25, 0.5, 1, 1.25, 2; *miss_length* is the character length of the part of the passage that has no overlap with the ground truth. Here the 0.5 associated with *miss_length* is the penalty.

$$\frac{\sum_{i \in passage \cap groundtruth} Overlap\_length_i \times weight_i}{\sum_{i \in passage \cap groundtruth} Overlap\_length_i \times weight_i + \sum_{i \in passage - passage \cap groundtruth} miss\_length_i \times 0.5} \quad (2)$$

Table 5 summarizes the user performance measured by passage precision. We found no significant difference between the cumulative precision of notes collected by the users of both systems. Thus the users of the experimental system were able to collect about 10% more notes without sacrificing the quality of the notes.

Table 5 User performance

System	Baseline	Exp.	*P*
Passage precision	0.86	0.85	0.398
Passage precision (Top N=292)	0.86	0.96	<0.001

However, the cumulative data does not provide reliable evidence that the experimental system can improve both quantity and quality of collected notes. To compare the system on a deeper level, we examined how many good or bad notes the subjects made using each system. Figure 7 shows the distribution of user notes by their precisions. We can notice that that there is little difference between systems in the number of poor and average-quality notes, however, the experimental system helped users to find more *top-quality* (precision=1.0) notes: 219 vs. 182.

An easy way to combine both quality and quantity of notes is to order the notes collected by the users of both systems by the note quality and compare the graphs. This is done in Figure 8, which shows the change of average precision of top N annotations. This graph compares the average precision scores of two systems when the subjects made top N high precision annotations. Both are dropping as they collect more passages but the baseline drops more sharply and the experimental system shows higher precision until the N reaches 325. The right end point N=329 is the total number of annotations made using the experimental system and it is where the overall average precision in Table 5 was calculated. Subjects with the baseline made 292 annotations, so we compared the average precision at N=292 where the note counts of both systems are same and the experimental system showed significantly higher average precision (Wilcoxon signed rank test, $p < 0.001$).

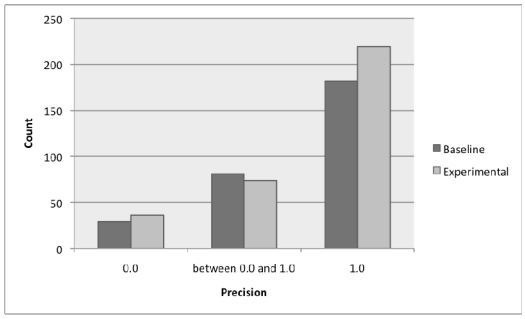

Figure 7 Number of passages per each precision level

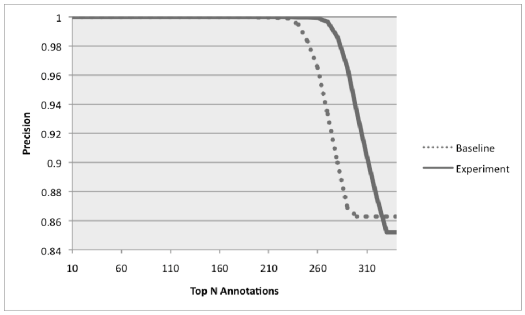

Figure 8 Average precision of top N Annotations

5.4 Task Model and Flexible Ranking

The task model is one of the key components of TaskSieve. We allow users to see the terms comprising the task model during the whole sessions and let them change the importance of the task model so that they can manipulate the power of its effect during the retrieval process. Users were provided with three task model weight presets: 1.0, 0.5, and 0.0. The question is, however, whether the users benefited from this transparency and flexibility. To answer this question, we analyzed the user action logs to determine how frequently they changed the task model visibility and the present ranking options.

According to the logs, none of the users ever hid the task model viewer. Given that it occupies a sizeable portion of the screen, which could be otherwise taken by the notebook, we can hypothesize that the visibility of the task model was important for

the user. Yet this hypothesis is not fully confirmed if the users never changed the default settings. The situation with the ranking flexibility is much clearer – users appreciated the ability to change the ranking and used it frequently. Figure 9 shows the frequency of the ranking preset change and the amount of time subjects used presets during their exploratory searches. They used the half-and-half preset most of the time (46% in frequency and 54% in time). The next-favored preset was alpha=0.0, which removes the effect of the task model completely (35% in frequency and 30% in time). The least favored preset was alpha=1.0, which considers the task model only. While subjects could consider this preset an extreme because it ignores the effect of their own queries, its use was quite considerable (19% in frequency and 17% in time).

Because we had provided the preset alpha=0.5 as a default, we also need to consider whether it was favored by the subjects simply because it was the default. In Figure 10, we discarded the default use of this preset (alpha=0.5). The data shows that the frequency of direct preset change to 0.5 (from 1.0 or 0.0) is 24% and the amount of time subjects spent with this preset when it was selected directly was 14%. In total, we registered 124 explicit preset changes – more than 12 for each subject at average! Even though users favored the alpha=0.0 preset when making explicit choices, we should not disregard that they were still using the task model more (alpha=1.0 or 0.5) than the query only preset (alpha=0.0), where the frequency was 43% and the time spent was just 30% of the whole sessions.

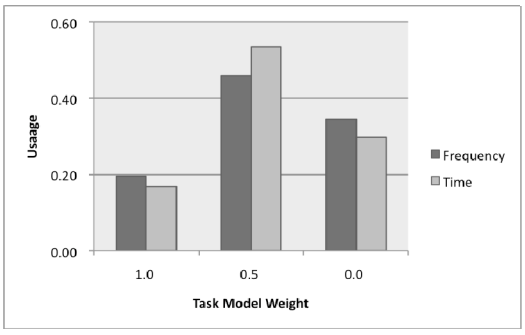

Figure 9 Task model weights

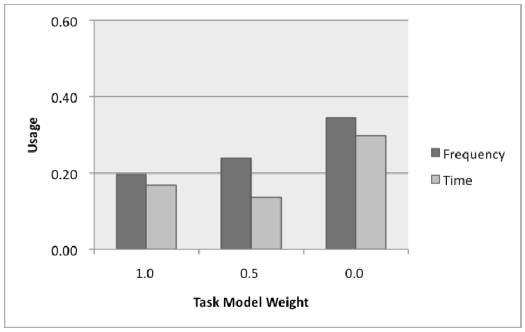

Figure 10 Task model weights (removed defaults)

We also tried to compare the user and system performances during each of these three presets. In Table 6 and Table 7, we separated the user/system performance measures discussed in the previous sections by the change of the presets. It clearly shows that the system performance (Table 7) was better when the subjects used the task model (when alpha=1.0 or 0.5) than when they just used the system in the query only mode (alpha=0.0). Also, alpha=1.0 task model weight mode was the best when we examined the note precision even though alpha=0.5 showed relatively poorer performance. User performance in terms of the open precision was about the same among the three modes, but the subjects showed higher rate of activity counts while they were taking notes, opening documents, and searching (217 vs. 110, 198 vs. 66, and 80 vs. 41 respectively). It is also interesting to note that the document access precision with preset alpha=0.0 (query only) was much higher than the average access precision in the baseline system. It provides some evidence that the users really mastered the preset manipulation picking up the most appropriate presets for different queries.

Table 6 User performance with varying task model weights

User Performance	1.0	0.5	0.0
Note Precision	0.97	0.80	0.90
Note Count	38	179	110
Document Access Precision	0.92	0.96	0.97
Document Access Count	51	147	66
Search Count	13	67	41

Table 7 System performance with varying task model weights

System Performance	1.0	0.5	0.0
Precision at Rank 10	0.98	0.95	0.87
Precision at Rank 5	0.96	0.95	0.87

Table 8 Post-questionnaire (*experimental system ONLY)

Q#	Text of Question
1	Were you familiar with this topic before the search?
2	Did the passages and their documents provide you sufficient information for your summary?
3	When choosing to view a full document, was it mostly because you found useful information in the passage?
4	Were you confident in the system's ability to find useful information on this topic?
5	Was it easy to mark up useful snippets using this system?
6*	Did you find the query-vs-profile weight adjustment helpful in finding useful information?
7*	Was displaying the terms in your task model helpful to you?
8*	Did you find the inclusion of passages with terms from your task model helpful in finding useful information?
9*	Did you find the highlighting of terms from your task model in the passages helpful?
10	Overall, did you have a positive experience with this system?

6. USER SUBJECTIVE ANALYSIS

Following each search task, subjects were given a post-questionnaire (Table 8) to assess their satisfaction with the version of TaskSieve used to complete the assigned task. Using a 5-point Likert scale, subjects were asked to rate their level of agreement (1=Not at All; 5 = Extremely) regarding their familiarity with the assigned task topic (Question 1); the sufficiency of news provided (Question 2); usefulness of the document summaries (i.e. surrogates) in the search results (Question 3); their ability to find useful passages (i.e. snippets) (Question 4), the system's ease of use (Question 5), and overall satisfaction with the system (final question.) For the experimental system only, subjects were asked to rate the utility of the features related to the task model – query-profile weighting adjustment, task model visualization, task-model based snippets, and highlighting of task model terms in snippets.

Chi-square tests were performed on the questionnaire data to determine significant differences in subject responses by system and by topic. Table 9 shows the mean post-questionnaire responses by system. There were no significant differences in any of the users'

subjective ratings of the baseline and experimental systems across all topics, nor were there any significant differences in the users' subjective ratings among topics across both systems. Although the average overall satisfaction is slightly higher for the experimental System, this difference is not significant either. Thus the evidence does not support our hypothesis that users are *more satisfied* using this system over the baseline (Hypothesis 2-1) At the same time the lack of significant difference is a result that speaks in favor of the experimental System. TaskSieve extended the traditional search interface with several innovative features. These features significantly improved user and system performance, yet they inevitably made system interface more busy and complicated. Despite that, the user self-evaluation of confidence, ease of use, and satisfaction, has not dropped: the users were as comfortable with the more complicated and powerful system as with a traditional search interface.

Still we hoped for more expecting that the empowered users will be more satisfied with the experimental system. What may have tempered the subjects' satisfaction? In exit interviews, half of the subjects remarked that despite changing their query, the same document(s) seemed to appear at the top of the ranked result lists for the 50-50 query-profile weighting option. Our tasks instructed subjects to find and collect useful snippets of text, implying that their notes contain both relevant and novel information. Because TaskSieve does not consider the novelty of information, subjects often were presented with highly relevant but redundant documents at the top of the ranked result lists. Furthermore, the TDT4 corpus includes numerous articles containing passages of text repeated from previous articles, contributing to this redundancy.

Table 9 Mean post-search questionnaire responses by system. (* Experimental System ONLY)

Question	Baseline	Experimental
Sufficiency of News	3.94	3.81
Usefulness of Document Surrogates	3.69	3.63
Ability to Find Useful Snippets	3.63	3.56
Ease of Use	3.94	4.25
Query-Profile Weighting *	-	3.38
Task Model Visualization *	-	3.00
Task Model Based Snippets *	-	3.50
Highlighting - Task Model Terms *	-	3.44
Overall Satisfaction	3.81	3.88

Regarding other hypotheses at the subjective level, most subjects in the exit interviews said they found the ability to view the task model more useful than the overall neutral rating from the questionnaires would indicate, thus providing support for Hypothesis 2-3. Those favoring the task model visualization said it either helped them discover new terms relevant to the task, or it helped them better understand how the experimental system constructed the task model. In support of Hypothesis 2-4, most subjects also found the highlighting of terms from the task model and/or query helpful in the early stages of a search task, but less so toward the end. All terms from the task model were highlighted in the document surrogates, which became problematic as a user's task model grew in size. Subjects suggested limiting the highlighting to only the n-highest weighted terms from the task model, or varying the shades of the highlighting according to term weight.

7. DISCUSSION AND FUTURE WORK

In this paper, we present our efforts to create a personalized system called TaskSieve for task-based information exploration. Although personalized search has become a hot research topic, most of the previous projects concentrated on simple lookup searches, and makes personalization in explorative search an under-studied area.

TaskSieve is unique because of its several innovative features. It aims to integrate users short-term interests (as queries) with their relative long-term task characteristics and preference (as the task model) to cope with the multiple iterations of the exploration of search space. The system also subjects the integration under the users control through a set of predefined combination modes, so that the system and the process are more flexible and transparent. As the second innovative feature, TaskSieve returns documents surrogates that are task-infused by the generation of their content and by the highlighting of terms within them. This gives the users more direct clues about the potential relevance of the documents to not only their queries, but also the task model. Finally, TaskSieve also provides on-screen visualization of the task model as the third innovation feature so that such information is always available to the users during all their searches.

We conducted an empirical study with human subjects using TaskSieve for task-based exploration searches. The study demonstrates that TaskSieve – compared to a traditional search system – can utilize the information available in the task model to return significantly more relevant documents at the top of the ranked lists. The data also show that the average precision values of the baseline system's ranked lists at the last 10 minutes is still lower than that of the experimental system's first 10 minutes. This shows that the improvement obtained through task model is even higher than that through human users learning about the search topic and the retrieval system over the time.

The study also shows that TaskSieve can help user performance, too. TaskSieve's users were not only able to select notes that contained significantly more relevant information, they also can select more notes even during the first 10 minutes of the search session when they were still relatively unfamiliar with the search tasks. This demonstrates that TaskSieve significantly improved the productivity of the users' searches.

The flexibility in controlling the integration mode between queries and the task model also demonstrates its usefulness. First, we observed subjects switching among the different modes in their searches. Second, the searches with the half-half mode produced the best results. Third, the searches in query-only mode produced better results than the baseline, which indicates that the users really mastered the preset manipulations and used the appropriate mode for different searches. Finally, it is clear that none of the modes significantly dominates all the searches. All of these indicate that it really makes sense for TaskSieve to let users decide the best mode for their searches.

However, our study did reveal some limitations to the current version of TaskSieve. The most salient one is lacking of novelty detection to remove redundancy in results. The search tasks require users to interact with the system multiple times, and their goals are to collect relevant and novel information. Therefore, the inability to identify and remove redundant information in the results was shown to affect system and user performance, as well as users' subjective views of the system.

Our future work, therefore, will be in the direction of introducing novelty into the system. We also plan to implement more flexible

integration modes for combining queries and the task model, and to design a more intuitive interface for enhanced searching and user control.

8. ACKNOWLEDGMENTS

This paper is partially supported by DARPA GALE project and by the National Science Foundation under Grant No. 0447083.

9. REFERENCES

[1] Ahn, J.-w., Brusilovsky, P., Grady, J., He, D., Syn, S.Y. Open user profiles for adaptive news systems: help or harm? in Proceedings of the 16th international conference on World Wide Web, WWW '07 (Banff, Canada), ACM 11-20.

[2] Allan, J. HARD Track Overview in TREC 2003 High Accuracy Retrieval from Documents in Proceedings of The Twelfth Text Retrieval Conference.

[3] Chen, L., Sycara, K. WebMate: A personal agent for browsing and searching in Proceedings of 2nd International Conference on Autonomous Agents (Agents'98) (St. Paul, MN), ACM Press 132-139.

[4] Claypool, M., Le, P., Wased, M., Brown, D. Implicit interest indicators in Proceedings of 6th International Conference on Intelligent User Interfaces (Santa Fe, NM, January 14-17, 2002), ACM Press 33-40.

[5] Díaz, A., Gervás, P. Personalisation in news delivery systems: Item summarization and multi-tier item selection using relevance feedback, Web Intelligence and Agent Systems (2005) 3, 3, 135-154.

[6] Dorr, B., Zajic, D., Schwartz, R. Cross Language Headline Generation for Hindi, This volume (2003).

[7] Fox, S., Karnawat, K., Mydland, M., Dumais, S., White, T. Evaluating implicit measures to improve web search, ACM Transactions on Information Systems (2005) 23, 2, 147-168.

[8] Gauch, S., Speretta, M., Chandramouli, A., Micarelli, A. User profiles for personalized information access, In: Brusilovsky, P., Kobsa, A., and Neidl, W., Eds. The Adaptive Web: Methods and Strategies of Web Personalization. Springer-Verlag, Berlin Heidelberg New York, 2007, 54-89.

[9] Harman, D.K. Relevance feedback revisited in Proceedings of Proceedings of ACM-SIGIR 92.

[10] He, D., Brusilovsky, P., Grady, J., Li, Q., Ahn, J.-w. How Up-to-date should it be? The Value of Instant Profiling and Adaptation in Information Filtering in Proceedings of the 2007 international conference on Web Intelligence, WI '07 (Silicon Valey, CA, USA), IEEE in press.

[11] He, D. et al. An Evaluation of Adaptive Filtering in the Context of Realistic Task-Based Information Exploration, Information Processing and Management (2007).

[12] He, D. et al. Making MIRACLEs: Interactive Translingual Search for Cebuano and Hindi, ACM Transactions on Asian Lnaguage Information Processing (2003) 2, 3, 219-244.

[13] He, D., Wang, J., Luo, J., Oard, D.W. iCLEF 2004 at Maryland: Summarization Design for Interactive Cross-Language Question Answering in Proceedings of The Proceeding of Cross-Language Evaluation Forum (CLEF 2004).

[14] Houseman, E.M., Kaskela, D.E. State of the art of selective dissemination of information, IEEE Transactions on Engineering Writing and Speech (1970) 13, 2, 78-83.

[15] Jung, S., Harris, K., Webster, J., Herlocker, J.L. SERF: Iintegrating human recommendations with search in Proceedings of ACM 13th Conference on Information and Knowledge Management, CIKM 2004 (Washington, DC, November 8-13, 2004 571-580.

[16] Kantor, P.B., Boros, E., Melamed, B., Meñkov, V., Shapira, B., Neu, D.J. Capturing human intelligence in the net, Communications of the ACM (2000) 43, 8, 112-116.

[17] Kelly, D., Teevan, J. Implicit feedback for inferring user preference: a bibliography, SIGIR Forum (2003) 37, 2, 18-28.

[18] Koenemann, J., Belkin, N.J. A case for Interaction: A Study of Interactive Information Retrieval Behavior and Effectiveness in Proceedings of Proceedings of CHI '96 (New York 205-212.

[19] Korfhage, R.R. 1997 Information storage and retrieval. Wiley Computer Publishing, N.Y.

[20] Marchionini, G. Exploratory search: From finding to understanding, Communications of the ACM (2006) 49, 4, 41-46.

[21] Micarelli, A., Gasparetti, F., Sciarrone, F., Gauch, S. Personalized search on the World Wide Web, In: Brusilovsky, P., Kobsa, A., and Neidl, W., Eds. The Adaptive Web: Methods and Strategies of Web Personalization. Springer-Verlag, Berlin Heidelberg New York, 2007, 195-230.

[22] Pazzani, M.J., Billsus, D. Content-based recommendation systems, In: Brusilovsky, P., Kobsa, A., and Neidl, W., Eds. The Adaptive Web: Methods and Strategies of Web Personalization. Springer-Verlag, Berlin Heidelberg New York, 2007, 325-341.

[23] Robertson, S., Walker, S., and Hancock-Beaulieu, M. Okapi at TREC-7: automatic ad hoc, filtering, VLC and Interactive track. In Proceedings of the Seventh Text REtrieval Conference. Gaithersburg, USA, November 1998.

[24] Rocchio Jr, J.J. Relevance feedback in information retrieval, In: Salton, G., Ed. The SMART Retrieval System: Experiments in Automatic Document Processing. Prentice-Hall, Englewood Cliffs, NJ, USA, 1971, 313-323.

[25] Ruthven, I., Lalmas, M. A survey on the use of relevance feedback for informaton access systems, The Knowledge Engineering Review (2003) 18, 2, 95-145.

[26] Salojärvi, J., Puolamäki, K., Kaski, S. Implicit Relevance Feedback from Eye Movements in Proceedings of 15th International Conference on Artificial Neural Networks, ICANN 2005 (Warsaw, Poland, September 11-15, 2005), Springer-Verlag 513-518.

[27] Waern, A. User involvement in automatic filtering - an experimental study, User Modeling and User Adapted Interaction (2004) 14, 201-237.