Directing Exploratory Search: Reinforcement Learning from User Interactions with Keywords

Dorota Głowacka
Helsinki Institute for
Information Technology HIIT
Dep. of Computer Science
University of Helsinki
glowacka@cs.helsinki.fi

Tuukka Ruotsalo
Helsinki Institute for
Information Technology HIIT
Aalto University
tuukka.ruotsalo@aalto.fi

Ksenia Konyushkova
Helsinki Institute for
Information Technology HIIT
Dep. of Computer Science
University of Helsinki
ksenia.konyushkova@helsinki.fi

Kumaripaba Athukorala
Helsinki Institute for
Information Technology HIIT
Dep. of Computer Science
University of Helsinki
kumaripaba.athukorala@hiit.fi

Samuel Kaski
Helsinki Institute for
Information Technology HIIT
Aalto University
samuel.kaski@aalto.fi

Giulio Jacucci
Helsinki Institute for
Information Technology HIIT
Dep. of Computer Science
University of Helsinki
giulio.jacucci@helsinki.fi

ABSTRACT

Techniques for both exploratory and known item search tend to direct only to more specific subtopics or individual documents, as opposed to allowing directing the exploration of the information space. We present an interactive information retrieval system that combines Reinforcement Learning techniques along with a novel user interface design to allow active engagement of users in directing the search. Users can directly manipulate document features (keywords) to indicate their interests and Reinforcement Learning is used to model the user by allowing the system to trade off between exploration and exploitation. This gives users the opportunity to more effectively direct their search nearer, further and following a direction. A task-based user study conducted with 20 participants comparing our system to a traditional query-based baseline indicates that our system significantly improves the effectiveness of information retrieval by providing access to more relevant and novel information without having to spend more time acquiring the information.

Author Keywords
Adaptive Interfaces; Data Mining; Machine Learning; Recommender Systems; Information Filtering

ACM Classification Keywords
H.5.2. User Interfaces: User-centered design

General Terms
Information Retrieval

INTRODUCTION

In a typical interaction with an information retrieval system, the user expresses a specific information need, entered as a query, investigates the results returned by the search engine, and alters the query to direct the search to a chosen direction: a more specific subtopic or an alternative direction. As a result, users frequently have to carefully investigate the results to be able to reformulate their query. This behavior may have a confounding effect: result lists may be long and tedious to investigate, and relevant information may be widely scattered, and therefore hard for the user to extract [28].

Disadvantages of Query–Based Search Techniques

Searching for electronic information can be a multistage process. The information need evolves throughout the course of the search and even when the searched item is known, instead of jumping directly to the target, users typically navigate with small, local steps using their contextual knowledge as a guide. A recent survey found that users use long highly specific queries in less than a half of their searches, despite almost always knowing their information need upfront [26]. This allows users to specify less of their information need at once and provide a context in which to understand the intermediate results, but also lessens the users' cognitive burden by saving them from having to articulate exactly what they are looking for and allowing them to rely on established habits to get within the vicinity of their information need. Even in cases where users know what they are searching, it has been suggested that users try to avoid the cognitive overhead imposed by the difficulty of expressing their information needs exactly, by using simple and short queries and learning how to alter the query to achieve their goal. The effect of this finding becomes even more prominent in the case of exploratory information seeking, where users' needs are uncertain and underspecified in the first place, and to reduce the uncertainty users direct their search by exploring the information space step-by-step [25].

Relevance Feedback and IR Systems

While development usually aims at an optimal search engine that is able to return the best matching information based on a well-defined information need, the exploratory nature of the human information retrieval behavior has been addressed, e.g. by providing interactivity and control to users for better orientation and engagement [18, 26]. The first approach developed by the information retrieval community was the relevance feedback mechanism: users mark documents as relevant or non-relevant and the query model is updated based on the features present in the documents. Empirical studies have shown that experimental interactive IR systems benefit from term relevance feedback features [14]. However, evidence from user studies indicates that relevance feedback features are not used, or if they are, they are unlikely to result in retrieval improvements. Two main reasons for this were found [14]: (1) the relevance feedback often leads to a context trap, i.e. after a few iterations of feedback users have specified their context so strictly that the system is not able to propose anything new and users are trapped within the present set of results; (2) the cognitive load caused by the requirement to select relevant and non-relevant documents is high compared to typing in a new query and prevents users from actively engaging with the relevance feedback mechanisms.

Interface Design and User Feedback

The reason for the lack of use of relevance feedback has often been attributed to the specific intelligent user interface designs that do not allow users to conveniently provide feedback at a suitable level of granularity. Research has targeted to support users in giving feedback through a variety of techniques involving computational methods, to predict user needs based on the query history [1], rich user interface support with learning algorithms to assist users to comprehend the results and the existing information space [5, 8], and visualizing and summarizing the resulting information to enable faster relevance judgement of the quality of the information returned by the search engine [10, 13, 15, 19, 27].

Faceted Search and Search Context

The problem of getting trapped by context has been approached by using global features instead of contextual features or exploration techniques. One of the most successful ways to use global document set features in directing search is faceted search, also called faceted navigation or faceted browsing [30]. It is a technique for accessing information organized according to a faceted classification system, allowing users to explore a collection of information by applying multiple filters. A faceted classification system classifies each information element along multiple explicit global dimensions, enabling the classifications to be accessed and ordered in multiple ways rather than being based on contextually selected options. The problem with this approach is that the number of global features can be very large and users may have to be forced to select from a huge amount of options, making the user interface inconvenient and cognitively demanding [30].

Reinforcement Learning and Information Retrieval

Recent studies have shown that by modeling the search context, a system can provide much richer information about the search intention, limit the number of alternatives users need to select from to direct their search, and automate the tedious query reformulation process [29]. Reinforcement learning is a promising approach that can allow the system to utilize the search context in relevance feedback and, at the same time, to avoid the context trap. Reinforcement learning allows the system to trade between exploitation (moving towards more specific subtopics) and exploration (going towards alternative topics), and has been shown to be helpful in information retrieval [12, 23, 33]. Most of the existing work concentrates on using reinforcement learning for search personalization [23, 33]. Long-term implicit data acquisition is used to build user models that are then used as the basis for filtering content. In [22] a setting was introduced where the system learns the user preferences in an online manner without a labeled dataset, while taking into account similarities between documents and feedback from the users. Exploration–exploitation techniques are often used in tasks involving information retrieval or recommender systems, such as filtering [4], recommendation [12, 31, 34], ads placement [7, 17, 20] or image retrieval [3]. However, most of the information retrieval/recommender systems that employ reinforcement learning techniques rely on collecting information on users' habits and interests over a prolonged period of time, while in a typical search scenario users are more interested in the overall improvement of the search results within a given search session rather than hypothetical future search sessions.

Problem Solution

We propose that better support for exploration can be provided through learning from feedback on higher level representations of the data sets, such as topics or keywords, that are extracted from document features. This feedback provides enables applying machine learning techniques such as reinforcement learning to improve relevance, novelty and diversity of results. This allows users to direct their search using the offered keyword cues at any point of time without getting trapped in a context, or having to provide tedious document–level relevance feedback, or relying on implicit feedback mechanisms that may take long to converge. Users are experiencing diverse ways to interact with information, e.g. in smartphones and tablets, where multitouch revives the concept of direct manipulation as users expect visual representation of the option space, and rapid and incremental actions (and not only typing) [11]. Time is ripe fo the learning mechanism to predict the keywords based on the search session context and allow users to direct the search rather than merely re-ranking as in traditional personalization.

The resulting information access system couples advanced machine learning techniques with information visualization and interaction to boost exploratory search. The users can actively engage in an exploratory search loop where they manipulate article features such as keywords and ranking, and the underlying machine learning system offers them navigation options (keywords, articles) using an exploration–exploitation paradigm. We expect the search to become signicantly faster by allowing exploration and easier query manipulation. We have found a suitable abstract level on which it is convenient for the users to direct their search (in our case, the document keywords are the navigation options users can use to direct their search), and use observed interaction together with binary feedback to feed reinforcement learning–based optimization of further navigation options. We maintain that interactive choice of navigation options, such as topics and keywords, can facilitate exploration. We propose that this can support users in better directing the exploratory search nearer or further from the current context and following a direction.

System Assessment
We present results from a series of studies that demonstrate the benefits brought by the system. Computer simulations are first used to verify that it is beneficial to apply the reinforcement learning to user feedback on both the document features and the documents. A pre-experiment study is then carried out, aimed at evaluating the subjective experience and usability of the system. Finally, a task–based experiment with 20 subjects compares our system to a traditional query-based baseline. The performance is evaluated on three redefined measures based on the well known precision, recall and F-measures. They are redefined for the case of iterative queries, separately taking into account not only relevance but also obviousness and novelty. We also report the extent to which the suggested keywords are manipulated by users, as an indication of how much the system is used in directing the search.

SYSTEM OVERVIEW

The primary goal of the system is to assist scientists in finding and exploring the relevant literature on a given research topic quickly and effectively; although it can additionally be easily adapted to other domains. Reinforcement learning (RL) methods as well as visualization allow the user to assign relevance scores to the displayed keywords by moving them within the exploratory view provided by the system (keyword manipulation). Through keyword manipulation, the user can direct the search according to her interest, while the inbuilt RL mechanism helps the system to form a model of user's interests and suggest appropriate keywords in the next search iteration. The user modeling process is restarted once the user types in a new query and we build a new user model for each session to avoid the issue of "over-personalization". In this section, we describe the system design, its interface and the algorithms incorporated into the system.

Interface Design

The main idea behind the interactive interface is that instead of typing queries at each iteration, the user navigates through the contents by manipulating the keywords on the display, which results in new keywords appearing on the screen as well as a new set of documents being presented to the user. The user interface is presented in Figure 2 and an example search session in Figure 1.

The search starts with the user typing in a query, which results in a set of keywords being displayed in the circle on the left hand-side of the screen (exploratory view) and a set of articles being displayed on the right hand-side of the screen (Figure 2). The user can manipulate the keywords in the exploratory view to indicate their relevance: the closer to the center a given keyword is, the more relevant it is. The user can manipulate as many keywords as she likes in the exploratory view as well as drag keywords from underneath the displayed articles into the exploratory view to indicate their relevance to the search. After each iteration, new keywords and new articles are displayed. The search continues until the user is satisfied with the results.

We illustrate the interface and interaction design through a walkthrough that exemplifies a real information seeking task. In our example, a student writing an essay about "hand gestures" begins the seeking process by typing "hand gestures" as the query. The system retrieves a set of documents and adapts the content to match the user's feedback (Figure 1). On the first iteration, the documents and keywords are retrieved based on pseudo-feedback acquired from the top-ranked documents and visualized for the user. At this point, the students interest in "hand gesture recognition" increases (Iteration 1 in Figure 1) and she realizes that the keywords "language" and "communication" are not related to her information need. She provides feedback to the system by moving them outside of the exploratory view and by moving the keyword "recognition" to the center. The user then submits the feedback by clicking the center of the exploratory view. The system learns a more specific representation of the users information needs from the feedback, expresses it in terms of keywords, and retrieves and predicts a new set of documents and keywords (Iteration 2 in Figure 1). At the end of Iteration 2, the user decides to look at documents about "hidden Markov model". The document list now consists of documents related to "hand gesture recognition with hidden Markov models", but because of the novelty and diversity featured in the adaptation methods, the system also gives alternative keywords for the user to select. By moving them towards the center, the student could continue the seeking process.

System Design

The data flow from the system's perspective is illustrated in Figure 3. The three main blocks of the system are responsible for the initial retrieval and ranking of documents, and exploration in the keyword and the document spaces using RL.

The initial set of documents and their rankings are obtained through the Information Retrieval and Ranking module. Having received feedback on keywords, the system enters the exploratory loop. The explicit user feedback is sent to the Keywords Exploration and the Document Diversification modules. The Keywords Exploration module implements user model estimation using RL techniques (described below). The user model is a representation of the system's belief about the user's informational need at the current iteration of retrieval. The component receives feedback from the user and produces a list of keywords with weights which are passed on to the Information Retrieval and Ranking module, which predicts a new set of documents for the new search iteration based on the predicted user model. Thus, the dataset in the system is not static and it changes at every iteration based on the present, best estimation of the user model.

The Document Diversification module is responsible for determining the set and order of documents that are passed on to the Interface. The module uses exploration–exploitation

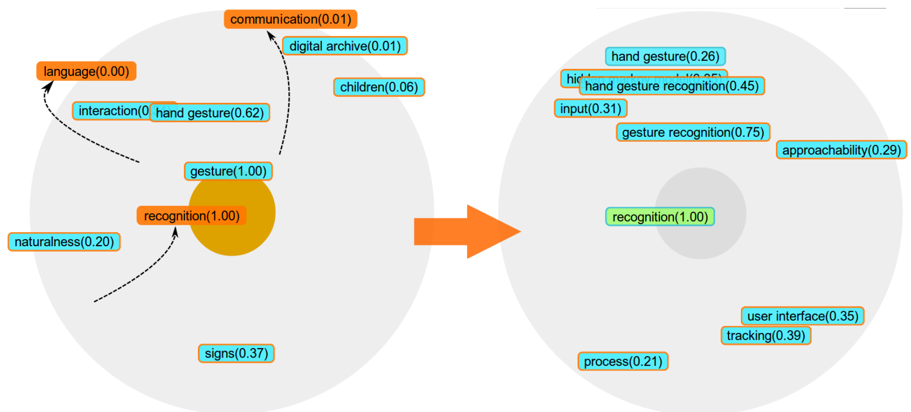

Figure 1. Feature manipulations. After receiving the initial set of keywords in the first iteration (left), the user indicates an increased importance of the keyword "recognition" by moving it towards the center of the exploratory view and indicates a reduced importance of the keywords "language" and "communication" by moving them outside the exploratory view. The keywords explicitly manipulated by the user are colored in orange. In the second iteration (right), new keywords have been predicted by the system on the basis of their estimated relevance and are positioned in the exploratory view. The new keywords can be distinguished by their orange borders.

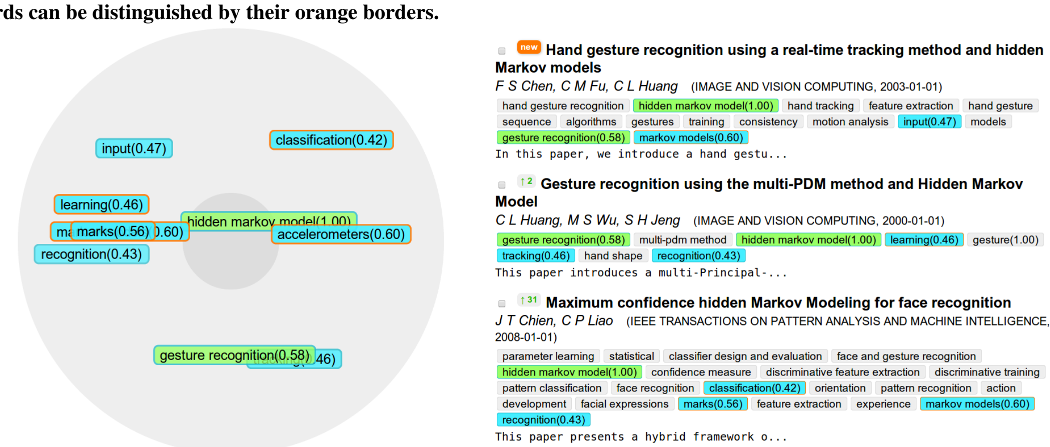

Figure 2. The document list after Iteration 2 has both new documents (labeled new) and documents whose rank increased from the previous round. The user has now obtained documents matching the information need. The exploratory view (left) also offers options for continuing the exploration in other potentially relevant directions, such as "accelerometers", "learning" or "classification".

techniques to sample a set of documents to display to the user, while keeping the ranking obtained from the Information Retrieval and Ranking module. The new set of documents is used in Keywords Exploration module to capture dependencies between keywords.

The user model is visualized in the exploratory view, which allows the user to give feedback to the system through keyword manipulation. A list of articles is also presented to the user. The system gets new feedback from the user and continues in the iterative feedback loop.

Retrieving and Ranking Documents

The system includes a separate retrieval model whose input is the user model and which produces a ranking for the documents and keywords attached to each of the ranked documents. The keywords are used to build a new estimate of the user model and the documents are shown for the user in the ranked order after diversification. The retrieval model allows a fast ranking of the top-n documents. This is motivated by a learning-to-rank approach where only the top-ranked documents and document features are used as the context for more

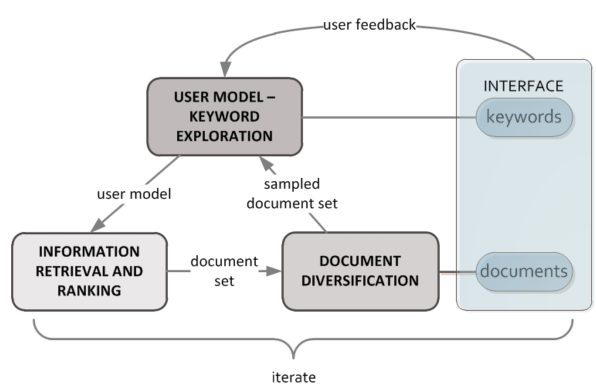

Figure 3. Overview of data flow in the exploratory search system

complex learning processes. In this way, we can reduce the number of documents and features that the RL methods have to deal with.

In order to estimate the relevance of each document d_j, we employ a standard language modeling approach of information retrieval. We estimate the initial ranking of documents d given a user model represented as the weighted keyword vector k consisting a weight $\hat{w}_i$ for each keyword. The ranking is computed according to the probability of documents given the user model. We use a probabilistic multinomial unigram language model. Documents are ranked by the probability that a query would be observed as a random sample from the respective document model. The probability of a document producing the user model is given by the language model M_{d_j} for document d_j by using maximum likelihood estimation (MLE):

$$\hat{P}(k|M_{d_j}) = \prod_{k_i \in k}^{i} \hat{w}_i \hat{P}_{mle}(k_i|M_{d_j}),$$

where M_{d_j} is the language model of document d_j and $\hat{w}_i$ is the estimation for the weight of a keyword k_i given by the present user model. Now the maximum likelihood of a keyword k_i is:

$$\hat{P}_{mle}(k_i|M_{d_j}) = \prod_{k_i \in k}^{i} P_\mu(k_i|d_j).$$

We avoid zero probabilities and improve the estimation by using the Bayesian Dirichlet smoothing [32]. The smoothed model is given by:

$$P_\mu(k|d_j) = \frac{c(k;d_j) + \mu p(k|C)}{\sum_k c(k;d_j) + \mu},$$

where the probability of a document generating a keyword is smoothed using the collection statistics $p(k|C)$, which is the estimate for a keyword k based on the whole collection C as opposite to only a relative count $c(k;d_j)$ in a single document d_j. The parameter μ is set to 2000 as suggested in the literature [32].

To keep the maximum likelihood estimation for the whole set of keywords occurring in the user model fast, we restrict the i being evaluated to the top ten keywords receiving the most of the weight mass at each iteration. The same set of keywords is also visualized for the user in the exploratory view.

User Model Estimation

We assume that the user is looking for a set of keywords related to her interest. The user feedback is given by a relevance score $r \in [0,1]$, where 1 indicates that the keyword is of high relevance. The user provides feedback by moving a keyword closer to or further from the center of the exploratory view: keywords in the center have relevance score 1 with the value getting smaller the further away from the center a keyword is moved (see Figure 1). Keywords placed on the edge of the exploratory view or beyond have relevance score 0. Keywords with relevance score 0 are excluded from appearing again in the exploratory view for the remainder of a given search session. The formal iteration protocol of is as follows:

- In each iteration, the system selects j keywords and presents them to the user.
- The user provides relevance scores $r_i \in [0,1], i = 1,\ldots,j$ for the displayed keywords.

We assume that the relevance score r_i of a keyword k_i is a random variable with expected value $\mathbb{E}[r_i] = k_i \cdot w$, such that the expected relevance score is a linear function of the keywords. The unknown weight vector w is essentially the representation of the user's query and determines the relevance of keywords, i.e. the weight vector is the user model that describes user's interests and predicted directions based on the search history in the present session.

In order to preserve the dependency between the keywords and the documents, initially we represent each keyword k as a binary vector of length n, where n is the number of documents under consideration. The vector indicates the presence or absence of a keyword in a given document. In order to boost the significance of documents with rare keywords and to lower the significance of those with frequent keywords, we convert the binary vector representation of keywords into the *tf-idf* representation [24].

In order to help the user to explore the keyword space, we use LinRel [2], an algorithm that has already been proven to work well in other interactive retrieval systems [3]. At each iteration, LinRel suggests new keywords to be presented based on the feedback from the user obtained in previous iterations. The algorithm maintains a representation of the estimate $\hat{w}$ of the unknown weight vector w which maps keyword features to relevance scores. When selecting the next set of keywords to display, the system might simply select the keywords with the highest estimated relevance score. But since the estimate $\hat{w}$ may be inaccurate, this exploitative choice might be suboptimal. Alternatively, the system might exploratively select a keyword for which the user feedback improves the accuracy of the estimate $\hat{w}$, enabling better keyword selections in subsequent iterations.

In each iteration, LinRel obtains an estimate $\hat{w}$ by solving a linear regression problem. Suppose we have a matrix K, where each row k_i is a feature vector of keywords presented so far. Let $r = (r_1, r_2 ... r_p)^\top$ be the column vector of relevance scores received so far from the user, where p is a number of iterations. Thus, LinRel tries to estimate $\hat{w}$ by solving the linear regression problem $r = K \cdot w$. Based on $\hat{w}$, LinRel calculates an estimated relevance score $\hat{r_i} = k_i \cdot \hat{w}$ for each keyword k_i. In order to deal with the exploration-exploitation trade-off, we present keywords not with the highest score, but with the largest upper confidence bound for the relevance score. Thus, if σ_i is an upper bound on standard deviation of relevance estimate $\hat{r}_i$, the upper confidence bound of keyword i is calculated as $r_i + \gamma\sigma_i$, where $\gamma > 0$ is a constant used to adjust the confidence level of the upper confidence bound. In each iteration, LinRel calculates:

$$s_i = x_i \cdot (X^\top \cdot X + \lambda I)^{-1} X^\top.$$

and the keywords that maximize $s_i \cdot r + \frac{\alpha}{2}\|m_i\|$ are selected for presentation.

Document Set Diversification

After each iteration, the search engine returns a list of ranked documents to display. In order to diversify the set of the displayed documents and expose the user to more novel documents, we sample l of documents from the ranked list provided by the search engine and display them to the user. In order to obtain this, we use the *Dirichlet Sampling Algorithm* [9]. The algorithm maintains weights α_i for all the documents in the set returned by the search engine, and we sample l documents from this list by sampling the Dirichlet distribution. A fast method to sample a random vector f from a n-dimensional Dirichlet distribution with parameters $\alpha = \{\alpha_1, \ldots, \alpha_n\}$ is to draw n independent random samples from the Gamma distribution

$$f \sim Gamma(\alpha, 1) = \frac{f^{\alpha-1}e^{-f_i}}{\Gamma(\alpha)}$$

and normalise the resulting vector f. In our case, the input is a vector α representing the weights or probabilities of each document, while the vector f represents a random number associated with each document. We display to the user the top l documents with the highest values of f_i. Thus, the higher the value of α, the more likely it is that the corresponding document will be selected.

After each search iteration, we updated the weights α of the entire document set. However, the users do not provide explicit feedback on all the documents, so we need to update the document weights based on the keywords selected by the users. Thus, the documents are partitioned into two sets. One set consists of documents containing in their feature representation a keyword that received positive feedback from the user, while the other set contains all the remaining documents. The documents that received implicitly positive feedback have their weights α_i increased by 1. After the weight updates, the vector α is normalized so that in future iterations we have a better estimation as to which documents might be more relevant to the user's interests.

EVALUATION METHODS

We conducted a set of studies to evaluate our system. We first performed a simulation study to automatically evaluate the role of exploration in different parts of the system. Based on the results of this study we chose the final setup of the system that was evaluated in a comparative user study. The comparative study included two conditions: our system that is from now onwards referred as RLR (Reinforcement Learning Retrieval) system, and a baseline system. Before the user study, we conducted a pre-experiment study to ensure the usability of the RLR system. Then, we ran the actual user study in order to evaluate the retrieval performance of the RLR system compared to the baseline system.

Data

We used a dataset including over 50 million scientific documents from the following data sources: the Web of Science prepared by THOMSON REUTERS, Inc., the Digital Library of the Association of Computing Machinery (ACM), the Digital Library of Institute of Electrical and Electronics Engineers (IEEE), and the Digital Library of Springer. The information about each document consists of: title, abstract, author names, publication year and publication forum. Both the baseline and RLR systems used the same document set.

Simulations

The RL techniques are incorporated into two modules of the RLR system: the Keyword Exploration module and the Document Diversification module. Thus, there were three possible configurations of the system: (1) RL in both the keywords and the articles space; (2) RL only in the keyword space; (3) RL only in article space. The fourth possibility, i.e. RL in neither of the components is equivalent to using traditional query-based informational retrieval systems and we tested this setting in the user studies. However, running user studies involving all four settings would have required a very large number of participants with similar research backgrounds, and so in order to assess the effectiveness of the RL methods, we first ran a set of simulations to define the system configurations to be used in the user studies. The purpose of the simulations was to identify which of the three settings would provide the user with the most diverse search results.

First, in a modelled situation of algorithm usage we estimated the optimal values of parameters of the algorithms. Next, we collected 10 examples of user's actions, i.e. keyword manipulations over ten iterations, while performing different searches. Each of the three system settings was run ten times using the collected user actions, i.e. for each setting we collected ten hypothetical searches with ten iterations each. We calculated the fraction of new articles appearing at each iteration in the studied systems. The setting with exploration in both keywords and documents gave us at least 20% more articles on average than other settings and this is the setting we applied in the users studies.

Pre-Experiment Study

The target of the Pre-Experiment study was to measure the general usability of the RLR system and the subjective usability of the search directing support of the RLR system. We used a task-based information retrieval evaluation method [16]: The participants performed search tasks with the system. In each task, the participant was asked to imagine that she is writing an essay on a given topic, and her task was to search and collect documents that she finds useful for the topic specified in the task description. Two tasks were defined and their topics were "semantic search" and "reinforcement learning". In order to preserve the exploratory nature of the task, we only recruited participants who have little knowledge in the topics of the tasks. We recruited ten participants who each performed one of the tasks using the RLR system. Once the participants completed the tasks, they filled two questionnaires to provide their subjective usability assessment on the system. As usability assessment questionnaires we used the standard System Usability Scale (SUS) [6] and a questionnaire based on user-centric evaluation framework named ResQue designed for the evaluation of recommender systems [21]. We used SUS because it is the most used, technology

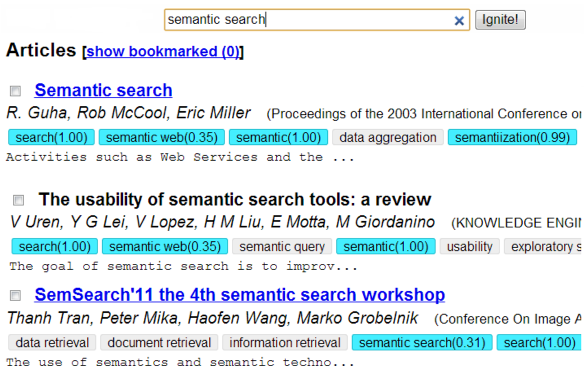

Figure 4. Screen shot of baseline system

independent questionnaire for measuring perceptions of usability. ResQue was chosen because it specifically measures the interaction adequacy and preference expression capabilities offered by a system. Participants used a five-point Likert scale to provide their answers for the ResQue questionnaire. In addition to the questionnaires we also conducted post task interviews with these participants to assess their subjective feedback on the characteristic features of the RLR system.

Baseline System

We used as the baseline a traditional query–based alternative, where neither user modeling, exploratory view nor diversification was used. The document retrieval algorithm was the same in both systems. In the baseline, users can express their information need only through typing queries and the results are presented as a list of documents followed by keywords in each document. Unlike in the RLR system, in the baseline system users cannot interact with the keywords visualized underneath each document to provide feedback; these keywords can only be used as clues for formulating queries. Figure 4 shows a screen shot of the baseline search user interface.

User Experiment

We conducted user experiments to compare the baseline and the RLR system. The aim of the experiments was (1) to evaluate the ability of the RLR system to support users in directing their search in an exploratory search setting, and (2) to measure the retrieval performance of the compared systems. We used a task-based information retrieval evaluation, where participants were situated in a scientific writing scenario. The participants were asked to imagine that they are writing an essay on a given topic and to answer a set of questions related to the topic. The participants used a given system to seek the information. The study used a between–subjects design in which half of the participants performed the task using RLR and the other half using the baseline system. We also counterbalanced between the tasks.

Tasks

We recruited two post-doctoral researchers as experts from the domains of information retrieval and machine learning to define the tasks. The task fields chosen by the experts were "semantic search" and "robotics". The experts wrote task descriptions using the following template: "imagine that you are writing a scientific essay on the topic of semantic search/robotics. Search scientific documents that you find useful for this essay". In order to provide clear goals for exploration, the experts provided the following guidelines on what content should be included in the essays:

- Robotics - This essay should include sub-fields, application areas and algorithms commonly used in the field of "robotics".
- Semantic search - This essay should include techniques used to acquire semantics, methods used in practical implementation and organization of results in semantic search.

Participants and Procedure

We recruited 20 researchers from our university to participate in this study. All the participants were either research staff or students (11 PhD researchers and 9 Masters students), and all had backgrounds in computer science or related fields. We particularly selected researchers as participants because our current database only consists of scientific documents. Therefore we needed participants who have experience in scientific document search. Prior to the experiment, we conducted a background survey of the participants to ensure that they have conducted literature search before and they are not expert researchers in the topics of the search tasks so that their prior knowledge will not influence the exploratory nature of the tasks. Prior to the study we provided training for each participant with the system that they are going to use in the experiment. We limited the time available to complete the task to 30 minutes to make sure that the participants were actively searching for the information during the experiment and had equal time to complete the task.

Data Logging

When the participants were performing tasks, we logged all their interactions with the system and all the articles and keywords presented by the systems in response to these interactions. Data logged from the interactions with the RLR system included details of the articles displayed, keywords predicted by the system, manipulated keywords, queries typed by the participants and times at which each interaction occurred. Similarly, from the baseline system we logged the queries typed, articles displayed, articles bookmarked by the participants and the times at which participants entered queries.

Ground Truth and Relevance Assessments

We pooled all the documents displayed to the participants in either condition, i.e. based on the logs we extracted all documents that were found by any user using either the baseline or the RLR system. Next, we asked the domain experts who defined the tasks to provide relevance assessments on these documents. In order to avoid any biasing of the expert assessments, the assessment process was double-blinded, meaning that the experts did not know from which of the two conditions the document originated nor for which of the participants the document was viewed. The experts provided relevance assessments on three levels: (1) relevance - whether this article was relevant to the search topic; (2) obviousness - whether this article is well known in a given research area; and (3) novelty - whether a given article was uncommon, yet showing a new aspect of the topic. Experts reviewed every article under each category and rated them as 1 if they belong to the category and 0 otherwise. This provided the ground truth that was used to evaluate the retrieval performance of the systems. To measure the quality of the relevance assessments, part of the assessments were conducted by two experts. We run the Cohen Kappa test to measure the inter-annotator agreement between the experts. Kappa indicated a substantial agreement (Kappa = 0.71, $p < 0.001$).

Retrieval Performance Measures

We defined three measures based on precision, recall and F measures of information retrieval to measure the performance of the two systems. We could not use these measures directly because in our setting users are shown a limited set of documents, i.e. top 50, on each iteration instead of the total ordering and the users perform several, but varying number of queries and interactions over time during a search session. The measures capture the performance of the system as a function of time when used to seek answers for the task. For example, if a user finds relevant documents on the first iteration, just ten seconds after starting to use the system, but the system does not assist the user to direct the search, then the performance may not be much better when investigated after 120 seconds of use. On the other hand, if the system assists the user to gain more relevant documents, the recall after 120 seconds may have been increased because the user could easily interact with the system to gain more relevant results.

We generalized the precision, recall and F measures to take into account the temporal dimension, where users cumulatively gain new information while searching. We start with a definition of cumulative document sets:

$$\cup_t = \bigcup_0^t documents,$$

which represents the cumulative set of documents being presented for the user, i.e. the union of documents from the start time 0 until time t. The rationale is that if the same documents are presented for the user in different moments of time, they are only added once in the $\cup_t$. We define temporal precision as:

$$P_t = \frac{\cup_t(relevant)}{\cup_t(retrieved)}.$$

It measures the proportion of relevant documents cumulatively shown to the user, compared to all documents cumulatively shown to the user until time point t. Similarly, the the temporal recall is defined as:

$$R_t = \frac{\cup_t(relevant)}{all\ relevant},$$

which measures the proportion of relevant documents cumulatively shown for the user compared to all relevant documents in the system.

The temporal recall is expected to grow over time as users are interacting with the system and viewing more documents, and the temporal precision is expected to stay the same or increase when users direct their search to more specific or more general, yet relevant documents. The best way to evaluate a system is to combine these measures to investigate how well the system is able to balance between precision and recall. The temporal F-measure is defined as:

$$F_t = \frac{2P_tR_t}{P_t + R_t},$$

which is the harmonic mean of R_t and P_t up to the time point t. These measures allow us to investigate how the precision–recall tradeoff develops over time and compare systems in task settings where users may use different queries at different points of time to find broad sets of articles. For example, by setting $t = 60$ seconds, we could investigate how many relevant articles a user was able to collect during the first minute of use by measuring R_t and what proportion of the collected articles were relevant by measuring P_t.

Aspect	Question	Mean (m) and Standard deviation(std)
Preference Elicitation	**Provides adequate way to express preferences**	**m=3.93 std=0.583**
Preference Revision	**Provides adequate support to revise preferences**	**m=3.70 std=0.952**
Interface Adequacy	**layout of the system is clear**	**m=4.48 std=0.750**
Interaction Adequacy	**Offered me useful interaction options to express my information need**	**m=3.73 std=0.828**
Ease of Decision making	**I quickly became productive by using the system**	**m=3.60 std=0.894**

Table 1. Summary of Results from the questionnaire based on ResQue framework

EXPERIMENTAL RESULTS

In order to measure the usability of the RLR system, we evaluated the feedback provided by the participants of the Pre-Experiment study. The RLR system received an average score of 79.5 out of 100 (standard deviation = 15.36) in the SUS questionnaire which shows that RLR is favored by the participants. Another questionnaire, based on the ResQue framework, consisted of questions addressing five different aspects of recommender systems. Table 1 presents a summary of the results, which show that the interface of the RLR system provides adequate support for the users to express and revise their preferences. Users also perceived the layout and the new interaction options of the RLR system to be clear. Finally, the users also indicated that with this system they performed more productively than before.

The post-task interviews with the participants also provided favorable results for the RLR system. The feature that received positive feedback from the majority of subjects was interactive keyword visualization, both in the exploratory view and underneath each article. Some comments from the participants were: "suggesting keywords make the system very easy to use and identify related keywords that I didn't know", "visual search is awesome, seeing the centrality of the keywords for the search from the circle", "I like the option of seeing keywords and both (de-)selecting and weighting them", "the rich keyword selection option provides new cues, even for the well known topics", "keyword visualization is very intuitive". Some of the participants commented that even though they like to see new keywords appearing after each interaction they prefer to have a back button to return back to the previously displayed keywords because sometimes keywords change too fast: "I would like to have a back button to go to previous results, so that I can try with new keywords and need not to worry about missing previous research results", "keyword view is changing too fast". Some of the participants liked the idea of performing fewer query alterations and providing more visual feedback using the mouse: "The fact that I could use the system only by using the mouse made it easy. So there was no need to constantly re-type the query". However, some of the participants found it difficult to judge documents without the citation count: "I am more used to judge articles based on citation data".

User Experiments Results

Table 2 shows the general system performance results from the user studies. In all the assessed categories, i.e. relevance, novelty and obviousness, the RLR system outperforms the baseline in terms of temporal F-measure, i.e. the measure of the quality of documents obtained throughout a search session. Temporal recall in the RLR system outperforms the baseline for all categories, while temporal precision in the RLR setting is greater only for novelty and non-obviousness. For temporal precision, the difference between the RLR system and the baseline was only 3% (69% against 72%). The RLR system beats the baseline in temporal F-measure due to the fact that it provides much better temporal recall while not sacrificing temporal precision.

The last row in Table 2 shows the numbers of relevant, novel, and obvious documents, as judged by the experts. These counts are derived from all the articles retrieved during all search sessions. Despite each experiments lasting the same amount of time, the RLR system provided the users with access to a much larger set of documents as users performed significantly more actions compared with the baseline system. With respect to *obviousness*, the number of articles displayed by both systems is not significantly different. Thus, the study indicates that both systems provide users with access to the most common, i.e. obvious, documents in a given research field, however, the RLR system exposes users to a higher number of relevant and novel articles that could not be easily found with a simple query-based search.

	Relevance		Novelty		Obviousness	
System	RLR	BL	RLR	BL	RLR	BL
F-measure	0.25	0.15	0.18	0.09	0.22	0.20
Precision	0.69	0.72	0.40	0.33	0.26	0.34
Recall	0.15	0.09	0.12	0.05	0.20	0.17
♯ of Articles	882	522	570	228	253	223

Table 2. Comparison of the performance of the RLR system and the baseline system (BL).

Search Performance

Figure 5 presents all the measures with respect to *relevance*. Figure 5a shows how temporal precision measure evolves as the search progresses. The RLR system is mildly outperformed by the baseline with the gap between systems shrinking over time. At the same time, the RLR system's temporal recall increases much faster than the baseline after 200 seconds and at the end of the search session reaches a much higher value (Figure 5b). The temporal F-measure shows how good the system is in balancing between these measures by taking their harmonic mean. The RLR's impressive temporal recall and similar performance with respect to temporal precision explains why the system outperforms the baseline in temporal F-measure after 200 seconds. We would like to stress the fact that within the first 200 seconds, the users on average preformed only one search iteration using the RLR system and spent a lot of time looking at the presented documents and keywords. Thus, the users do not need to wait for over 3 minutes for the RL techniques to take effect since the user model is instantiated right after the first search iteration.

An interesting finding is that in the first few minutes, the baseline performs slightly better than the RLR system. The same effect can be seen in Figures 6a and 6b with temporal F-measure with respect to *novelty* and *obviousness*. The baseline initially outperforming the RLR system with respect to *obviousness* is illustrated in Figure 6b. As the search begins, the baseline outperforms the RLR system, however, after a few minutes the average temporal F-measure of the baseline drops below that of the RLR system. A possible explanation is that at the beginning of the search, the users working with the baseline can easily create queries resulting in a large number of frequent or obvious documents. As the search progresses and the users need to think of more specific queries, temporal recall of relevant or obvious documents drops. However, when reinforcement learning is employed to build a user model, the user is able to direct the search more efficiently while at the same time preserving the search context. This helps users to preserve the precision of documents, but at the same time gain relevant documents faster than the baseline. We attribute this to the combination of user modeling and document diversification that allows users to better interact with the system and obtain a wider set of documents.

User Interaction

Users of the RLR system performed significantly more iterations (on average 14.7) than users of the baseline (on average 8) within the same time restrictions, which indicates that the RLR system allows the user to direct the search more easily through the displayed keywords, resulting in a high rate of interactions as the search progresses. As Figure 6c shows, at each iteration the system recommends new keywords to the user, which the user manipulates. The graph indicates that the keywords displayed and manipulated over time in the RLR system indeed support the users in directing their search and the users do take advantage of this opportunity.

SUMMARY AND CONCLUSIONS

To support users in directing the exploration of the information space during search, we developed an interactive infor-

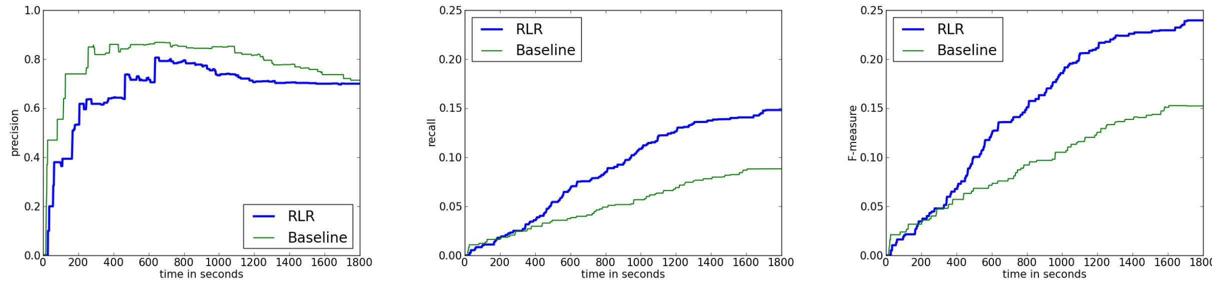

Figure 5. Average measures of relevance: temporal precision, recall and F-measure of the baseline and the RLR systems over a search session. The RLR system sacrifices little precision to get significantly higher recall, and outperforms the baseline in temporal F-measure after 200 seconds.

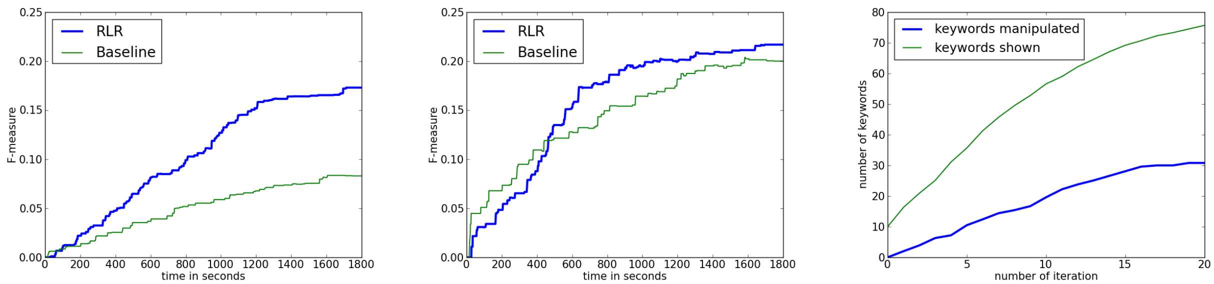

Figure 6. Average F-measure of the RLR system and the baseline system over the search session with respect to (a) novelty and (b) obviousness; (c) average cumulative number of keywords displayed and manipulated with the RLR system over the search session.

mation retrieval system that applies RL to user feedback on keywords. Besides writing queries, users can explore the information space by manipulating keywords on a display scoring them for interest. Such interactions result in predictions of new keywords and documents matching the user interest at the current iteration of retrieval. To achieve this, we developed three components: (1) Information Retrieval and Ranking, (2) Keywords Exploration, and (3) Document Diversification. The Information Retrieval and Ranking component allows fast ranking given the user model where only the top-ranked documents and document features are used as the context for the more advanced learning. The explicit user feedback is sent to Keywords Exploration that produces a list of keywords with weights which are passed on to the Information Retrieval and Ranking module as a user model estimation. The Document Diversification module is responsible for determining the set and order of documents that are passed on to the document list shown to the user. Our interactive information retrieval system is novel in the way that RL is applied not to feedback on documents but on higher-level features such as keywords. We carried out simulations to verify that RL should be applied to both Keywords Exploration and Document Diversification. A pre-experiment study confirmed that the system was functional and that users found it usable for exploration during search. Finally, we designed and carried out a task-based experiment comparing our system with a baseline query based system that did not offer user modeling, exploratory view or diversification. To be able to compare the performance of the two systems, we re-defined precision, recall and F-measures to take into account the fact that users perform several, and varying amount of different queries and interactions over time during a search session.

In all the categories assessed during the study, i.e. relevance, novelty and obviousness, our system outperforms the baseline in terms of the temporal F-measure, that is a measure of the quality of documents obtained through the search session. Analyzing differences in recall and precision in different categories, the results show that the RLR system provides much better temporal recall while not sacrificing the temporal precision, keeping it approximately the same or a bit better. The RLR system exposes users to a higher number of relevant and novel articles that could not be easily found with a simple query-based search system, but still gives access to the common ones. We further analyzed the temporal performance and user interaction finding evidence of how RLR supports users in directing search. When analyzing the performance over time, the F-measure for obviousness is better at the beginning for the baseline system while it becomes better for RLR only when RL is used to build a user model and the user is able to direct the search more efficiently. More importantly, the ability of users to direct search is emerging from the fact that users do score positively keywords recommended by RLR.

ACKNOWLEDGMENTS

The data used in the experiments is derived from the Web of Science prepared by THOMSON REUTERS, Inc.; the Digital Library of the Association of Computing Machinery (ACM); the Digital Library of Institute of Electrical and Electronics Engineers (IEEE), and the Digital Library of Springer. D. Głowacka and S. Kaski acknowledge the support of The Academy of Finland (Finnish Centre of Excellence in Computational Inference Research COIN, 251170).

REFERENCES

1. Alonso, O., Baeza-Yates, R., and Gertz, M. Exploratory search using timelines. In *SIGCHI 2007 Workshop on Exploratory Search and HCI Workshop*, no. 1 (2007).
2. Auer, P. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research 3* (2002), 397 – 422.
3. Auer, P., Hussain, Z., Kaski, S., A.Klami, J.Kujala, Laaksonen, J., Leung, A., Pasupa, K., and Shawe-Taylor, J. Pinview: Implicit feedback in content-based image retrieval. In *JMLR Workshop and Conference Proceedings: Workshop on Application of Pattern Analysis*, vol. 11 (2010), 51–57.
4. Balabanović, M. Exploring versus exploiting when learning user models for text recommendation. *User Modeling and User-Adapted Interaction 8* (1998), 71–102.
5. Baldonado, M., and Winograd, T. Sensemaker: an information-exploration interface supporting the contextual evolution of a user's interests. In *Proc. of SIGCHI* (1997), 11–18.
6. Brooke, J. Sus-a quick and dirty usability scale. *Usability Evaluation in Industry 189* (1996), 194.
7. Chapelle, O., and Li, L. An empirical evaluation of thompson sampling. In *Proc. of NIPS*, vol. 25 (2011).
8. Chau, D., Kittur, A., Hong, J., and Faloutsos, C. Apolo: making sense of large network data by combining rich user interaction and machine learning. In *Proc. of SIGCHI* (2011), 167–176.
9. Głowacka, D., and Shawe-Taylor, J. Content-based image retrieval with multinomial relevance feedback. In *JMLR Workshop and Conference Proceedings: 2nd Asian Conference on Machine Learning*, vol. 13 (2010), 111–125.
10. Havre, S., Hetzler, E., Perrine, K., Jurrus, E., and Miller, N. Interactive visualization of multiple query results. In *Proceedings of the IEEE Symposium on Information Visualization* (2001), 105.
11. Hutchins, E. L., Hollan, J. D., and Norman, D. A. Direct manipulation interfaces. *Hum.-Comput. Interact. 1*, 4 (Dec. 1985), 311–338.
12. Joachims, T., Freitag, D., and Mitchell, T. Webwatcher: A tour guide for the world wide web. In *Proc. of IJCAI* (1997).
13. Käki, M. Findex: search result categories help users when document ranking fails. In *Proc. of SIGCHI* (2005), 131–140.
14. Kelly, D., and Fu, X. Elicitation of term relevance feedback: an investigation of term source and context. In *Proc. of SIGIR* (2006), 453–460.
15. Kules, B., Wilson, M., Schraefel, M., and Shneiderman, B. From keyword search to exploration: How result visualization aids discovery on the web. *Human-Computer Interaction Lab Technical Report HCIL-2008-06, University of Maryland* (2008).
16. Leide, J., Cole, C., Beheshti, J., Large, A., and Lin, Y. Task-based information retrieval: Structuring undergraduate history essays for better course evaluation using essay-type visualizations. *Journal of the American Society for Information Science and Technology 58*, 9 (2007), 1227–1241.
17. Li, L., Chu, W., Langford, J., and Schapire, R. A contextual-bandit approach to personalized news article recommendation. In *Proc. of WWW* (2010).
18. Marchionini, G. Exploratory search: from finding to understanding. *Commun. ACM 49*, 4 (Apr. 2006), 41–46.
19. Matejka, J., Grossman, T., and Fitzmaurice, G. Citeology: visualizing paper genealogy. In *Proceedings of Human Factors in Computing Systems Extended Abstracts* (2012), 181–190.
20. Pandey, S., Agarwal, D., Chakrabarti, D., and Josifovski, V. Bandits for taxonomies: A modelbased approach. In *Proc. of ICDM* (2007).
21. Pu, P., Chen, L., and Hu, R. A user-centric evaluation framework for recommender systems. In *Proc. of RecSys* (2011), 157–164.
22. Radlinski, F., Kleinberg, R., and Joachims, T. Learning diverse rankings with multi-armed bandits. In *Proc. of ICML* (2008), 784–791.
23. Seo, Y., and Zhang, B. A reinforcement learning agent for personalized information filtering. In *Proc. of IUI* (2000), 248–251.
24. Spärck-Jones, K. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation 28*, 1 (1972), 11–21.
25. Spink, A., Wolfram, D., Jansen, M. B. J., and Saracevic, T. Searching the web: The public and their queries. *Journal of the American Society for Information Science and Technology 52* (2001), 226–234.
26. Teevan, J., Alvarado, C., Ackerman, M. S., and Karger, D. R. The perfect search engine is not enough: a study of orienteering behavior in directed search. In *Proc. of SIGCHI* (2004), 415–422.
27. Terveen, L., Hill, W., and Amento, B. Constructing, organizing, and visualizing collections of topically related web resources. *ACM Transactions on Computer-Human Interaction (TOCHI) 6*, 1 (1999), 67–94.
28. Tombros, A., and Sanderson, M. Advantages of query biased summaries in information retrieval. In *Proc. of SIGIR* (1998), 2–10.
29. White, R. W., Bennett, P. N., and Dumais, S. T. Predicting short-term interests using activity-based search context. In *Proc. of CIKM* (2010).
30. Yee, K.-P., Swearingen, K., Li, K., and Hearst, M. Faceted metadata for image search and browsing. In *Proc. of SIGCHI* (2003), 401–408.
31. Yue, Y., and Joachims, T. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proc. of ICML* (2009).
32. Zhai, C., and Lafferty, J. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst. 22*, 2 (Apr. 2004), 179–214.
33. Zhang, B., and Seo, Y. Personalized web-document filtering using reinforcement learning. *Applied Artificial Intelligence 15* (2001), 665–685.
34. Zhang, Y., Xu, W., and Callan, J. Exploration and exploitation in adaptive filtering based on bayesian active learning. In *Proc. of ICML* (2003), 896–903.