Improving Controllability and Predictability of Interactive Recommendation Interfaces for Exploratory Search

Antti Kangasrääsiö
Helsinki Institute for Information Technology HIIT
Dept. of Computer Science
Aalto University
antti.kangasraasio@hiit.fi

Dorota Głowacka
Helsinki Institute for Information Technology HIIT
Dept. of Computer Science
University of Helsinki
dorota.glowacka@hiit.fi

Samuel Kaski
Helsinki Institute for Information Technology HIIT
Dept. of Computer Science
Aalto University;
Dept. of Computer Science
University of Helsinki
samuel.kaski@hiit.fi

ABSTRACT

In exploratory search, when a user directs a search engine using uncertain relevance feedback, usability problems regarding controllability and predictability may arise. One problem is that the user is often modelled as a passive source of relevance information, instead of an active entity trying to steer the system based on evolving information needs. This may cause the user to feel that the response of the system is inconsistent with her steering. Another problem arises due to the sheer size and complexity of the information space, and hence of the system, as it may be difficult for the user to anticipate the consequences of her actions in this complex environment. These problems can be mitigated by interpreting the user's actions as setting a goal for an optimization problem regarding the system state, instead of passive relevance feedback, and by allowing the user to see the predicted effects of an action before committing to it. In this paper, we present an implementation of these improvements in a visual user-controllable search interface. A user study involving exploratory search for scientific literature gives some indication on improvements in task performance, usability, perceived usefulness and user acceptance.

Author Keywords

Controllability; exploratory search; information retrieval; interactive user modelling; predictability; probabilistic user models; user interfaces

ACM Classification Keywords

H.3.3 Information Search and Retrieval: Relevance feedback; H.5.2 Information Interfaces and Presentation: User Interfaces

IUI 2015, March 29–April 1, 2015, Atlanta, GA, USA.

http://dx.doi.org/10.1145/2678025.2701371

INTRODUCTION

Traditionally, research in information retrieval systems focuses on improving the predictive accuracy of recommendations mostly by developing new algorithms. Recent studies [9, 13] have shown that visual features and enhanced interaction can greatly improve the user engagement in the search process, and consequently also the performance of the retrieval system. Some of the interface characteristics studied are transparency, explainability, predictability and controllability, of which user controllability is the least explored aspect of information retrieval systems.

In this paper, we concentrate on user controllability and system predictability in the context of exploratory search. In exploratory search, the user searches for information in a domain that she is not initially familiar with. Because of this, search interfaces assisting the user in exploratory search are faced with a difficult problem: how to help the user direct the exploratory search with uncertain feedback. If the user were an expert and the feedback certain, it could be interpreted in a purely exploitative manner. However, since this is not the case in exploratory search, there needs to be a suitable amount of exploration mixed in to help the user with the search task.

Probabilistic user models can be used to handle the exploration/exploitation trade-off, but there can be usability problems if the user feedback is interpreted simply as datapoints to fit a model. As the user is not a passive function that is sampled by the system, but an active entity that is trying to steer the system through iteratively improving the model, there needs to be a layer of interpretation between the user and the system. In this work we propose a layer which translates the user input into requirements for the state of the system, and makes the system better predictable by showing the user on-line estimates of the effects her actions will have on the system.

A user study gives some indication that by implementing this layer of interpretation, users are able to perform better in focused exploratory information retrieval tasks, they are able to better predict the consequences of their actions, and to understand the connections between different feedback items in the search interface. There is also some evidence of improvement in the usability and perceived usefulness of the search system.

RELATED WORK

In recent years, exploratory search has attracted attention from, among others, information retrieval (IR), Human Computer Interaction (HCI), and Machine Learning (ML) communities, which have proposed several techniques and systems to facilitate exploratory search.

Some of the initial solutions include result clustering [4], relevance feedback [7], query suggestion [1], and faceted search [15]. However, the proposed techniques are rarely used in practice due to the high cognitive load of going through a large list of suggestions or providing feedback for a large number of items [7].

There have also been numerous attempts to engage the user into the feedback loop through interactive visualizations combined with learning algorithms to support users to comprehend the search results [3], and visualization and summaries of results [8]. These solutions give users more control; however, they do not adapt to the evolving information needs of the user [12]. Recently, reinforcement learning (RL) techniques have been applied to facilitate exploratory search [5, 6, 11]. The RL-based systems prevent the user from getting trapped in a local context and expose the user to a larger area of the information space. However, they do not allow the user to anticipate the effects of their actions.

One approach to improve the controllability of recommendation interfaces is by allowing the user to interactively combine different recommendation algorithms using weights [9]. We deal with the same problem through improving the user's control over a single algorithm.

PROPOSED APPROACHES

User Feedback as a Goal for an Optimization Problem

The user feedback will have the effects intended by the user, if the feedback values are interpreted as the goal for an optimization problem regarding the next state of the system, instead of just additional datapoints to fit a model. This way the user in effect has an automatic assistant that steers the system towards the desired target indicated by the user. In this modelling approach the user is assumed to be an active entity trying to steer the system, instead of a passive entity that is sampled by the system, as is usual in reinforcement learning approaches. In general, we formalize the problem of interfacing users with reinforcement learning based systems as follows: "What feedback values should be given to the model update algorithm so that the resulting model fulfills certain optimality criterion, which is parametrized by the feedback values given by the user?"

Solving this problem requires making three design choices: what optimality criterion to use, from what space the optimal feedbacks are sought and what algorithm to use for solving the optimization problem.

Predicting Effects of Feedback Actions

When the user is giving feedback, it is not obvious that she will be able to predict the effects these actions will have on the system, given that the behavior of the system relies on a potentially complex underlying user model. If making the user model simpler is not possible without sacrificing performance, one way to solve this problem is to enable the user to see a prediction of the effects of any action available to her before she commits to it. This way the user will be less surprised by the effects of the action and is able to choose the action based on the expected consequences.

However, there are some practical problems with accurately predicting the effects of different actions. For example, the system may be so complex that accurately simulating and visualizing the effects for any possible action is infeasible in real time. Further problems may arise if the system has randomized elements in it, for example in order to support exploration, because in this situation the number of possible future states may be practically infinite.

It is thus more practical to use approximate prediction. The interface should visualize the probable effects of the action, while still being computationally feasible in real time. Our solution for the approximate prediction is to sample the possible future states of the system, fit a simpler function approximation to these samples and use it for visualizing the effects.

Constructing this approximation requires making two main choices: which points to sample and what function family to use for constructing the approximation based on these samples.

SYSTEM ARCHITECTURE

Introduction to the system

We use an existing search engine, the SciNet [5], as a case study and implement these new approaches in it. SciNet allows the user to direct the search by interacting with a visualization of the search intent model of the user, composed of keywords and their estimated relevances. The model is visualized to the user using an Intent Radar, where the top 10 relevant keywords are shown on a circular layout so that the closer to the center a keyword is, the more relevant it is. Additional keyword suggestions are also visualized on the edge of the radar view. An example of the visualized user model is shown in Figure 1, along with a scenario illustrating one of the problems with the baseline system, resulting from the user feedback being interpreted only as additional datapoints.

The user gives feedback on the user model by moving one keyword at a time to a new location on the radar. When the user lets go of the keyword, feedback for this keyword is calculated based on the distance from the center. The user model is then updated, new articles are retrieved, and the new state is visualized to the user.

Implementation of the Proposed Approaches

We implemented the optimization of the user model as a linear greedy incremental search, illustrated in Algorithm 1. This algorithm was motivated by the underlying user model being approximately linear. At each step, first the maximal and minimal relevance feedback values (r_{max}, r_{min}) for the keyword (k) the user moved are used to tentatively update the current model ($\mathbf{M}_{\mathrm{t+1}}$), to get the estimates of the corresponding error values ($\mathrm{e}_{max}, \mathrm{e}_{min}$). Then, the next approximation for optimal relevance feedback (r') is chosen by linear

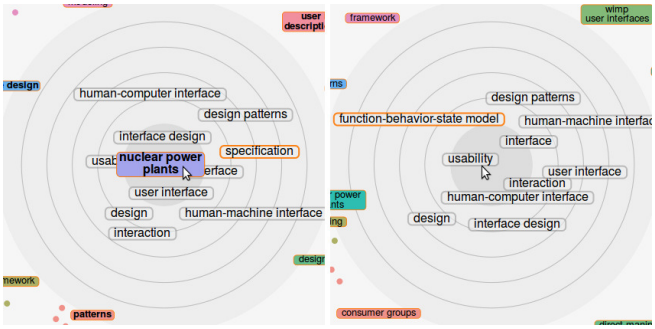

Figure 1. In the baseline system, sometimes even giving maximal relevance feedback to a keyword by dragging it to the center of the Intent Radar (left) may result in a new model that does not contain this keyword within the top 10 keywords (right). The central keywords represent the top 10 keywords.

interpolation, or at the maximum values. This value is then used to update the current model, and the relevance value is appended to a list containing the relevance values used to update the model (**r'**). The algorithm terminates either when the model is close enough (ϵ) to the optimum ($|\tilde{o}(\cdot)|$), or when a maximum number of iterations (N_{max}) has been reached.

We implemented the approximate prediction as a per-keyword linear interpolation. The idea of the algorithm is to calculate two possible user models using the extreme feedback values (r_{min}, r_{max}) for the keyword the user is dragging, and to use a linear interpolation as the predicted user model for any relevance value between the extremes. As the user is dragging any single keyword over the radar, the (other) central keywords will be moved on the radar, in the radial direction, to the place corresponding to the predicted new relevance value, as illustrated in Figure 2.

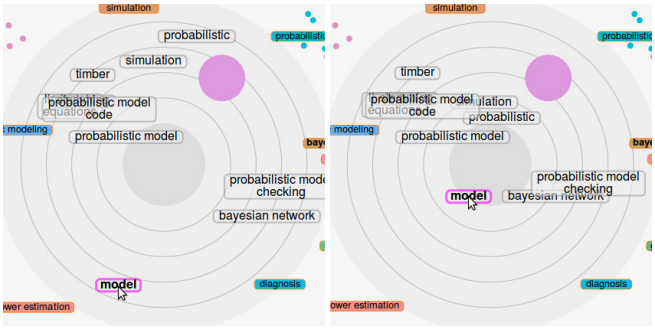

Figure 2. In the improved system, while the user is dragging a keyword over the Intent Radar, the locations of (other) central keywords move simultaneously according to the prediction model. The purple dot indicates the original location of the keyword, and the user can move the keyword back there to cancel the prediction for the current keyword if she wants to try out another one.

EVALUATION

We conducted a preliminary user study on 12 naïve university students and staff members. Each user performed two search tasks, one using the search engine without the proposed improvements (baseline system) and another where the improvements were implemented (improved system). One

Input: user model $\mathbf{M}_t$, keyword k, relevance feedback r, model update rule $u(\cdot)$, accuracy ϵ, maximum number of iterations N_{max}, limits for relevance values (r_{max}, r_{min})
Output: new user model $\mathbf{M}_{t+1}$, relevance feedback values **r'**

$\tilde{o}(\mathbf{M}) = (\text{relevance value of keyword k in } \mathbf{M}) - r$
$\mathbf{M}_{t+1} \leftarrow \mathbf{M}_t$
r' $\leftarrow$ empty list
while $|\tilde{o}(\boldsymbol{M}_{t+1})| > \epsilon$ **do**
  $e_{max} \leftarrow \tilde{o}(u[\mathbf{M}_{t+1}, k, (\mathbf{r'}, r_{max})])$
  $e_{min} \leftarrow \tilde{o}(u[\mathbf{M}_{t+1}, k, (\mathbf{r'}, r_{min})])$
  if $e_{max} e_{min} < 0$ **then**
    $r' \leftarrow \dfrac{r_{min} e_{max} - r_{max} e_{min}}{e_{max} - e_{min}}$
  else
    if $|e_{max}| < |e_{min}|$ **then**
      $r' \leftarrow r_{max}$
    else
      $r' \leftarrow r_{min}$
    end
  end
  $\mathbf{M}_{t+1} \leftarrow u(\mathbf{M}_{t+1}, k, r')$
  r' $\leftarrow (\mathbf{r'}, r')$
  if number of iterations $\geq N_{max}$ **then**
    break
  end
end

Algorithm 1: Our implementation of user model state optimization. The $\mathbf{M}_t$ is (the data structure representing) the user model at time t. The pair (k, r) is the feedback, indicating that keyword k has relevance r for the user's search intent. The model update rule $u(\cdot)$ takes a model, a keyword and one or more relevance feedback values for that keyword, and returns an updated model. The r' is a list containing the optimal relevance values (length less than or equal to N_{max}). The accuracy ϵ defines the limit for acceptable error in the optimality criterion $o(\cdot) = |\tilde{o}(\cdot)|$. The r_{max} and r_{min} are the maximum and minimum relevance values the user can give, e.g. 1 and 0. The result of appending value b to list a is denoted by (a, b).

task was a focused exploratory task and the other a broad exploratory task. Both tasks were about fact retrieval regarding topics the users were not very familiar with. Familiarity in the topic was rated in 1 to 5 Likert scale and all the users reported familiarity less than 5, with average rating of 2.0. In the broad task the questions had multiple correct answers, whereas in the focused task the question scopes were more narrow. The study was balanced with respect to the combination of the type of interface, task and order. The users answered a standard SUS [2] and a modified 15-question ResQue [10] questionnaire after each task. After both tasks a short semi-structured interview was conducted.

The answers to the task questions were rated in a double-blind manner by an expert in both fields. The grading was done in a 1 to 5 Likert scale per question where 5 corresponded to an excellent answer and 1 to a completely wrong answer. One third of the answers were rated separately by another expert. The average inter-rater reliability based on Spearman rho was 0.75, which can be considered adequate. Also the keywords and articles viewed by the users were rated for rel-

evancy by an expert. P-values for the results were calculated using the two-sided Wilcoxon rank-sum algorithm. The interviews were analyzed using content analysis [14].

RESULTS

Based on the results we deemed two users as outliers because on one task they received the lowest possible points on task performance and over 80 % of the articles they viewed were rated irrelevant. It was likely that these tasks were too difficult for these users. These two users were excluded from the analysis.

The improved system resulted in better performance in the focused exploratory task (3.1 for improved, 2.2 for baseline) and worse performance in the broad exploratory task (3.0 for improved, 3.8 for baseline), but these differences were not statistically significant ($p = 0.2$ and $p = 0.1$ respectively). However, the difference of task performances for the baseline between the two tasks was significant ($p = 0.01$). This indicates that the baseline is more efficient in broad than focused exploratory tasks. For the improved system this difference did not exist ($p = 0.6$).

The improved system had slightly better SUS score (64.5 for improved, 62.8 for baseline) and ResQue score (36.0 for improved, 32.7 for baseline). However, the differences were not statistically significant ($p = 0.8$ and $p = 0.7$ respectively). The per-question scores for SUS are shown in Table 1 and for ResQue in Table 2.

I	B	Question
3	3	I think that I would like to use this system frequently
2.2	**1.9**	I found the system unnecessarily complex
3.9	3.3	I thought the system was easy to use
1.6	1.6	I think that I would need the support of a technical person to be able to use this system
3.3	3	I found the various functions in this system were well integrated
2.8	**2.7**	I thought there was too much inconsistency in this system
3.8	3.7	I would imagine that most people would learn to use this system very quickly
2.7	2.7	I found the system very cumbersome to use
3.2	3.1	I felt very confident using the system
2.1	2.1	I needed to learn a lot of things before I could get going with this system

Table 1. SUS score question averages for improved (I) and baseline (B) system. Questions were answered in 5-point Likert scale from 1 (disagree) to 5 (agree). The better value on each row has been boldfaced; higher is better for odd numbered questions and lower is better for even numbered questions.

In the interviews 7 out of 10 users reported that they felt that the visualized prediction helped them in the task. The main stated reason for this was that it helped them predict the effects of their actions, but it was also felt useful for illustrating which keywords were related to each other, as dragging a keyword over the Intent Radar would cause related keywords to move in a similar fashion.

Based on the interviews, the majority of the users stated that they prefer the improved interface over the baseline. Five users preferred the improved system overall, 2 had mixed preferences depending on the situation, 1 preferred the baseline system overall and 2 indicated no explicit preference.

I	B	Question
3.1	3	The items recommended to me matched what I was searching for
3.7	3.4	The recommender system helped me discover new items
4.2	**4.3**	The items recommended to me are diverse
3.4	3.2	The layout of the recommender interface is adequate
2.7	2.3	The recommender explains why the items are recommended to me
3.4	2.6	The information provided for the recommended items is sufficient
3.1	2.8	I found it easy to tell the system what I want / don't want to find
4.1	4	I became familiar with the recommender system very quickly
3.4	3.1	I found it easy to modify my search query in the recommender
3.1	2.9	I understood why the items were recommended to me
3.3	3	Using the recommender to find what I like is easy
3.4	3.4	The recommender gave me good suggestions
3.1	2.9	Overall, I am satisfied with the recommender
3.3	3.3	The recommender can be trusted
3.7	3.5	I would use this recommender again, given the opportunity

Table 2. ResQue score question averages for improved (I) and baseline (B) system. Questions were answered in 5-point Likert scale from 1 (disagree) to 5 (agree). The better value on each row has been boldfaced; higher is better.

DISCUSSION AND FUTURE WORK

In this paper we identified two problems with the usability of interactive search engines where the user is assumed to be a passive provider of feedback, and has no practical means to anticipate the effects of her feedback actions on the system. We proposed a solution for improving the usability of these kinds of systems, demonstrated how it can be implemented in practice, and presented results indicating improvements in task performance, usability, perceived usefulness and user acceptance. We intend to carry out a larger user study with the next generation of the system to confirm these results.

ACKNOWLEDGEMENTS

This work has been partly supported by the Academy of Finland (Finnish Centre of Excellence in Computational Inference Research COIN, 251170) and TEKES (Re:Know). The research leading to this results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no 611570. Certain data included herein are derived from the Web of Science prepared by THOMSON REUTERS, Inc., Philadelphia, Pennsylvania, USA: Copyright THOMSON REUTERS, 2011. All rights reserved. Data is also included from the Digital Libraries of the ACM, IEEE, and Springer.

REFERENCES

1. Anick, P. Using terminological feedback for web search refinement: A log-based study. In *Proc. of SIGIR*, ACM (2003), 88–95.

2. Brooke, J. SUS–A quick and dirty usability scale. *Usability Evaluation in Industry 189* (1996), 194.
3. Chau, D. H., Kittur, A., Hong, J. I., and Faloutsos, C. Apolo: making sense of large network data by combining rich user interaction and machine learning. In *Proc. of CHI*, ACM (2011), 167–176.
4. Cutting, D. R., Karger, D. R., Pedersen, J. O., and Tukey, J. W. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proc. of SIGIR*, ACM (1992), 318–329.
5. Głowacka, D., Ruotsalo, T., Konyushkova, K., Athukorala, K., Kaski, S., and Jacucci, G. Directing exploratory search: Reinforcement learning from user interactions with keywords. In *Proc. of IUI*, ACM (2013), 117–128.
6. Karimzadehgan, M., and Zhai, C. Exploration exploitation tradeoff in interactive relevance feedback. In *Proc. of CIKM*, ACM (2010), 1397–1400.
7. Kelly, D., and Fu, X. Elicitation of term relevance feedback: An investigation of term source and context. In *Proc. of SIGIR*, ACM (2006), 453–460.
8. Kules, B., Wilson, M., Schraefel, M. C., and Shneiderman, B. From keyword search to exploration: How result visualization aids discovery on the web. Tech. rep., 2008.
9. Parra, D., Brusilovsky, P., and Trattner, C. See what you want to see: Visual user-driven approach for hybrid recommendation. In *Proc. of IUI*, ACM (2014), 235–240.
10. Pu, P., Chen, L., and Hu, R. A user-centric evaluation framework for recommender systems. In *Proc. of RecSys*, ACM (2011), 157–164.
11. Radlinski, F., Kleinberg, R., and Joachims, T. Learning diverse rankings with multi-armed bandits. In *Proc. of ICML*, ACM (2008), 784–791.
12. 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 CHI*, ACM (2004), 415–422.
13. Verbert, K., Parra, D., Brusilovsky, P., and Duval, E. Visualizing recommendations to support exploration, transparency and controllability. In *Proc. of IUI*, ACM (2013), 351–362.
14. Weber, R. *Basic content analysis*. Sage, 1990.
15. Yee, K.-P., Swearingen, K., Li, K., and Hearst, M. Faceted metadata for image search and browsing. In *Proc. of CHI*, ACM (2003), 401–408.