Finding *my* needle in the haystack: effective personalized re-ranking of search results in Prospector

Florian König[1], Lex van Velsen[2], Alexandros Paramythis[1]

[1]Johannes Kepler University
Institute for Information Processing and Microprocessor Technology (FIM)
Altenbergerstraße 69, A-4040 Linz, Austria
{alpar, koenig}@fim.uni-linz.ac.at

[2] University of Twente, Dpt. of Technical and Professional Communication
P.O. Box 217, 7500 AE Enschede, The Netherlands
l.s.vanvelsen@utwente.nl

Abstract. This paper provides an overview of Prospector, a personalized Internet meta-search engine, which utilizes a combination of ontological information, ratings-based models of user interests, and complementary theme-oriented group models to recommend (through re-ranking) search results obtained from an underlying search engine. Re-ranking brings "closer to the top" those items that are of particular interest to a user or have high relevance to a given theme. A user-based, real-world evaluation has shown that the system is effective in promoting results of interest, but lags behind Google in user acceptance, possibly due to the absence of features popularized by said search engine. Overall, users would consider employing a personalized search engine to perform searches with terms that require disambiguation and / or contextualization.

Key words: personalized web search, Open Directory Project (ODP), collaborative search, user evaluation, scrutability, adaptive search result re-ranking

1 Introduction

The continuously increasing rate at which information gets generated and accumulated on the Internet constitute a strong motivator for devising approaches that support the personalized retrieval and delivery of information items to users. Personalization in this context is intended to tailor the information- or result- spaces to individuals, in order to improve the relevance of retrieved items to their information needs. Prospector [1, 2] is a system that applies a range of techniques towards this end.

Specifically, Prospector is an adaptive meta-search engine / front-end that retrieves results from a user-selected, underlying search engine (e.g., Google, MS Live Search, Yahoo!) and personalizes their order (through re-ranking) to better match its users' interests. The later are modeled according to the users' ratings of results. The ontology of the Open Directory Project (ODP)[1] categorizes more than 4 million web sites

[1] For information on the Open Directory Project please refer to: http://www.dmoz.org

in over 700.000 topics, and provides the semantic meta-data for classifying results, recording ratings in the model(s) and calculating the relevance probabilities for re-ranking. This meta-data also allows the system to disambiguate homonymous interests and queries. Furthermore, Prospector makes use of group models (in the form of collective models for individuals with similar search interests) which are used for "bootstrapping" individual user models, and for predicting the user's search intentions using the preferences of like-minded users as a guide. Overall, Prospector implements a hybrid web search support layer, using concepts and techniques both from collaborative web searching, and from semantically enriched result filtering / re-ordering, to achieve high levels of search result personalization.

In this paper we first briefly relate the system to existing implementations of personalized search engines in section 2. In section 3 we present the main features of the third generation of the Prospector system, which has been preceded by two other versions, described in [1, 2]. We discuss several new features that are based on an evaluation of the second version, reported in [3]. The most important modifications implemented include better use of existing meta-data (see 3.1), usability enhancements (see 3.2), and a more stable re-ranking algorithm (see 3.3). In section 4 we report on a user-based, real-world evaluation of this version of the system, which has shown it to be effective in promoting results of interest to individual users. Section 5 discusses these and other results obtained, as well as their significance in the context of personalized search. Finally, directions for future work are also outlined.

2 Related work

Prospector can be broadly categorized as an adaptive search engine. The literature on this type of system is quite extensive (see [4] for an overview of work on personalized web search) and there are a number of personalized search systems that are related to Prospector in one or more characteristics. To start with, users in Prospector have a personal model of interests like in [5, 6, 7], the OBIWAN [8] and Persona [9] systems, and the first [10] and second [4] attempts at personalized search by Google. This sets it apart from purely collaborative engines like I-SPY [10] and makes it more similar to the Quickstep [11] system and its follow-up development Foxtrot [12]. In the current incarnation[2] of Google's personalized search, a similar approach is pursued, allowing users for example to move results up and down in the list and view other users' aggregated ratings and the corresponding ranking.

Whereas other systems rely on implicit feedback like search/click history [4, 5, 9, 10, 11, 12], general web surf history [8] or the current work context (documents, e-mails) [6], Prospector elicits explicit relevance feedback from its users. This approach has also been implemented in Persona [9], Quickstep [11], Foxtrot [12] and Google's current personalization features; this approach, although arguably more demanding for users, has the benefit of precision.

Similar to Chirita et al. [7] and Google's first search personalization engine, Prospector allows users to manually set high-level interests. However, in contrast to the

[2] A prototype is described at http://www.google.com/experimental/a840e102.html

aforementioned systems, these settings are not the only indications but merely act as a starting point for bootstrapping the user model and facilitating early personalization.

Quite a number of personalized search engines base the structure of their models on an ontology like the one from the Open Directory Project. Mostly they use just a few top levels and corresponding nodes of the mainly hierarchical topic categorizations [5, 6, 7, 10, 11, 12]. Prospector allows its models to be as fine grained as the full ODP ontology, similar to the OBIWAN [8] and the Persona [9] system. Specific measures to handle the resulting large size and possible sparsity of the model had to be employed as described in section 3.1.

In order to map query terms or search results to topics in the ontology, so that they can be related to the relevance information in the models, most of the systems mentioned up to now employ classification methods from machine learning. Prospector, like Persona [9] and the system by Chirita et al. [7] relies on the web resources already classified in the ontology. While this approach can only map a fraction of the total number of sites on the Internet, it avoids ambiguities and, as will be seen in the evaluation section (4.3), can still assign ontological meta-data to a high number of search results encountered in real-life searches.

Most of the systems mentioned so far don't allow users to view or edit their user model. Notable exceptions are Google's first and second personalized search engine as well as the Foxtrot system [12]. Allowing users to give direct profile feedback was found beneficial for the latter and has been implemented in Prospector as well.

3 The Prospector System

The operation of Prospector can be summarized as follows: first, the underlying search engine is used to retrieve (non-personalized) results for the user's query. These results are classified into thematic topics using the ODP meta-data. Classifications and corresponding interest information stored in the system-maintained user- and group- models are then used to determine an appropriate (personalized) ranking of the results. Users are presented with the re-ranked results, which they can rate on a per-result-item basis. The system uses these ratings and the classifications to update the individual and group models. The rest of this section elaborates on the operation of the system, explaining the modeling and re-ranking algorithms and focusing on new and modified features in the current third version of the system. A full account of the technical details including all the necessary internal calculations can be found in [2].

3.1 Ontology-based modeling of user interests

Prospector uses taxonomies as overlay [13] over the ODP ontology for modeling user and group interests. The structure of the models follows the mainly hierarchical layout of topics in the ODP ontology. Topics are identified by their path (e.g., "Arts > Music > DJs"), which provides the structural information for the taxonomy in the models. The overlay is formed by storing probabilities of interest in specific topics, for later use in personalizing result rankings (e.g., Arts*[0.66]* > Music*[0.79]* > DJs*[0.34]*).

In addition to each user's personal model there exist a fixed number of thematic group models. They are named after 13 top-level ODP topics (e.g., "Arts", "Sports") and represent high-level interests that users may share. Users have a certain degree of affinity from 0 ("no interest") to 5 ("high interest") to each of those groups. This determines the impact of a user's ratings on the group models, as well as the level of influence of group models on the personalization of search results. User and group models each have separate taxonomies, and for both types all of the ODP ontology can be used in the overlay (i.e., group models are not constrained to the sub-topics of the their corresponding top-level topic). To counteract the cold start problem [14], group models are initialized with the maximum probability of 1.0 for the top-level topic corresponding to the name of the group (i.e., in the group model "Arts" the probability of top-level topic "Arts" is set to 1.0). The following description of the modeling algorithm outlines additional aspects of the utilization of the ODP ontology.

To start with, each result item returned from the underlying search engine is classified in the ODP ontology. The system first looks for the whole URL in the ODP data and, if no match is found, searches for the host and domain part only. Classification returns zero or more topics (i.e., a web resource may be referenced in more than one place in the ODP ontology), all of which are taken into account in the new modeling and personalization algorithms. Although simplifying subsequent calculations, it proved problematic to limit the classification to a single, "best matching" topic [3].

Successfully classified result items can then be rated by the user, thus recording the interest feedback for the topic(s) in which they were classified. This operation involves several steps for each of the topics: First, the path of topics and sub-topics (in whole or parts) is created in the model without interest probabilities if it does not exist yet (e.g., Sports[0.72] > *Soccer[]* > *Coaching[]*). Second, probabilities are derived for newly created topics either from topics higher-up in the hierarchy, group models or (as a weighted sum) from both if possible. If no probability can be derived, the neutral default probability of 0.5 (i.e., neither interested nor not interested) is used.

The first method (deriving from parent topics) implements inter-topic interest propagation [13] and addresses the problem of sparsity [14]. It ensures that probabilities are available for a large number of topics, even if only few receive ratings. To model the increasing uncertainty as the distance to the "source" topic grows, derived values progressively tends towards the neutral probability (e.g., Sports[0.72] > Soccer*[0.61]* > Coaching*[0.57]*). Probabilities of corresponding topics in group models are derived by weighting them with the user's affinity to the respective group and taking the average of these probabilities. The values are again calculated to tend towards 0.5, only this time with decreasing affinity. This second derivation method addresses the latency problem [14] and bootstraps model parts that have not yet received a rating.

The final step after deriving probabilities is recording the actual rating by increasing or decreasing the values in the user model and (weighted by the affinity) the group models. This affects the entire path, scaled to the distance from the target topic, which receives the largest change (e.g., a negative rating of a result in 'Coaching' causes Sports[0.72] > Soccer[0.61] > Coaching[0.57] to become Sports[0.67] > Soccer[0.50] > Coaching[0.41]). The amount of change is non-linearly dependent on the original probability: ratings around the neutral probability 0.5 show more effect than those near the bounds of 0 and 1. This allows topics with low interest specificity to quickly converge to a certain bias and at the same time keeps unambiguous interests stable.

3.2 Interactive features

In order to get personalized search results users first have to register. At the first login they are asked to specify their interest in the 13 top-level ODP topics. It is explained to the users that this way they will benefit from result ratings by users with similar interests. In the evaluation of the preceding version of the system [3], people had difficulty judging the meaning and scope of the topics by just their name. Therefore, the topics are now described in more detail by also listing representative sub-topics.

For each search, users may choose the underlying search engine to be used by selecting the corresponding tab above the query text field (see Fig. 1). When issuing a query, this engine is accessed, and its results are retrieved and classified as described above. The classification paths are displayed for each result, and the tree control on the left side of the results page lets users filter results by these topical categories.

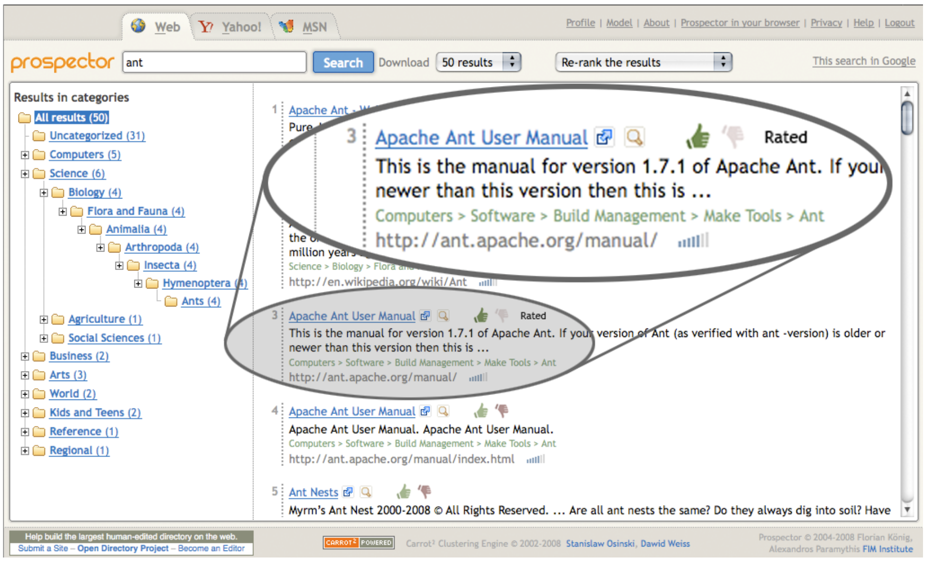

Fig 1. Prospector's main interface.

By rating individual results positively (thumbs up) or negatively (thumbs down) users implicitly express their preference for certain topics. In its previous version, Prospector displayed a rating frame at the top of the page of a clicked result. This approach was abandoned as it created both technical and usability problems [3]: some sites employ "frame breakout" mechanisms that cause the frame to disappear, and users were observed overlooking or disregarding the possibility to rate. Quickly evaluating a result is now possible by toggling an embedded preview below the result with the magnification glass icon; opening a result in a new window is facilitated by the arrow icon. When previewing or returning from an opened result, the user is notified / reminded of the possibility to rate that result, by pulsating its thumbs a few times.

Each rating modifies the appropriate user- and group- models, thus affecting the calculation of relevance probabilities of classified results in subsequent searches. A combination of these probabilities with each result's original rank is used for re-ranking (explained further in the next section). To give users a way to quickly assess the system-calculated relevance of a result, its probability is visualized next to the URL by means of a "connection quality" indicator, as used in mobile phones. Hovering over the bars with the mouse shows the exact relevance percentage in a tool-tip. This graphical representation was chosen because it better caters to users' preferences over (long) textual relevance information, as suggested by Coyle and Smyth [15].

For logged in users the ranking is by default personalized with respect to their user model and the models of associated groups. In addition, users can request that results be re-ranked using a particular group model (e.g., "re-rank for people interested in arts"). This feature is intended to let users focus on the specific high-level topic represented by the group, and is also available for anonymous, not logged in users.

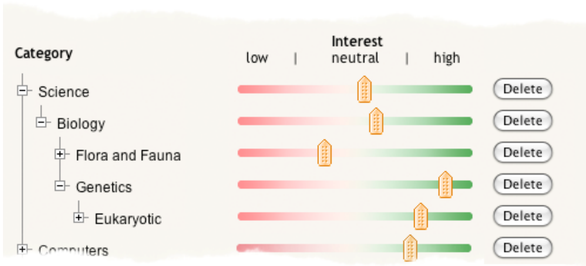

Fig 2. Prospector's user model view.

The user models in Prospector are scrutable [16], allowing users to inspect, correct and fine-tune them, while at the same time strengthening their confidence in the system. Affinities to groups, as set when registering, can be changed at any time. The interests inferred from the ratings can be viewed and changed as well (see Fig. 2): They are presented by means of a tree summarization of the user model taxonomy and show the paths to the topics for which a result has been rated. The interest probability of each topic is visualized by a stepless slider, ranging from low to high. Users can change the probabilities via these sliders. Topics can also be removed from the model, which gives users the means to purge old or invalid interest information.

3.3 Personalized re-ranking of search results

For re-ranking the relevance probability of each result item needs to be calculated. This value is composed from the interest probability of each topic in which the result has been classified and its original rank. If the user model does not contain an interest probability for a topic, the value is derived as described in section 3.1. If a value cannot be established by any of these methods, the neutral probability 0.5 is used. The calculated relevance probabilities of each topic are combined to a weighted average. The affinity of the user to the group with the same name as the top-level topic in the full classification path of the topic at hand is used as the respective weight.

Finally, each result's relevance probability is combined with its rank as returned by the underlying search engine (as suggested also by Teevan et al. [8]). The rank is normalized into the interval [0..1] (e.g., with 50 results: first result's rank becomes 1, the 25th one 0.5 and the last one 0). The relevance probabilities are normalized as well, in the same value space. These methods for rank and score normalization are described by [17] and provide the basis for rank fusing: the combination of individual rankings to improve the resulting order of items.

The normalized rank and score values are then combined with a weighted extension [17] of the *CombSUM* method [18]. Prospector uses this final value for re-ranking the result list accordingly. Compared to previous versions, taking the original rank into account has two essential benefits: (a) the ranking is more stable and results are only gently nudged up or down in the list instead of "jumping around" after rating and searching again; (b) the information contained in the original ranking, which is based on the query itself rather than on the interests of the user, is not lost and helps better represent the search needs of the user.

4 Evaluating Prospector

The current (third) version of Prospector underwent a longitudinal evaluation in which the participants were free to use the system as they liked for a longer period of time. This evaluation followed a task-based experimental evaluation of the system, reported in [3]. Where the first evaluation provided us with information on how Prospector functions in a controlled setting and how this can be improved upon, the evaluation reported here assesses the usefulness and usability of Prospector in a real-life setting, comprising a great diversity in search tasks per person, and changing interests. In other words, this approach ensures high ecological validity. The rest of this section describes the evaluation setup and our findings. A more detailed account of the evaluation with a stronger focus on the evaluation design, user-centered qualitative findings elicited by questionnaires and system-centered quantitative results from the data logs can be found in [19].

4.1 Evaluation setup

The evaluation involved 21 volunteers (out of 130 contacted) who agreed to use Prospector as their primary search engine for 12 days, to have their use with the system logged and, finally, to complete questionnaires. Specifically, users responded to three questionnaires, one before starting to use the system, one after five days of use, and one after the total 12 days of use. These questionnaires covered the following issues, primarily by means of open-ended questions: (a) demographics, internet use, experience with personalized systems and use of search engines; (b) expectations of using Prospector and the usefulness of a scrutable user model; (c) perceived usefulness of Google and of Prospector; (d) comparisons between Prospector and Google; (e) important (satisfactory or dissatisfactory) incidents the users experienced; (f) the clarity and correctness of the system's scrutable user model; and, finally, (g) different dimensions of usability (such as predictability, controllability and privacy) which have been shown to be of particular relevance to adaptive systems (see [20]).

4.2 Findings from Questionnaire Responses

Nineteen men and two women participated in the study, with an average age of 25.8 years (SD = 2.8). Most of them were students in the Johannes Kepler University in Linz, Austria. They rated their computer skills as high and used the Internet on a daily basis. All participants used Google as their primary search engine, and approximately two thirds performed at least five searches a day (up to more than 15).

Most participants expected Prospector to outperform Google, by reducing search times (six participants) and giving better results (six participants). However, end results showed that the users perceived Google as more useful than Prospector. They also indicated that they preferred Google for searching; interestingly, some of the participants stated that their preference depended on the nature of the search task. They liked Google better for searching for simple facts, but thought Prospector had an added value when conducting searches related to their personal interests or study. Furthermore, several participants appeared to prefer Google because Prospector did not offer results in German (the mother tongue of all participants). As one person stated: “I prefer Google, because it provides the possibility to search nationally. With Prospector one doesn't find regional results. The program doesn't like German words.”

The analysis of noteworthy incidents reported by the users revealed two causes that led to dissatisfaction with Prospector: irrelevant search results (mentioned 9 times) and illogical re-ranking of search results (mentioned 6 times). On the positive side, participants mentioned more than once Prospector’s particular helpfulness when submitting a query containing words with ambiguous meanings, and specific searches for which Prospector was useful, like product reviews or scientific articles.

As far as Prospector’s scrutable user models are concerned, participants almost uniformly stated that they understood the user model visualization. The majority of the participants also found their user model to be a mostly accurate reflection of their search interests, with several of the remaining participants stating that they had not provided the system with enough feedback to generate a correct user model. However, the data logs cast doubts over the participants’ answers. Even though all participants gave their opinion about the user model, the data logs show that only 16 of them inspected their user model during their use of the system, and only 11 had done so before answering about the understandability of the user model’s visualization. Therefore, the results regarding user modeling remain inconclusive.

Regarding the usability issues predictability, comprehensibility, unobtrusiveness and breadth of experience: most participants stated they thought they were fully or for a larger part in control over the system; privacy was not thought of as a barrier to use Prospector; the majority of the participants believed that Prospector could deliver the results they desired (interestingly, six participants commented that the system had the potential to deliver relevant search results, but conditionally – e.g., if results in the German language were to be taken into account).

4.3 Data Log Analyses

Recording of ratings and the calculation of a personalized score require a search result item to be classified in the ODP. Even though the size of the ontology is small compared to the total number of pages on the Internet, approximately 35% of the results returned from the underlying search engine could be classified by Prospector.

The average number of searches performed daily by all users over the 12 days of the study was 54.33 (*SD* = 27.97). The average total number of positive and negative ratings was 19.75 (*SD* = 18.20). The trend was declining in all cases. Of note is the fact that there is no significant difference between the number of positive and negative ratings, also over time [19].

To determine whether personalization has positively affected the ranking of search results, we examined whether the results participants clicked on were ranked higher than in the original set. Specifically, for all days, we calculated the distance between the personalized and original ranks of viewed results. This distance was positive if the result had been promoted, negative if it had been demoted, and 0 if it had retained its rank. Fig. 3 displays the average distance between these ranks for each day. For most days, the difference was positive for the personalized rank. For all 12 days, the viewed pages had been promoted by, on average, 4.75 ranks (*SD* = 11.52). To test the re-ranking effect, we compared the average rank distance for each viewed result to 0. This difference is significant (t = 9.14; df = 490; p<.01): Search results that participants viewed were, on average, placed higher up in the list, due to personalization.

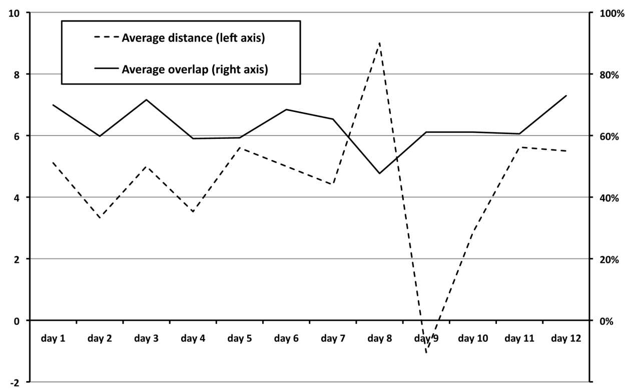

Fig 3. Average distance between original and re-ranked results, and average percentage of original results still ranked between 1 and 12 after re-ranking ("overlap") per day

To ascertain that participants did not consult search results simply because they were ranked highly, regardless of their relevance, we examined whether the first 12 results contained a disproportionately high percentage of items brought there by Prospector. We chose 12, as on an average-sized screen a user would see 6 results in one screen-full and most people do not look beyond the first 10 [21] – we rounded that number up to two screen-fulls. Fig. 3 displays the daily average percentage of results among the first 12 that were originally there. Over 12 days, the mean percentage is 65.10%. This implies that although the majority of "visible" results would have been there anyway, users chose the re-ranked results on purpose and not because they were conveniently placed at the top of the list. In addition, the improved ranking algorithm (see 3.3) allows re-ranked results to stay interspersed with the not re-ranked ones.

Furthermore, the two metrics "Rank scoring" [22] and "Average Rank" [23] were employed. Rank scoring shows how close to the optimal ranking a set of search results is, whereby 'optimal' denotes a situation in which all the consulted results appear at the top of the list. The importance of the ranks decreases exponentially (e.g., a correct rank 1 creates a better score than a correct rank 10). We performed a paired samples t-test between the original rank score average (M = 5.05, SD = .59) and the personalized rank score average (M = 6.75, SD = 1.19). The averages were calculated from the rank score values of the 12 days. This difference is significant (t = -6.92; df = 11; p<.01): Personalized rank scores were higher than the original ones, which means that the personalized rankings were closer to the optimal ranking (see Fig. 4).

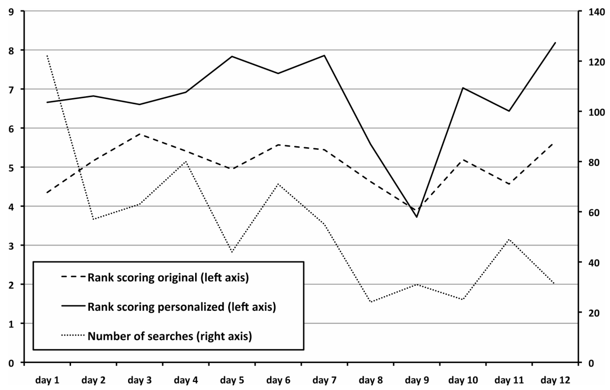

Fig 4. Rank scoring of the personalized and original ranks of viewed results. Higher scores mean, that the ranking is closer to the optimal ranking (i.e., all user-clicked results at the top).

The average rank measure was calculated for the original and the personalized ranking of consulted results on a per day basis. The personalized results had a lower average rank in all cases, except on day 9. A lower average rank means that the consulted results appeared higher up in the result list. These results reinforce the findings derived through the average distance measure (see also [19]).

5 Conclusions and Future Work

In this paper we have discussed the Prospector system and a real-life, user based evaluation. This study has shown that Prospector effectively brings up search results with relevant informational value in a list of search results. However, user perceptions on the usefulness of the system were not in favor of Prospector: the participants thought their primary search engine (Google) was more useful. The comments they provided throughout the evaluation led us to think that this opinion was partly due to missing features popularized by the Google search engine (specifically, localized search, spelling suggestion). As has been shown, people attribute high value to the appearance and features their primary search engine offers [21]. Prospector offered a different interface with different features and this may have biased the participants' perception of its usefulness, regardless of the system's actual value for searching. One way to design for this implication is to replicate the features and 'look and feel' people expect of a search engine – the 'look and feel' has been shown to have a major impact on users' perception of performance, independently of the search results themselves [24]. A different solution would be to offer Prospector as a browser plug-in that personalizes the results page of an existing search engine "in place".

The Prospector algorithm can currently be adjusted via two weight values: one gearing user models and group models unto each other, the other concerned with balancing personalized ranking against original ranking. It will be crucial to acquire optimal ranges for these values by means of an experiment with several versions of Prospector, each set to a different configuration of weight values.

The participants had very high expectations of Prospector, and expected it to outperform Google. This meant that the search result they needed should be listed first or second, but not lower. These expectations are apparently hard to meet, especially as users will want to see an added value fast and it may take some time for a personalized search engine to deliver top-quality results. This emphasizes the fact that introducing a new personalized search engine in a market dominated by Google is a very hard challenge. In addition, the slight decrease in perceived usefulness over time suggests that the system's measures against the cold start problem can still be improved. A more controlled study could help in keeping the usage (searching, rating) at an equal level throughout and may therefore indicate whether decreasing usage was the reason for bad performance and perception.

In closing, the evaluation has suggested some circumstances in which personalized search might be more rewarding for users. These are the searches which our participants described as 'personal' or searches without a clear-cut answer. Typical for these searches are, as Marchionini terms it, relatively low answer specificity, high volume and high timeliness [25]: the answer to the search is not easily recognized as being correct (e.g., a suitable hotel in Amsterdam), it has to be found in a large body of information and, finally, the user has to invest some time in finding the right answer. This implies that personalized search may have more success in specialized areas of search (e.g., digital libraries) than it has for finding general information.

Acknowledgements. The work reported in this paper has been supported in part by: (a) the EU-funded "Adaptive Learning Spaces" (ALS) project (229714-CP-1-2006-1-NL-MPP); and (b) the "Adaptive Support for Collaborative E-Learning" (ASCOLLA) project, supported by the Austrian Science Fund (FWF; project number P20260-N15).

References

1. Schwendtner, C., König, F., Paramythis, A.: Prospector: an adaptive front-end to the Google search engine. In: LWA 2006: 14th Workshop on Adaptivity and User Modeling in Interactive System, pp. 56–61. University of Hildesheim, Institute of Computer Science, Hildesheim (2006)
2. Paramythis, A., König, F., Schwendtner, C., Van Velsen, L.: Using thematic ontologies for user- and group-based adaptive personalization in web searching. Paper presented at the 6th International Workshop on Adaptive Multimedia Retrieval, Berlin (2008)
3. Van Velsen, L., Paramythis, A., Van der Geest, T.: User-centered formative evaluation of a personalized internet meta-search engine. (In review)
4. Micarelli, A., Gasparetti, F., Sciarrone, F., Gauch, S.: Personalized Search on the World Wide Web. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web, LNCS, vol. 4321, pp. 195–230. Springer, Heidelberg (2006)

5. Pretschner, A., Gauch, S.: Ontology based personalized search. In: 11th IEEE International Conference on Tools with Artificial Intelligence, pp. 391–398. IEEE Computer Society (1999)
6. Liu, F., Yu, C., Meng, W.: Personalized web search by mapping user queries to categories. In: 11th international conference on Information and knowledge management, pp. 558–565. ACM, New York (2002)
7. Chirita, P.A., Nejdl, W., Paiu, R., Kohlschütter, C.: Using ODP Metadata to Personalize Search. In: 28th ACM International SIGIR Conference on Research and Development in Information Retrieval, pp. 178–185. ACM, New York (2005)
8. Teevan, J., Dumais, S.T., Horvitz, E.: Personalizing search via automated analysis of interests and activities. In: 28th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 449–456. ACM, New York (2005)
9. Tanudjaja, F., Mui, L.: Persona: A contextualized and personalized web search. In: 35th Annual Hawaii International Conference on System Sciences, pp. 67–75. IEEE Computer Society Press, Los Alamitos (2002)
10. Smyth, B., Balfe, E.: Anonymous personalization in collaborative web search. Information Retrieval 9, 165–190 (2006)
11. Middleton, S.E., De Roure, D.C., Shadbolt, N.R.: Capturing Knowledge of User Preferences: ontologies on recommender systems. In: 1st International Conference on Knowledge Capture, pp. 100–107. ACM, New York (2001)
12. Middleton, S.E., Shadbolt, N.R., De Roure, D.C.: Ontological User Profiling in Recommender Systems. ACM Transactions on Information Systems 22, 54–88 (2004)
13. Brusilovsky, P., Millán, E.: User Models for Adaptive Hypermedia and Adaptive Educational Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web. LNCS, vol. 4321, pp. 3–53. Springer, Heidelberg (2007)
14. Anand, S., Mobasher, B.: Intelligent Techniques for Web Personalization. In: Mobasher, B., Anand, S. (eds.) Intelligent Techniques for Web Personalization. LNCS, vol. 3169, pp. 1–36. Springer, Heidelberg (2005)
15. Coyle, M., Smyth, B.: Supporting intelligent web search. ACM Transactions on Internet Technology 7, 20 (2007)
16. Kay, J.: Stereotypes, student models and scrutability. In: Gauthier, G., Frasson, C., VanLehn, K. (eds.) Intelligent Tutoring Systems. LNCS, vol. 1839, pp. 19–30. Springer, Heidelberg (2000)
17. Renda, M.E., Umberto, S.: Web metasearch: rank vs. score based rank aggregation methods. In: 2003 ACM symposium on Applied computing, pp. 841–846. ACM, New York (2003)
18. Shaw, J., Fox, E.: Combination of Multiple Searches. Paper presented at the Text REtrieval Conference, Gaithersburg, USA (1993)
19. Van Velsen, L., König, F., Paramythis, A.: Assessing the Effectiveness and Usability of Personalized Internet Search through a Longitudinal Evaluation. In: 6th Workshop on User-Centred Design and Evaluation of Adaptive Systems, held in conjunction with the International Conference on User Modeling, Adaptation, and Personalization, pp. 44–53. CEUR-WS.org (2009)
20. Paramythis, A., Weibelzahl, S.: A decomposition model for the layered evaluation of interactive adaptive systems. In: Ardissono, L., Brna, P., Mitrovic, A. (eds.) User Modeling 2005. LNCS, vol. 3538, pp. 438–442. Springer, Heidelberg (2005)
21. Keane, M.T., O'Brien, M., Smyth, B.: Are people biased in their use of search engines? Communications of the ACM 51, 49–52 (2008)
22. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: 14th Annual Conference on Uncertainty in Artificial Intelligence, pp. 43–52. Morgan Kaufman, San Francisco (1998)
23. Dou, Z., Song, R., Wen, J.: A large-scale evaluation and analysis of personalized search strategies. In: 16th international conference on WWW, pp. 581–590. ACM, New York (2007)
24. Jansen, B.J., Zhang, M., Zhang, Y.: The effect of brand awareness on the evaluation of search engine results. In: CHI '07 Extended Abstracts on Human Factors in Computing Systems, pp. 2471–2476. ACM, New York (2007)
25. Marchionini, G.: Information seeking in electronic environments. Cambridge University Press, New York (1995)