Using thematic ontologies for user- and group-based adaptive personalization in web searching

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

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

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

Abstract. This paper presents Prospector, an adaptive meta-search layer, which performs personalized re-ordering of search results. Prospector combines elements from two approaches to adaptive search support: (a) collaborative web searching; and, (b) personalized searching using semantic metadata. The paper focuses on the way semantic metadata and the users' search behavior are utilized for user- and group- modeling, as well as on how these models are used to re-rank results returned for individual queries. The paper also outlines past evaluation activities related to Prospector, and discusses potential applications of the approach for the adaptive retrieval of multimedia documents.

1 Introduction

The phenomenal growth of the web in the past decade has resulted in an unprecedented amount of information being available in accessible electronic form, and this trend can only be expected to strengthen in coming years. This proliferation of information, though, has rendered locating the items of information that are indeed interesting to a user an increasingly difficult task. To address the needs of modern web searching, several approaches have been proposed and practically applied that improve upon traditional term-matching information retrieval techniques.

Significant innovations include the works reported in [1] and [5] that have exploited document connectivity information to significantly improve retrieval quality. More recently, other researchers have sought to exploit context as a means of supplementing vague queries and so "guiding" search [6]. A different line of work has looked at clustering techniques as a way to impose order on a collection of search results, with a view to identifying different conceptual groupings of results [3][4].

In the realm of collaborative search systems, which utilize the collective experiences of like-minded groups of users to improve upon search results, a representative and widely acclaimed system is I-SPY [8][9]. I-SPY implements an adaptive collaborative search technique that enables it to selectively re-rank search results according to the learned preferences of a community of users. Effectively I-SPY actively promotes results that have been previously favored by community members during related searches so that the most relevant results are at the top of the result list [8]. I-SPY monitors user selections or "hits" for a query and builds a model of query-page relevance based on the probability that a given page will be selected by the user when returned as a result to a specific query.

Along a different line of work, researchers have addressed the use of semantic metadata to represent user interest profiles and adapt searches so that results better fit those profiles. Two such systems based on metadata provided by the Open Directory Project[1] are the ones described in [10], and [2]. The first paper describes Persona, a system which utilizes ODP metadata for creating taxonomies of user interests and disinterests and tree coloring to represent user profiles. Taxonomy nodes visited are 'colored' by the number of times they have been visited, by user ratings if available, and by the URLs associated with the node [10]. In the second paper, Chirita et al. [2] have used ODP metadata to create user profiles, and then used various approaches to calculating the distance between a given search result item and the user's profile, to decide that item's rank. Users pre-select ODP categories that they are interested in for the creation of their profiles; the system does not have an adaptive component, so these profiles do not evolve over time. The distance calculation approaches range from ones based primarily on graph node distances, to a version of the PageRank [1] algorithm modified to include a measure of the semantic similarity between nodes in a taxonomy. User-based experiments have shown that these approaches to search personalization deliver superior results to their non-personalized counterparts [2].

This paper presents the Prospector system [7], an adaptive meta-search layer on top of mainstream search engines. Prospector implements a hybrid approach, whereby: (a) ODP ontological metadata is used as the basis for dynamically maintained models that capture the search preferences of individuals; and, (b) group models are created and maintained alongside the individual user models, and subsequently used to improve upon search results of individuals belonging to a group.

2 The Prospector system

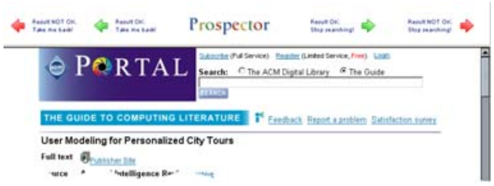
Prospector is a generic front-end to mainstream search engines. There have been two versions of the system so far, the first described in [7] and the second in this paper. The most important difference between the two versions is the employment of custom probabilistic algorithms for the modeling and adaptation processes, which are described herein (the previous version used item weighting algorithms). This change has brought about: (a) improved system effectiveness in adaptively reordering search results for individuals and thematic groups; and (b) increased comprehensibility by end users of result item relevance ratings, and of their own models / profiles.

[1] Open Directory Project: http://dmoz.org/

(b) Searching

(a) Setting up a user profile

(d) Inspecting and manipulating the user model



(c) Rating search result items

Fig. 1. A quick visual tour of Prospector.

Here, as well as in [7], we present the system running on top of the Google search engine. Due to space restrictions, we only provide a brief overview of system interactivity here, focusing on features that have changed since the first version. A more complete account of the interactive aspects of the system can be found in [7].

In Prospector, a basic anonymous search returns exactly the same results as a search made directly on the Google site. Adaptivity comes into play in two guises: firstly, while still remaining anonymous, the user can have the results re-ranked by the system, according to thematically-based group models; and, secondly, users can register (and log in), progressively building up their personal interest profile, which is then used to automatically re-rank search results.

Modeling of search behavior in Prospector takes place on the basis of ODP metadata. Group modeling, in the first two versions of the system, has used a fixed set of groups, one for each of the top-level categories in ODP. In compensation to the absence of dynamically determined groups, the system allows users to define the "degree of affiliation" to these groups, through the definition of their level of interest in the respective thematic categories (see Fig.1-a). Immediately after the creation of the user profile, which in turn results in the creation of a corresponding user model, the user can perform queries and have the results re-ranked according to their individual user model, as well as according to the models of groups that the user is interested in (see Fig.1-b). Users are able to rate individual search result items, either from within the results' page (Fig.1-b), or using the controls in the "rating frame" which appears when they follow the link of a search result item (Fig.1-c). Finally, Prospector allows users to both inspect and manipulate their personal model, to better and faster fine-tune it to their search preferences (Fig.1-d).

As already mentioned, Prospector is an effort to create a hybrid web search support layer, which uses concepts and techniques both from collaborative web searching, and from semantically enriched result filtering / re-ordering, to achieve high levels of personalization of search results. The rest of the paper will provide a detailed view of the modeling and adaptation algorithms used in the system, as well as on the preliminary conclusions we have drawn from our work on the system thus far.

3 Ontology-based modeling of search behavior

3.1 Basic concepts

Before we proceed to presenting the modeling and re-ranking algorithms used in Prospector, we need to establish the basic concepts used.

To start with, both user and group models in Prospector are, in effect, overlay models over the ODP category ontology[2]. Specifically, the models are structured hierarchically following the topic relations in the ODP ontology, and each node in a user- or group- model contains the probability that the user or group, respectively, are interested in items (web sites) that are associated with the node's category. A category "path" is the branch in the hierarchy that has a specific category as its end node.

Based on the above, the following conventions are used in the rest of this paper: *U* denotes the set of all system users. *G* denotes the set of top-level ODP categories used by Prospector as groups. When a user first registers in the system, they are presented with a form in which they specify their level of interest in each of these categories / groups using a 5-point scale. This value is denoted as *Interest*(*u*, *g*), $u \in U$ (the current user), and $g \in G$; the value space for *Interest*(*u*, *g*) is therefore {*0, 1, 2, 3, 4, 5*}, with a value of zero signifying no interest in the respective category / group.

Using the above, we define *AverageInterest*(*g*) to be:

$$AverageInterest(g) = \frac{\sum_{k=1}^{N} Interest(u_k, g)}{N} \quad (1)$$

[2] In the ODP dataset, categories are termed "topics".

where N is the number of users for whom *Interest*(u, g) > 0. In other words, *AverageInterest*(g), is the average interest value for g, over all users that have any interest in g.

This, in turn, is used to determine the influence of group models on the individual user model, and vice versa. This value is encapsulated in an influence coefficient:

$$Influence(u,g) = \frac{Interest(u,g)}{AverageInterest(g)} \frac{1}{N} \quad (2)$$

where, again, N is the number of users for whom *Interest*(u, g) > 0. Note that for a given user, this coefficient is defined for all groups in which the user has expressed interest in, and is quite likely to differ for each such group.

We use R to denote the set of search result items returned by a query q, and $c(r)$ (or, simply, c) to denote the primary ODP category associated with a search result item r, with $r \in R$. This category is derived as follows. Let $C(r)$ be the set of all ODP categories associated with result item r. For each category $c \in C(r)$, *Items*(c) denotes the total numbers of items (links, PDF documents, RSS feeds, etc.) associated with that category. The strategy used in the current version of Prospector is to use the category with the highest number of items associated with it, i.e.,

$$c(r) = c \in C(r) \quad and \quad Items(c) \geq Items(c'), \forall c' \in C(r) \quad (3)$$

Finally, for an ODP category c, we define *Depth*(c) to be the number of path elements that the category comprises (e.g., *Depth*("Top/Computers/Software") = 3).

3.2 User modeling using search result item ratings

As already seen, users have the possibility to rate a specific search result r. Such ratings modify the probability that the user is interested in the respective category $c(r)$ – denoted as c for brevity. The desiderata for probability modifications in this context are: the first few ratings for a category should have "immediate" effects, making the system quickly converge to the user's preferences with respect to the category at hand; for categories with well-established user preferences (e.g., categories with many negative ratings), an "opposite" rating, i.e., a rating that goes against the usual user ratings for the category, should not have major effects on the probability.

To attain the desired behavior, the probability $P(u, c)$ that user u is interested in items associated with the ODP category c is defined as follows:

$$P(u,c) = \frac{\cos\left(\left(1 - \frac{positive(u,c) + 5}{10}\right) \cdot \pi\right)}{2} + 0.5 \quad (4)$$

Where, *positive*(u, c) is defined as the sum of the number of positive ratings made by user u for items in category c, minus the sum of the number of negative ratings made by user u for items in category c. The value of *positive*(u, c) is constrained to the space [-5 .. 5]. The actual amount by which the probability in the user model changes is then: $\Delta P(u,c,a) = P_{after}(u,c) - P_{before}(u,c)$, where P_{before} and P_{after} refer to $P(u, c)$, after and before the user rating *a* respectively.

When a rating is made, not only the probability for the specific search result item's category is modified, but also the probabilities of all categories that are its ancestors. In other words, ratings for items in a category affect the entire category "path". The effects of a rating are scaled to reflect the distance of path nodes from the actual category for which the rating was made – closer nodes are affected more than distant ones. Thus, for all categories *c′* that are ancestors of *c*, and for a rating *a* we have:

$$\Delta P(u,c',a) = \left(P_{after}(u,c') - P_{before}(u,c')\right) \cdot \frac{Depth(c')}{Depth(c)} \qquad (5)$$

To apply this propagating modification approach, it is of course necessary that the user model contains the entire path that corresponds to a category. If the path does not exist prior to a rating, it is created and populated on the fly using "bootstrap" probabilities derived from the models of groups that the user is interested in, and from the user's own model, if the later contains any segment of the path. The approach to deriving probability values in this context is the same as the one used for predicting values for unrated categories, and is discussed in section 4 below.

3.3 Group modeling using search result item ratings

When a rating is made, the group models for all groups / categories that the user is interested in are also modified. In essence, user ratings have similar effects on group models as they do for user models, but these effects are scaled to reflect both the user's own interest in a group, and the number of users that are interested in that group. The rest of this section will provide formal definitions for the above.

To start with, we define $P(g, c)$ to be the probability that a user that is highly interested in group *g* (i.e., has an interest value of 5 for *g*) will be interested in items associated with the ODP category *c*. When the system starts for the first time, all group models are initialized to reflect a high interest in all categories that are subcategories of that group (i.e., $P(g, c) = 1.0$, when $c=g$).

When a rating is made for a category *c*, by user *u*, then for all groups *g* that *u* is interested in, $P(g, c')$ (where c' is an ancestor of *c*, including *c* itself) is modified as follows:

$$\Delta P(g,c',a) = \Delta P(u_g,c') \cdot \frac{Depth(c')}{Depth(c)} \cdot Influence(u,g) \qquad (6)$$

where, $\Delta P(u_g,c',a)$ denotes the modification in $P(g, c')$ that would have resulted if an imaginary user u_g representing the entire group had made the rating *a*. $\Delta P(u_g,c',a)$ is calculated using a formula similar to the one in equation (4), with *positive*(u, c) replaced with *rating*(u_g, c, a), defined as follows:

$$rating(u_g,c,a) = 1 - \frac{\arccos\left(\left(P(g,c) - 0.5\right) \cdot 2\right)}{\pi} + increment(a) \qquad (7)$$

with *increment*(a) given the value of 0.1 for positive ratings and -0.1 for negative ratings. Note that as equation (6) suggest, the propagating modification approach is used for group models in the same way as for individual user models.

4 Ontology-based reordering of search results

The re-ranking of results is based on the probability that a given result will be of interest to a given user. The process of determining this is as follows: When a user performs a search, result items r that correspond to sites that exist in the dataset are associated with a category $c(r)$, as described in section 3.1 above. $P(u, r)$ denotes the probability that user u is interested in item r. This probability is defined as follows:

- If the item r cannot be associated with an ODP category, then $P(u, r) = 0.5$
- If the user's individual model already includes a value for $c(r)$ (on the basis of previous user ratings), that value is used verbatim
- If the user's individual model does not include a value for $c(r)$, then derive a value from the models of groups in which the user is interested, as well as from the user's own model, if the later contains values for ancestor categories of c

In the context of the third of the above cases, the following definitions are made: $P_{group}(u, c)$ denotes the probability that user u is interested in category c, as derived from the models of groups in which the user is interested; $P_{inherited}(u, c)$ denotes the probability that user u is interested in category c, as calculated from the user's own model, using ancestor categories of c. $P_{predicted}(u, c)$ which denotes the overall predicted probability that user u is interested in category c, is then defined as:

$$P_{predicted}(u,c) = \lambda \cdot P_{groups}(u,c) + (1-\lambda) \cdot P_{inherited}(u,c) \quad \textbf{(8)}$$

The factor $\lambda \in [0\ldots1]$ can be modified to favor predictions based on group models or predictions based on the user's own model. In the current version of Prospector, its value has been set to 0.75 (thus favoring groups), but its effect has not been experimentally validated at the time of writing. If either of $P_{group}(u, c)$ or $P_{inherited}(u, c)$ are not available, then the other one is used exclusively. If neither of them are available, a "default" probability of 0.5 is used instead. In the above equation, $P_{group}(u, c)$ is defined as:

$$P_{groups}(u,c) = \frac{\sum_{i=1}^{N}\left(\left(P(g_i,c)-0{,}5\right)\cdot\frac{Interest(u,g)}{5}+0.5\right)}{N} \quad \textbf{(9)}$$

where g_i are all the groups in which user u is interested. $P_{inherited}(u, c)$ is defined as:

$$P_{inherited}(u,c) = \frac{\sum_{i=1}^{N}\left(\left(P(u,c_i)-0.5\right)\cdot\frac{Depth(c_i)}{Depth(c)}+0.5\right)}{N} \quad \textbf{(10)}$$

where c_i are all the ancestor categories of c for which there already exists a node in the model.

Note that $P_{predicted}(u, c)$ is also used when “bootstrapping” a user model node, as described in section 3.2 above. When bootstrapping occurs within group models, an appropriately modified version of $P_{inherited}(u, c)$ is used instead.

5 Evaluation activities

The version of the Prospector system described in this paper underwent formative user-centered evaluation at the University of Twente. Three different evaluation methods were applied in this study: thinking aloud, interviews and questionnaires. Data logs of the participants’ activity in the system (including searching, viewing and rating results, inspecting and modifying their model, etc.) were also collected. A full report of evaluation activities and obtained results is being prepared for separate publication. Below we summarize the most important, preliminary, findings.

During the evaluation, 32 participants were given the task of searching for youth hostels and museums of modern art in large European cities. These tasks have been chosen because the related information is relatively easy to find and is presented on many websites categorized by ODP. Thus, a user profile that ensures a high degree of personalization on these topics can be created in a small amount of time. For one city, the information had to be found with Google, for four cities with Prospector.

For the Google searches, and the last (fully personalized) Prospector searches, for a youth hostel and a museum of modern art, the participants had to rate the perceived relevance of the five highest ranked search results on a 7-point Likert scale, ranging from very irrelevant to very relevant. In the case of the youth hostel search the average Google relevance score (M = 5.24, SD = .92) did not differ from the average, fully personalized, Prospector relevance score (M = 5.08, SD = .96), $t(30)$ = .64, n.s. In the case of the museum of modern art search, the Google relevance score, (M = 3.69, SD = 1.24) did differ from the fully personalized Prospector relevance score (M = 2.78, SD = 1.09), $t(31)$ = 3.93, p <.001. Note that the latter scores are on the negative side of the scale, indicating that this appeared to be a difficult search for the participants.

The thinking-aloud protocols provided us with several causes for these results, the most important one being the effects of the ODP category ‘news’, which appeared to cause problems for the generation of appropriately personalized results. Many users had a high interest in news and scored it as such on their interest page. However, when they search for information that is not related to news (e.g., a youth hostel in Oslo) they are presented with search results focusing on (possibly outdated) news which interfere with their search goal. This happened particularly during the searches for museums of modern art. Although there is no easy solution to this problem, a possible approach would be to categorize news only as such on the basis of the age of the respective page (e.g., not older than 10 days); this would be more in line with users’ perception of ‘news’.

Other results derived from this study include the following:

– The current incarnation of the rating frame above each opened search result appears to feel “unnatural” to some users. Participants forgot to use it as they are used to working with the browser’s back button, or did not use it correctly. For instance, they went back with the ‘Result OK! Take me back’ option and then removed the site with the ‘Unsuitable’ button. Consequently, the generation of the user model suffered from incorrect and / or missing usage data.

– The ODP categories, associated with a site, are sometimes wrong, which may influence the success of personalization. Sometimes, this is also affected by the exact URL used to refer to a site. For instance, the URL “en.wikipedia.org” is associated with 5 categories (i.e., “Arts / Television / Programs / Children's / Sesame Street”). None of these is indicative of the fact that Wikipedia is a free, on-line encyclopedia. In contrast, the URL “en.wikipedia.org/” (note the slash at the end of the URL), is associated with only one category: “Computers / Open Source / Open Content / Encyclopedias / Wikipedia”. This issue resulted in users having a misrepresentation of (dis)interests in their user models, with ratings being assigned to wrong or irrelevant categories. This, in turn, sometimes resulted in unexpected results, with low relevance to the user’s real interests.
– The wording of ODP categories brings along some serious problems when users indicate their interests or alter their user profile. Participants said they found the categories vague and needed more information about their meaning in order to indicate their interests or alter their profile properly. As a result, the success of personalization suffers: people may interpret categories wrong and create an incorrect user profile.

6. Conclusions and discussion

This paper has presented the second version of the Prospector system, which uses thematic ontologies from the ODP project, and a set of custom probabilistic algorithms for the adaptive reordering of search results on the basis of user- and group- models. Evaluation results have been very valuable in the iterative design process, leading to the currently under development third generation of Prospector, which addresses most of the issues mentioned in the previous section.

The proposed approach seeks to combine two sources of information for improving on the relevance of search results: (a) semantic information about the result items, and (b) individual- and group oriented- preferences and interests in thematic categories that directly relate to the aforementioned semantic information. Furthermore, it has been designed to operate at a meta-search level, being largely agnostic of the underlying search engine, and of the attributes of the items being searched (other than their thematic categories of course, and the relationships between those). These characteristics make this approach applicable “on top” of any user-oriented search system that maintains such semantic information for the items in its index, including multimedia search systems. The strengths of the approach lie with: the involvement of the user community in establishing the affinity of specific items to specific themes; and, the fact that the quality of system results increases with the number of individuals that use it, and with the number of ratings made by each individual.

The work we have done so far in applying this approach on open corpus document search with Prospector has provided a number of valuable lessons that should be heeded in any future work in this direction:

– The approach works best in contexts where disambiguation (due, e.g., to synonymity / homonimity issues) is required to provide relevant search results.
– The quality of the semantic information available for searchable items is of paramount importance. In this context, the correctness of existing semantic information is more central to the operation of the system, than the non-existence of said information. In other words, we have seen that the system can "cope" with missing pieces of categorization information better than with erroneous ones.
– The dependency on the availability of valid semantic information may render this approach inappropriate for employment in open corpora with little (if any) such information present. A better fit is envisaged for closed corpora, where semantic information is already available, or can be readily and progressively provided by the users of the search system. This would include systems where users can "tag" items in a structured or semi0structured manner.

References

1. Brin, S., and Page, L. (1998). The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems*, 30(1– 7):107–117, 1998.
2. Chirita, P. A., Nejdl, W., Paiu, R., and Kohlschütter, C. (2005). Using ODP Metadata to Personalize Search. In *Proceedings of the 28th ACM International SIGIR Conference on Research and Development in Information Retrieval*, Salvador, Brazil, August 2005. ACM.
3. Dell Zhang, Y.D. (2004). Semantic, Hierarchical, Online Clustering of Web Search Results. In *Proceedings of the 6th Asia Pacific Web Conference (APWEB)*, Hangzhou, China, 2004.
4. Hamilton, N. (2003). The mechanics of a deep net metasearch engine. In *Proceedings of the 12th International World Wide Web Conference*, Budapest, Hungary, 2003.
5. Kleinberg, J.M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.
6. Lawrence, S. (2000). Context in Web Search. *IEEE Data Engineering Bulletin*, 23(3):25–32, 2000.
7. Schwendtner, C., König, F., and Paramythis, A. (2006). Prospector: An adaptive front-end to the Google search engine. In *Proceedings of the 14th Workshop on Adaptivity and User Modeling in Interactive Systems (ABIS 2006)*, held in the context of Lernen-Wissensentdeckung-Adaptivität 2006 (LWA 2006) (ISSN: 0941-3014), pp.56-61, University of Hildesheim, Hildesheim, Germany, October 9-11, 2006.
8. Smyth, B. Balfe, E. Briggs, P. Coyle, M., and Freyne, J. (2003). Collaborative Web Search. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, pages 1417-1419, Acapulco, Mexico, August 2003. Morgan Kaufmann.
9. Smyth, B., Freyne, J., Coyle, M., Briggs, P., and Balfe, E. (2003). I-SPY: Anonymous, Community-Based Personalization by Collaborative Web Search. In *Proceedings of the 23rd SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 367-380, Cambridge, UK, December 2003. Springer.
10. Tanudjaja, F., and Mui, L. (2002). Persona: A Contextualized and Personalized Web Search. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 3*, pages 67 (9), Hilton Waikoloa Village, Island of Hawaii, January 2002. IEEE Computer Society.