DEWS2007 M5-5

Personalizing Web Search via Modelling Adaptive User Profile

Lin LI† and Masaru KITSUREGAWA†

† Institute of Industrial Science, The University of Tokyo
Komaba 4–6–1, Meguro-Ku, Tokyo, 153-8305 Japan

Abstract Personalized search has become an active on-going research field. Recent studies have stated that user interests could be learned automatically. As far as we know, these studies, however, neglect the changes of user interests. In this paper, we introduce an adaptive scheme to learn these changes from click-history data, and a novel rank mechanism to bias the search results of each user. A taxonomic hierarchy for modelling the user profile represents the user interests. Adaptation strategies are devised to capture the accumulation and degradation changes of user interests, and adjust the content and structure of the user profile to these changes. Experimental results show that a rank mechanism based on this scheme yields greater improvement over the compared rank mechanisms.
Key words personalized search, user profile, experiments

1. Introduction

Present search engines generally handle search queries without considering the contexts in which users submit queries. As a result, it becomes more difficult to obtain desired results than ever due to the ambiguity of user's needs. For example, suppose that a information retrieval researcher who wants to search information about Text Retrieval Conference and a engineer who is interested in taking advantage of the truly enormous quantities of solar energy falling, both input "TREC" on Google. Regardless of different intentions of the two users on the same query, the results turn out to be an official site of the Texas real estate commission, training resources for the environmental community, a site about educational research experiences, and so on. Current search engines are inadequate for making a difference among the various information needs of the users.

Studies [2], [9] on personalized search have focused on requiring users to explicitly enter their contextual information including interest topics, bookmarks, etc., and using these contextual information to expand users' queries or re-rank search results. Forcing users to submit their contextual interests would be a task that few users would be willing to do. Furthermore, it is very difficult for users to define their own contextual interests accurately. Much attention has been paid in [8], [11], [13], [14] to learn user interests automatically by modelling user profiles or user representations. Speretta et al. [13] creates user profiles by classifying information into concepts from the ODP(注1) taxonomic hierarchy and then re-ranks search results based on the conceptual similarity between page and user profiles. They, however, have not taken the hierarchy structure of the ODP into account when calculating the conceptual similarity.

In this paper, we emphasis on learning user profiles and utilizing the learned user profiles to re-rank search results. Most studies on learning user profiles have deemed user profiles to be static. A related problem occurs when user interests change over time. For instance, if a user changes her vocation from being an IT specialist to a lawyer, it is natural that her interests will shift with this change. It becomes important to keep the user profile up-to-date, and for a search engine to adapt accordingly. Therefore, suitable strategies are needed to capture the accumulation and degradation of changes of user interests, and then adapt the contents and structures of the user profiles to these changes. For re-ranking search results, our rank mechanism is similar to that proposed by [2] in which a semantic similarity measure is introduced for web page rank with consideration to the hierarchy of the ODP structure. Meanwhile, the technique proposed in [2] suffers from the problem of requiring users to select topics which best fit their interests from the ODP, and other shortcomings we will address in Section 3.2.

Our contributions in this paper could be summarized as:

(1) Adaptation strategies for modelling user profiles automatically are proposed. These strategies are based on click-history data while considering the accumulation and degradation changes of user interests.

(2) When user interests change, our user profiles, not only in contents, but also in structures, are modified to adapt to the changes.

(注1)：http://dmoz.org

(3) Finally, we propose a novel rank mechanism by measuring hierarchy semantic similarities between up-to-date user profiles and web pages. About 29.14% average improvement is gained over existing rank mechanisms.

The rest of this paper is organized as follows. In Section 2, we review the related work. In Section 3 we describe the model and adaptation strategies for the user profiles, and rank mechanisms. Section 4 presents the experimental results. Finally, we conclude in Section 5 with some directions for future work.

2. Related Work

2.1 Personalized Search

As we know, if the context information is provided by an individual user in any form, whether automatically or manually, explicitly or implicitly, the search engine can use the context to custom-tailor results. This process is named as a personalized search.

In this way, such a personalized search could be either server-based or client-based. The system in [4] is an available server-based search engine that unifies a hierarchical web-snippet clustering system with a web interface for the personalized search. Google and Yahoo! also supply personalized search services. With the cost of running a large search engine already very high, however, it is likely that the server-based full-scale personalization is too expensive for the major search engines at present.

On a client-based personalized search, studies in [3], [11], [14] focus on capturing all the documents edited or viewed by users through computation-consuming procedures. Allowing for scalability, the client-based personalized search could learn user contexts more accurately than the server-based personalized search, while it is unavoidable that keeping track of user contexts has to be realized by middleware in the proxy server or client. Users, however, may feel unsafe to install such a software even if it is guaranteed to be non-invasive, and may intend to enjoy the services provided by search engines instead. Moreover, if a user changes her computer from her office to home, keeping her contexts consistent becomes a problem.

In this paper, we focus on the use of suitable strategies to learn user profiles in a trade-off between scalability and accuracy for the server-based personalized search.

2.2 User Profile

There have been vast schemes of learning user profiles to figure user interests from text documents. We found that most of them model user profiles represented by bags of words like [1], [5], [12], [15] without considering term correlations. To overcome the drawbacks of the bag of words, the taxonomic hierarchy, particularly constructed as a tree structure, has been widely accepted in [2], [7], [10]. SchickelZuberF et al. [10] score user interests and concept similarity based on the structure of ontology. But their work needs users to express their interests by rating a given number of items explicitly.

Meanwhile, these studies neglect that user interests could change with time. Some topics will become more interesting to the user, while the user will completely or to varying extent, lose interests in other topics . Studies in [1], [5], [15] suggest that relevance feedback and machine learning techniques show promise in adapting to changes of user interests and reducing user involvements, while still overseeing what users dislike and their interest degradation. In [15] a multiple three-descriptor representation is introduced to learn changes in multiple interest categories, and it also needs positive and negative relevance feedback provided by users explicitly.

Our work, particularly our adaptation strategies for user profiles, are based on the idea that sufficient contextual information is already hidden in the web log with little overhead, and all the visited pages can be considered as user interests to various degrees because the users have accessed them. This contextual information motivates us to capture the accumulation and degradation changes of user interests implicitly, to learn user profiles automatically.

3. Personalized Web Search

3.1 Adaptive User Profile

3.1.1 Model for User Profile

The model for our user profile is a taxonomic hierarchy, a part of the Google Directory(注2). This part is composed of topics that have only been associated with the clicked search results. It is also called the user topic tree, for these topics are linked as a tree structure. Each node in the user topic tree means a topic in the Google Directory, and has a value of the number of times the node has been visited. For simplicity, we call this value the "*TopicCount*" that represents the degree of interests. Figure 1 illustrates the schema of a user profile. For example, node C is represented by the $[Internet, 18]$ which means one user has clicked a page associated with the topic "*Internet*" and the user has visited the "*Internet*" 18 times before this search. In our experiments node C is actually stored as the $[\backslash Root\backslash Compuetr\backslash Internet, 18]$ with a full path in the Google Directory.

3.1.2 Adaption Strategies for User Profile

Our adaption strategies for user profiles include two operations, the "adding" and "deleting" operations. In the Google Directory, each web page is associated with a topic.

(注2)：http://directory.google.com

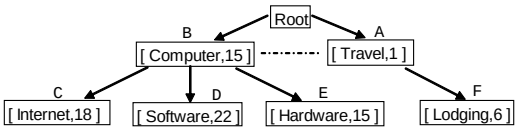

Figure 1 Schema of user profile

When dealing with the "adding" operation, topics associated with the clicked pages , and not all the search results, are added into the user topic tree click by click. The value of "*TopicCount*" is also accumulated by increments.

The "deleting" operation captures the degradation changes of user profiles. We begin with two examples demonstrating the intuition behind considering the degradation changes.

Example 1: User interests in current hot topics could change on a day-to-day basis.

Example 2: User click-history data are usually noisy. After a user clicks and browses a page, if she thinks that this page is not interesting to her, this click behavior will interfere with modelling her profile.

In both cases the nodes associated with the clicked pages have already added into the user topic trees. But these nodes do not represent the current user interests. To eliminate these noisy data, we periodically check all the nodes in the user topic tree. If the weight of "*TopicCount*" (i.e.,WT(i) described in Section 3.2.3) for one node becomes smaller than a threshold (i.e., 0.01), the node would be deleted from the tree. This "deleting" operation degrade user interests evenly after a period of time during which the user has not accessed some nodes.

The "adding" and "deleting" operations dynamically adapt the structures and contents of the user profiles to the user click behaviors.

3.2 Rank Mechanisms

3.2.1 Distance metric

The distance with what we deal, is the distance between each search result and the user topic tree, as described in [2]. The search result with the shorter distance, meaning the higher similarity to user interests, should be put in the topmost position of the ranking list. For each search result, there is an associated node in the Google Directory. The user topic tree is also composed of nodes. The distance computation is actually how the tree distance between two nodes in the tree structure is measured.

Chirita et al. [2] point out that the main drawback of the naïve tree distance is that it overlooks the depth of the subsumer (the deepest node common to two nodes). With the help of Figure 1, let us explain the problem clearly. $sub_{i,j}$ represents the subsumer of the node i and the node j. $E(i, sub_{i,j})$ represents the number of edges between the node i and the node $sub_{i,j}$. The naïve distance is defined as

$$D(i,j) = E(i, sub_{i,j}) + E(j, sub_{i,j})\,. \tag{1}$$

By applying Equation (1), $D(A,B)$ is 2, which is the same as $D(C,D)$, making it difficult to re-order search results.

3.2.2 Hierarchy Semantic Similarity

Li et al. [6] try to tackle this issue by extending Equation (2), which takes the depth of the subsumer h and the naïve distance l between two nodes into the calculation. α and β are the parameters scaling the contribution of the naïve distance and the depth respectively. The semantic similarity is defined as

$$S(i,j) = e^{-\alpha \cdot l} \cdot \frac{e^{\beta \cdot h} - e^{-\beta \cdot h}}{e^{\beta \cdot h} + e^{-\beta \cdot h}}\,, \quad \alpha \geqq 0\,, \beta > 0\,. \tag{2}$$

In [6], the experiment results show that the optimized values of the two parameters are, α=0.2 and β=0.6. For example, $S(A,B)$ is unequal to $S(C,D)$ based on Equation (2). Because the subsumer of A and B, i.e., "*Root*", is in the different level from the subsumer of C and D, i.e., "*Computer*'. However, Equation (2) only solves problem partially. Let us see another example. Due to the same value (i.e., 3) between $D(A,C)$ and $D(B,F)$, and the same subsumer (i.e., "*Root*") between the pairs (A,C) and (B,F), $S(A,C)$ is equal to $S(B,F)$.

Under this situation, Chirita et al. [2] separate l into l_1 and l_2, and then gives different weights to the two variables through the parameter δ in Equation (3). The extension of Equation (2) is defined as

$$S^*(i,j) = ((1-\delta) \cdot e^{-\alpha \cdot l_1} + \delta \cdot e^{-\alpha \cdot l_2}) \cdot \frac{e^{\beta \cdot h} - e^{-\beta \cdot h}}{e^{\beta \cdot h} + e^{-\beta \cdot h}}\,. \tag{3}$$

Equation (3) can work well for common cases. However, we find that the parameter δ in Equation (3) is sensitive to the semantic meanings between the two topics, as illustrated in [2]. Furthermore, even if we compute the similarity by Equation (3), $S(C,D)$ is still equal to $S(E,D)$ because of the same value between l_1 and l_2. In our system, we extend Equation (2) in another way, as the "*TopicCount*" has much better effect on the overall performance than the weak parameter δ. Comparative experiments are in Section 4.

3.2.3 Our Rank Mechanism

When a user submits a query to the search engine, the search results are re-ranked by our semantic similarity in Equation (4), the degree by which the search result is similar to the user profile. i is a node in the user topic tree ($i = 1, 2, \cdots, size(UserTopics)$). j is the associated node with a search result in the Google Directory ($j = 1, 2, \cdots, size(Results)$). $WT(i)$ weighs the degree of interests of a node in the user topic tree, defined as

$TopicCount(i)/\sum_{i=1}^{size(UserTopics)} TopicCount(i)$.

The larger the WT is, the more interested the user is in one topic. One user topic tree represents one user. Hence, we define the semantic similarity between one search result and the user topic tree as the maximum value among all the values ($i = 1, 2, \cdots, size(UserTopics)$). The equation is

$$CS^*(User, j) = Max(WT(i) * S(i, j)) . \quad (4)$$

To keep our rank mechanism from missing the high quality pages in Google, Equation (4) is integrated with PageRank (PR) as in Equation (5). Here γ is a parameter in [0,1] which blends the two ranking measures. The user could vary the value of γ to merge our rank mechanism and PageRank in different weights. In our experiments, γ is set to 0.5, which gives equal weight to the two measures.

$$FR(User, j) = (1 - \gamma)CS^*(User, j) + \gamma * PR(j) . \quad (5)$$

4. Experiments

4.1 Evaluation Metric

In terms of the user satisfaction, an effective rank mechanism should place relevant pages close to the top of the rank list. We ask the users to select the pages they considers relevant to their interests for our evaluation. The quality of our system is measured as Equation (6):

$$AveRank(u, q) = \sum_{p \in S} (R(p))/Count(p) . \quad (6)$$

Here S denotes the set of the pages selected by user u for query q, $R(p)$is the position of page p in the ranking list, and $Count(p)$ is the number of selected pages. A smaller $AveRank$ represents a better quality.

4.2 Experimental Setup

Our rank mechanism could be combined with any search engine. In this study we choose the Google Directory search as our baseline in that Google applies its patented PageRank technology on the Google Directory to rank the sites based on their importance. It is convenient for us to combine and evaluate our rank mechanism with Google. Main modules in the experiments are listed as follows:

· Google API module: Given a query, we are offered titles, snippets, and page-associated Google directories beside the URLs of web pages by the Google API(注3). Our paper regards a Google directory as a topic in the user topic tree.

· Log module: We monitor user click behaviors, recording the query time, clicked search results, associated topics.

· User profile: It has been described in Section 3.1.

The necessary steps are depicted as follows:

1) Issuing the query submitted by an online user through the Google API module ;

2) Re-ranking search results by our rank mechanism based on the current user profile and then going into the Log module;

3) Adapting the user profile to click-history data provided by the Log module through our strategies:

Updating the structure and the degree of interests by the "adding" operation;

If needed, degrading the model by the "deleting" operation.

4) Waiting until the online user submits a new query, and then going to 1).

The topics in the different levels of Google Directory consist of user topic trees. Since the number of levels to add to the user profile is unknown, in our experiments, the first step is to determine the number of levels by a preliminary analysis of system performances on different levels. Based on the results of this analysis, we then construct user profiles and re-rank search results.

4.3 Dataset

For each search, the Google API module got the order of the top 20 Google results due to the limited number of the Google API licenses we have. We randomized the order of the results before returning the 20 results to the user at run-time. For evaluation, 12 subjects are invited to search through our system. The 12 subjects are graduate students (5 females and 7 males) researching in several fields, i.e., computer, chemistry, food engineering, electrical engineering, art design, medical, math, architecture, and law.

Our search interface was available on the Internet, and convenient for the subjects to access it at any time. They were asked to query topics closely related to their interests and majors. In the first four days, subjects input the queries on their majors, and then in the next three days the queries on their hobbies were searched. Finally, in the last three days, the subjects were required to repeat some queries done before. This repeated procedure gave a clear performance comparison between the current and earlier systems, as user profiles were updated search by search. After the data were collected over a ten-day period (From October 23nd, 2006, to November 1st, 2006), we got a log of about 300 queries averaging 25 queries per subject and about 1200 records of the pages the subjects clicked in total.

4.4 Experimental Results

4.4.1 Number of Levels for User Profile

As described in Section 3.1, a user profile was created by categorizing each search result and accumulating the re-

(注3)：http://code.google.com/apis/soapsearch

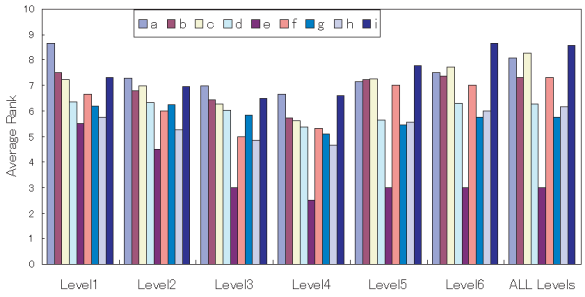

Figure 2 Number of Levels for User Profile

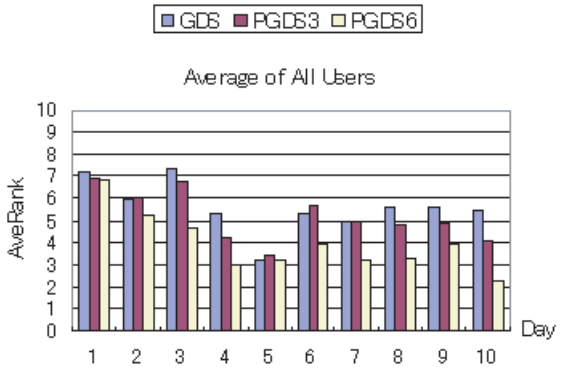

Figure 3 Quality of Personalized Search System (Lower is better)

turned topics and weights. One question that needs to be resolved was, since the Google API returns an ordered list of topics associated with search results, how many of these levels per search result should be used to create the profile and then update it.

To investigate this question, we randomly selected 9 subjects and performed a detailed analysis of the levels. For each subject, the top 20 search results returned by the Google API were manually judged as relevant. Figure 2 illustrates that average rank (Equation (6)) per user versus the number of levels considered per user profile. It shows that the top 4 levels assigned per user profile yielded better overall performance. Thus, in the experiments that follow, we built the user profiles considering only the top 4 levels from the Google Directory.

4. 4. 2 Results of Quality of Personalized Search System

Now, we compare the performance improvements of the following three ranking mechanisms by using Equation (6).

· Google Directory Search (GDS), using the Google API

· Personalized Google Directory Search (PGDS3), combing Equation (3) and the PageRank

· Personalized Google Directory Search (PGDS6), using Equation (5)

Figure 3 illustrates the average improvement over all users day by day. As a result of requiring the subjects to change queries from their majors to hobbies, we see that from the fourth day to the fifth day, the values of AveRank experience a sudden increase. But after three days on learning the changes, our PGDS6 shows better results than the GDS and the PGDS3. More accurately, compared with the GDS, our PGDS6 outperforms the PGDS3 with a 60% improvement for the tenth day, while for the fifth day the improvement is only around 2%. This difference demonstrates that the changes of user interests will lower the improvement that our strategy could achieve. Nevertheless, our rank mechanism still greatly improves over the GDS and the PGDS3 overall. The average improvements of our PGDS6 and the PGDS3 over the GDS, are 29.14% and 7.36% respectively.

5. Conclusion

In this paper we introduced how to capture the changes of user profiles from click-history data and how to use the user profiles to re-rank the search results, thus creating personalized views of the web. First, we designed a hierarchy model for a user profile. Then, we adapted the user profile, including the content and the structure, to the accumulation and degradation changes of user interests by our adaptation strategies. Finally, we proposed a novel rank mechanism to re-rank search results. Experimental results on real data demonstrate that our dynamic adaptation strategies are effective and our personalized search system performs better than the selected rank mechanisms.

In the future, we plan to do some comparative experiments when the user varies the value of γ in Equation (5). In addition, when computing for the node distance in the tree, we plan to consider the edge distance, assigning a different weight for each edge, because each pair of two nodes linked by an edge has different semantic similarity.

References

[1] D. Billsus and M. J. Pazzani. A hybrid user model for news story classification. In *Proc. of the 7th Int'l Conf. on User modeling (UM'99)*, pages 99–108, Secaucus, NJ, USA, 1999.

[2] P. A. Chirita, W. Nejdl, R. Paiu, and C. Kohlschütter. Using ODP metadata to personalize search. In *Proc. of the 28th Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR'05)*, pages 178–185, Salvador, Brazil, 2005.

[3] S. T. Dumais, E. Cutrell, J. J. Cadiz, G. Jancke, R. Sarin, and D. C. Robbins. Stuff I've seen: A system for personal information retrieval and re-use. In *Proc. of the 26th Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR'03)*, pages 72–79, Toronto, Canada, 2003.

[4] P. Ferragina and A. Gulli. A personalized search engine based on web-snippet hierarchical clustering. In *Proc. of the 14th Int'l Conf. on World Wide Web - Special interest tracks and posters (WWW'06)*, pages 801–810, Chiba, Japan, 2005.

[5] W. Lam, S. Mukhopadhyay, J. Mostafa, and M. J. Palakal. Detection of shifts in user interests for personalized infor-

mation filtering. In *Proc. of the 19th Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR'96)*, pages 317 – 325, Zurich, Switzerland, 1996.

[6] Y. Li, Z. Bandar, and D. McLean. An approach for measuring semantic similarity between words using multiple information sources. *IEEE Trans. Knowl. Data Eng.*, 15(4):871–882, 2003.

[7] B. Markines, L. Stoilova, and F. Menczer. Bookmark hierarchies and collaborative recommendation. In *Proc. of The 21st National Conf. on Artificial Intelligence and the 8th Innovative Applications of Artificial Intelligence Conference (AAAI'06)*, Boston, Massachusetts, USA, 2006.

[8] F. Qiu and J. Cho. Automatic identification of user interest for personalized search. In *Proc. of the 15th Int'l Conf. on World Wide Web (WWW'06)*, pages 727–736, Edinburgh, Scotland, 2006.

[9] H. rae Kim and P. K. Chan. Personalized ranking of search results with learned user interest hierarchies from bookmarks. In *Proc. of the 7th WEBKDD workshop on Knowledge Discovery from the Web (WEBKDD'05)*, pages 32–43, Chicago, Illinois, USA, 2005.

[10] V. Schickel-Zuber and B. Faltings. Inferring user's preferences using ontologies. In *Proc. of The 21st National Conf. on Artificial Intelligence and the 8th Innovative Applications of Artificial Intelligence Conference (AAAI'06)*, Boston, Massachusetts, USA, 2006.

[11] X. Shen, B. Tan, and C. Zhai. Implicit user modeling for personalized search. In *Proc. of the 2005 ACM CIKM Int'l Conf. on Information and Knowledge Management (CIKM'05)*, pages 824–831, 2005.

[12] S. J. Soltysiak and I. B. Crabtree. Automatic learning of user profiles- towards the personalisation of agent services. *BT Technology Journal*, 16(3):110–117, 1998.

[13] M. Speretta and S. Gauch. Personalized search based on user search histories. In *Proc. of the IEEE / WIC / ACM Int'l Conf. on Web Intelligence (WI'05)*, pages 622–628, Compiegne, France, 2005.

[14] J. Teevan, S. T. Dumais, and E. Horvitz. Personalizing search via automated analysis of interests and activities. In *Proc. of the 28th Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR'05)*, pages 449–456, Salvador, Brazil, 2005.

[15] D. H. Widyantoro, T. R. Ioerger, and J. Yen. Learning user interest dynamics with a three-descriptor representation. *JASIST*, 52(3):212–225, 2001.