TOWARDS AN INTELLIGENT AND PERSONALIZED RETRIEVAL SYSTEM

Sung H. Myaeng and Robert R. Korfhage
Department of Computer Science and Engineering
Southern Methodist University

ABSTRACT. Development of an information retrieval system that can be personalized to each user requires maintaining and continually updating an interest profile for each individual user. Since people tend to be poor at self-description, it is suggested that profile development and maintenance is an area in which machine learning and knowledge base techniques can be profitably employed. This paper presents a model for such an application of AI techniques.

KEYWORDS: information retrieval, user profile, learning, personalized system

1. Introduction

In the context of conventional information retrieval systems (IRS), the search process is initiated and completed by a set of queries from a user. Each query, usually in the form of a vector or Boolean expression, consists of a set of key terms to be matched with the contents of relevant items. To improve the retrieval effectiveness, modification of the user query through the application of user feedback has been studied with some successful results [13].

There have also been systems, called selective dissemination of information systems (SDI), that selectively distribute incoming information to appropriate users based on user interest profile. However, only recently has a set of models been proposed that effectively combines the two different modes of the systems, thereby attempting to enhance the quality of retrieved items [3,8,9].

One of the major stumbling blocks in the conventional IRS is the problem of formulating a query which accurately matches the user's needs and the contents of potentially relevant items[1,12]. Unfortunately, different users expect different sets of items from the same query and make different relevance judgements on the same retrieved items, directly related to their individual needs. But the conventional retrieval system disregards the individual user's characteristics and the fact that diverse users have different perceptions of the underlying system. While it is natural that a user perceives the system in the light of his or her experience and needs, both the restricted structure of a query and the nature of the conventional system itself make this perception unavailable to the system. We believe that knowledge captured in a user profile embedded in the system will play an essential role in making a personalized system. One effect can be to retrieve a broader range of items, some of which would never be brought to the user's attention on the basis of the query alone. People prefer a librarian who can surprisingly provide information not explicitly requested but judged to be important to them. Profile information will also help the system tailor the retrieved items to a particular user's needs and rank them appropriately. Again, a friendly and intelligent librarian can eliminate some information which is not of the user's concern but would have been retrieved by a novice librarian who had to rely

totally on the user's request per se. Ultimately, our goal is to develop an IRS that is effective from the user's point of view and cooperative with the user in terms of achieving his or her goal.

Since we never guarantee that a user's characteristics and environment stay the same over time, it becomes necessary for the system to dynamically change the knowledge kept in the profile. Upon learning various aspects of a user's information needs and behavior, the system will use this information to respond in an intelligent and friendly manner. We elaborate on the concept of a dynamic user profile (DUP) with a learning strategy for modeling the DUP and discuss the heuristics and models that utilize the DUP. The next section shows how the system is configured as a learning system. Our main emphasis in this paper is in Section 3 where a strategy for learning users' interests and other characteristics is discussed. The rest of paper, showing the representation of the DUP, addresses the issues involved in the utilization of the DUP.

2. Overall System Configuration

We have developed a full retrieval system for the purpose of testing the validity and the sensitivity of the theoretical models with static profiles [8]. This base system can be modified to reflect the functions of DUP. Since our system should conduct learning, it is not surprising that its configuration is well projected on the synthesized model of learning systems proposed by Smith et al [14]. We adopt terms used in this model to show the function of each component in the system. The proposed model consists of six functional components: performance element, instance selector, critic, learning element, blackboard, and world model. The performance element uses the learned information to perform the stated task. The instance selector selects training instances from the environment of the learning system whereas the critic analyzes the current abilities of the performance element. The learning element, which is an essence of the learning system, is an interface between the critic and the performance element, responsible for translating the abstract recommendations of the critic into specific changes in rules or parameters used by the performance element. The blackboard is a global database used as a system communication means. It holds two types of information: the information in the knowledge base and the temporary information used by the the learning system components. Finally, the world model contains the fixed conceptual framework within which the system operates.

Documents in the database are assumed to contain key words with associated weights. These weights can be assigned on a frequency-related basis, as is quite standard in information retrieval. While it is possible to adjust the weights dynamically on the basis of user response, for present purposes we assume the weights are fixed.

In our system, as shown in the Fig.1, the query processor/responder is considered as a learning system performance element. It is the nucleus in conventional systems and, based on a query, actually retrieves items, providing the user with a set of items ranked on the basis of the weights in the query and items in the database. In our system design, this component also integrate the user-dependent information from the profile.

The profile controller serves mainly as a learning element with some additional functions taken care of by an instance selector and a critic. This component observes the interactions between the system and the user, selects useful instances, and makes specific changes to the profile and possibly the query in such a way that the system's performance will eventually approach the desired level.

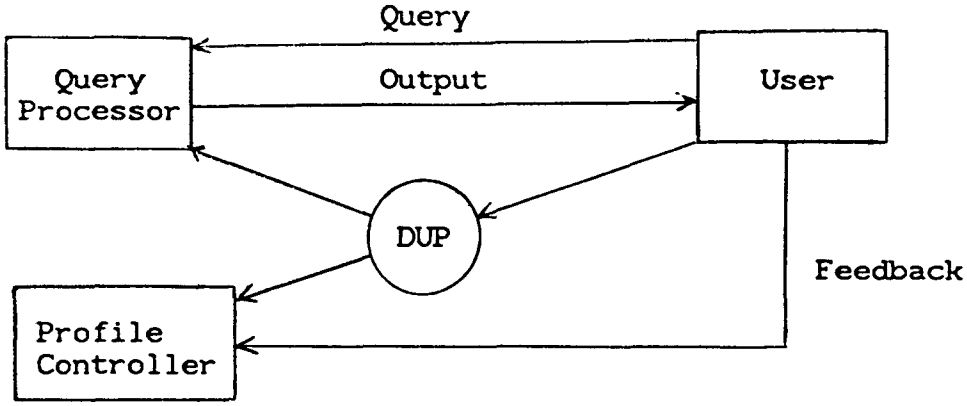

Fig. 1

In the context of an IRS, the role of a critic is performed primarily by human users although the statistics gathered through operation of the system can be of importance. Currently, the user's relevance feedback on the retrieved items is the only valuable information from the critic. Feedback information from each user is interpreted using the profile, and therefore part of the critic's role is transferred to the profile controller.

3. A Strategy for Learning

Our ultimate purpose in having the learning element is to build an IRS that incorporates an individual user's characteristics as much as possible, in an automatic and time-dependent manner. Although this can only be achieved by monitoring the user's interaction with the system, initial dialogue with each user is expected to play an important role in obtaining skeletal information that will provide a direction to the system's inference. Without this kind of information available, the uncertainty we have to deal with is so high that either we could never be sure that the system is on the right track in terms of learning, or the usability of DUP would be limited. This difficulty will arise especially with users whose background or interests lie in diverse fields and whose queries are not consistent with respect to a single field of interest.

Interest

The area of a particular user's interest represents a concept that should be maintained in DUP. Many knowledge or concept representation schemes have been developed [16,17], which are geared to solve domain-specific problems. However, the application of those symbolic representations is not feasible for the information retrieval environment, especially when we deal with an enormously large domain of concepts in a heterogeneous document database. Therefore, we adopt a vector model for the purpose of representing concepts such as an individual user's interest, a query, and the content of each document. With $d_{i,j}$, q_j, and p_j being the degree of importance of jth term in representing the concept described by ith document, the query, and the profile, respectively, we use the following vector notations:

$$\mathbf{D}_i = <d_{i,1}, d_{i,2}, \ldots, d_{i,j} \cdots, d_{i,n}>$$

$$\mathbf{P} = <p_1, p_2, \ldots, p_j, \ldots, p_n>$$

$$\mathbf{Q} = <q_1, q_2, \ldots, q_j, \ldots, q_n>$$

where $0 \leq d_{i,j} \leq 1$ and $-1 \leq p_i q_j \leq 1$.

As described in Section 2, the role of the critic in learning systems is essential. In the information retrieval context, user feedback, as well as the queries, serves as a crucial source of information that aids learning process. Although relevance feedback has been of traditional use for query modification, we believe that more extensive use of feedback is not only possible but also fruitful. In our work, we make a distinction among relevance, pertinence, and usefulness that will be used by a user to determine his degree of satisfaction based on the retrieved set of documents. Relevance is to be judged more or less objectively on how reasonable a retrieved document is in response to the stated query. This measure is believed to measure the system performance. In contrast, pertinence is to be judged on closeness of the concept of a particular document with respect to the concept of a query the user intended. This more subjective measure compares a response to the hypothetical document the user expected to get at the time of the query formulation. Low pertinence is related to poor system performance or bad query formulation. On the other hand, usefulness is to be judged based on the user's long-term and short-term interest, regardless of the concept embedded in the query formulated. Thus a pertinent document will generally be relevant, but a relevant document may not be pertinent (if, for example, the user already knows about it). Similarly, any relevant or pertinent document will generally be useful, but a stray retrieved document that is neither relevant nor pertinent may still be useful, that is, related in some other way to the user's general interest. In order for users to make a distinction among the three different judgements operationally, the following types of questions can be asked to obtain the information on relevance, pertinence, and usefulness, respectively:

- Is the document a reasonable one to expect in response to the query?
- Is this really what you wanted to retrieve by asking the question?
- Is the document related to your general interest?

The first of these questions is related to the validity of the information model, while the second relates to the user's ability to properly formulate a query. The third question covers serendipitous discovering of unrequested information. To avoid any biased interpretation of the above questions, it seems critical that a user answer them only after all questions have been read and understood correctly.

Given the vector model and the three feedback measures, the mechanism of interest learning can be characterized as two processes. While we have to adjust the importance factor of each term taking part in describing the concept of the user's interest, we also need to introduce new terms with appropriate importance factors. Furthermore, considering that the relevance judgement is only related to the system performance, and focusing only on the pertinence and usefulness judgements for the purpose of interest learning, there are four cases we have to handle somewhat differently on the basis of combinations of two different pieces of feedback information: a retrieved document can be pertinent and useful, impertinent but useful, pertinent but not useful, or impertinent and not useful. Although different strategies are employed for different cases, we first introduce a general formula that will be used whenever it is necessary that a particular query or document concept affect the profile.

To introduce new terms, we define a vector **R** for the reserve list. Elements in this vector are terms not included in the profile **P**, but which could potentially be included. An inclusion decision is based on the weight of a term in **R**. Obviously the inner product of two vectors, **R**·**P**, should be zero all the time. Van Melle introduced a formula for certainty factor calculation [15]. The following modification of the formula shows how **P** and **R** are changed after a cycle of query processing has been completed: a

query is requested, a set of documents are retrieved, and user feedback on document relevance, pertinence, and usefulness is obtained. Italics indicate the new value generated after the cycle, and x_j is the weight of term j in either the query or a document.

$$p_j = \begin{cases} p_j + x_j(1 - |p_j|) & \text{if } p_j \cdot x_j > 0 \\ (p_j + \alpha x_j) & \text{if } p_j \cdot x_j < 0 \\ f(r_j) & \text{if } p_j = 0 \\ p_j & \text{if } x_j = 0 \\ 0 & \text{if new } p_j < \delta, \end{cases}$$

where $\alpha \leq 1$ is a constant. This reflects our strategy that the importance factor of an existing profile term should be stable to some extent against some conflicting information. When a new term is to be introduced, i.e., $p_j = 0$, r_j is updated in a simple manner.

$$r_j = r_j + \frac{x_j}{\beta},$$

Here, β is a non-zero constant reflecting the learning strategy which determines how important a new piece of information should be in terms of introducing a new term to the profile. Then $f(r_j)$ is defined as follows;

$$f(r_j) = \begin{cases} r_j & \text{if } r_j \geq \delta \\ 0 & \text{if } r_j < \delta \end{cases},$$

where δ is a threshold for inserting a new term into the profile, which also determines the minimum importance factor for each term in the profile. When r_j is bigger than δ, the current value of r_j becomes the new value of p_j otherwise r_j is set to zero to maintain the inner product property. As is shown in the first set of formulae, the similar idea is used to remove a term with the importance factor below δ from the profile and return it to the reserve list.

If, as the first case, a retrieved document is determined to be both pertinent and useful, it can be inferred that, while the query was successfully formulated so that the system's response was satisfactory, the retrieved document was also well fitted to the user's interest. By assuming that a user's short-term interest (reflected in the current query) is always consistent with his long-term interest, we can safely allow the system to integrate the concepts shown in the query and the document into the profile by means of the formula. In other words, each term shown in the query and the document needs to be substituted for the x_j's. If consistency assumption doesn't hold, the use of the reserve list will alleviate the problem of integrating a concept far from the user's interest.

In the second case where a retrieved document is determined to be impertinent but useful, the action that the learning element has to take is similar to the previous case. We have no information on how well the query was formulated; either things like document indexing and the search process were not very effective although the query was reasonable, or the query just didn't reflect the user's need, or both. Consequently, we need to treat the terms in the retrieved document and query differently. Terms occurring in the document should be given a higher importance factor than those comprising the query. But these important factors should be no higher.

On the other hand, the third case, where the retrieved document is determined to be not useful but pertinent, is somewhat delicate. This case can arise when the user presents a query that is not related to his general interest. Although the system's performance was guided correctly by the query, and the query was properly formulated with respect to the user's hypothetical document, it may be that, for example, the query is on a topic of only momentary interest. Therefore, we may want to consider the terms in the query and document as carrying small amount of information about the interest, due to the uncertainty on why it was judged to be useless. With the similar line of reasoning, we will have to admit that no information is available in the last case as far as consolidation of the profile terms is concerned.

However, the useless document is a source of valuable information for a weakening process. That is, if the concept reflected by the useless document has been included in the profile, the importance factor of the concept should be lowered appropriately. This can be done by negating the value of the importance factor of each term and then applying the same formula. Since we are uncertain about the cause of the uselessness, the assignment of low important factor to such a term would be desirable.

Habits

In addition to the need to automatically capture the user's interest, knowing information regarding individual user's habits seems also necessary. Typically the following are recognized as learnable characteristics:

- Reading habits, i.e, preference on the kind of a document (e.g. theoretical vs. practical)
- Perception on feedback
- Preference on either high recall or high precision

The reading habits can be obtained by simply accumulating statistics. Given a multi-dimensional space on which each periodical can be plotted based on the general trend of its difficulty or practicality, for example, the learning element of the system extracts the user's preference along each dimension by observing his feedback on each document retrieved. If he assesses a document in JACM as pertinent and/or useful, for instance, the scale about his reading habit should move toward a more theoretical and difficult document group. The initial default value can be assigned based on each user's background information which is expected to be given to the system explicitly. It is to say that the higher education a user has received, the more theoretical and difficult documents he would tend to read. On the other hand, the more he is related to the industry, the more he might prefer a practical document to theoretical one.

A user's perception on feedback seems to have an implication for any learning strategies. Since feedback, as a critic, plays an essential role in learning, a user's general habits on how to assess a retrieved document along different criteria must be taken into account so that any individual bias can be eliminated. It is expected that a conservative user will tend to rate a smaller number of documents positively whereas a more liberal user will rate a larger number of documents positively. Therefore, the history on how a user has evaluated retrieved documents will be a useful source of information. This implication not only facilitates unbiased learning of a user's characteristics and interest but also makes it possible to measure system performance more accurately, taking the bias into account.

In the IRS environment, there has been a tradeoff between recall and precision. Therefore, it is desirable for the system to know individual's preference so as to either

employ a different search method or tailor the retrieved output to his preference. A valuable source of information is the query. We can infer that the more terms included and the more negative weights are used, the more specific documents the user wants, favoring high precision.

While these habits can change over time and in response to specific needs, we are assuming for our present study that any change in habit is gradual and has only a secondary impact on our results. This assumption seems reasonable within the context of a user working on one project over a period of perhaps six months to one year.

4. Representation

Since we are ultimately planning to develop the user profile as a knowledge structure representing each individual user's relevant characteristics, it seems important to decide on a proper knowledge representation scheme among various representational methods. As categorized in Rich's paper [11], two extremes are possible for maintaining user models: models specified explicitly by the system designer or by the users themselves vs. models inferred by the system on the basis of users' behavior. Due to the nature of the IRS and the state-of-art in artificial intelligence, it seems natural to take a strategy between the two extremes, i.e., profiles are constructed and modified by users themselves while automatic changes are invoked whenever possible. In this way, we avoid forcing users to change the profile as much as possible, achieving user-friendliness.

With this strategy of modifying the profile, a frame-based structure, often used in AI-based systems, seems to be a good choice in that some incomplete information about a user needs to be derived indirectly, based primarily on the heuristics and statistical information. That is, we will be able to fill empty slots based on interactions between a user and the system by exploiting such features as IF-NEEDED demons, inheritance mechanisms, and default reasoning [17]. For example, a user's frame can be connected to a background frame as shown in Fig. 2. If the value for the preference slot is not available, then it can be inherited from the background frame. The initial values for other slots for a user can be computed providing that a background frame has been identified.

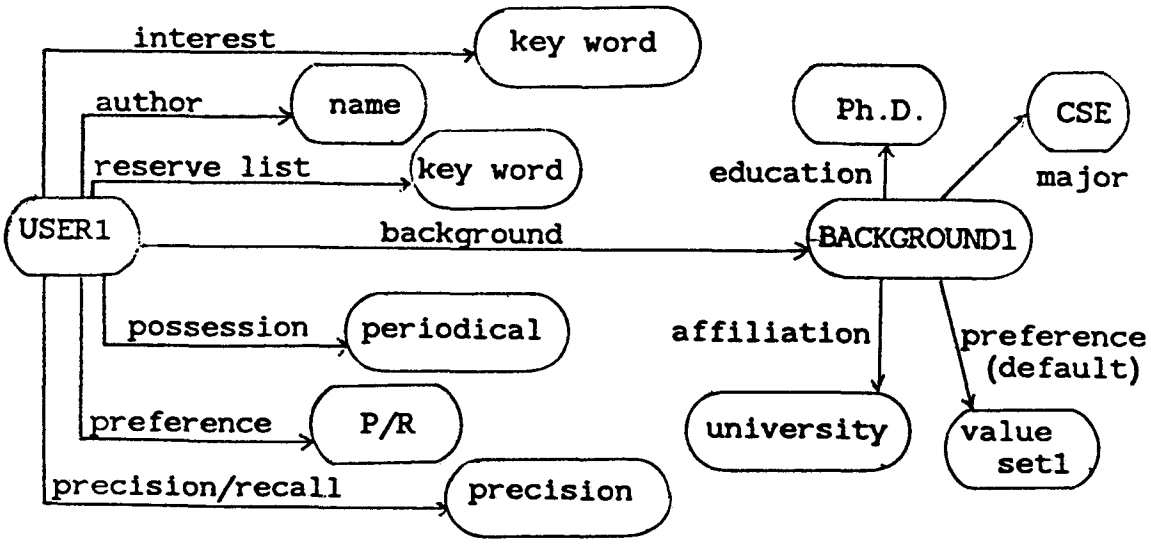

Fig. 2

5. Utilizing DUP

Assuming that the DUP always maintains correct information about a user through the learning process, the profile can be utilized in two ways: to enhance the retrieval effectiveness and to personalize the system, achieving better user-friendliness. Along the idea of applying the user profile to the query as a modifying factor, several models have emerged. One way to do this is to modify the query term by term, according to the occurrence and weight of each term in the profile, which reflect the user's interest. The other method is to consider the profile and query as separate and distinct points, and to judge each document considered for retrieval in terms of its relationships to both of these points [3,8,9]. Considering the enormous number of experiments to be done to test all different models and to draw conclusions, Korfhage developed the unified model frame to organize his expermental work [7].

On the other side of the coin is the possibility of exploiting a variety of the user's characteristics other than just interest. Since more uncertainty is involved in learning a user's habitual characteristics, a cautious use of this information seems necessary, not making any drastic change in terms of the retrieved set. Instead of adding or deleting documents in the basic retrieved list, we only intend to alter the importance factors of documents within the list so that different presentations of the output can be given to users. In this way, we can achieve user-friendliness in different level compared to that in ordinary interface level. One interesting question to be resolved is how to solve the problem of conflicting information, not in terms of learning but in term of different characteristic criteria. For instance, what decisions should be made in terms of adjusting the ranking if a document with a relatively high weights based on his interest belongs to one of very theoretical and difficult journals which he owns, but the user holds a Ph. D. and works for a production company?

6. Related Work

Since our effort has been devoted to explore the idea of combining and applying techniques in the realm of learning and user-friendliness to the area of information retrieval, a number of related works along different axes have been identified. The work done by Rich [10, 11] serves as an excellent guideline for building and manipulating general user models. Also in the area of user modeling is the Carbonell's introductory article addressing issues related to the design of natural language interface [2]. Like most of the work done in the area of user-friendly interface design, for example, user-driven interface [5], however, it focuses on user models from a designer's perspective rather than an actual system's perspective, deviating from our immediate concern. With the similar motivation but with different approach and philosophy, Corella et al. [4] have shown how to obtain cooperative responses to a database query in the form of natural language. No explicit user modeling was included in this work, but the effort to identify user's intention seems to deserve attention in relation to our work. An interesting but rather crude approach was proposed by Hause et al. in an attempt to build a self-tuning and adaptive information retrieval system [6]. Through the use of query-term thesaurus and by storing query-document matrix maintaining the processed queries in the past together with retrieved documents, the view of a frequent user of an IRS was extracted.

7. Concluding Remarks

We have explored the concept of the DUP with the stress on a strategy for learning. It is our belief that, without proper interpretation of a user's need and thus

personalization of the underlying IRS, overcoming the bottleneck of performance in such system is unrealistic until a breakthrough in natural language processing occurs. In addition, an attempt to achieve the true user-friendliness can be successful only when we are concerned about the quality and the kind of output as well as the ease of issuing commands. Obviously one way of approaching this end is to use the DUP. Nevertheless, the only way to prove the importance of exploiting the DUP for the enhancement of the system effectiveness and friendliness is through a set of experiments. By doing so, we can also identify more and better strategies for learning and exploiting information in the DUP.

REFERENCES

1. Blair, D.; Maron, M. Evaluation of Retrieval Effectiveness for a Full-Text Document-Retrieval System. CACM Vol. 28, No.3. March, 1985, 289-299.
2. Carbonell, J. G. The Role of User Modeling in Natural Language Interface Design. In Applications in Artificial Intelligence. Andriole, S. J. (Ed.). Petrocelli Books, Inc., Princeton, New Jersey, 1985, 213-226.
3. C. G., Hector. A User Profile-Query Model for Document Retrieval. Ph. D. Dissertation, 1982. Southern Methodist University, Dallas, Texas.
4. Corella, F.; Kaplan, S.; Wietherhold, G; Yesil, L. Cooperative Responses to Boolean Queries. International Conference on Data Engineering. April, 1984, 77-85.
5. Good, M. D.; Whiteside, J. A.; Wixon, D. R.; Jones, S. J. Building a User-Derived Interface. CACM Vol. 27, No. 10. Oct 1984.
6. Hausen, Hans-Ludwig. Outline of a Dynamic Self-Tuning and Adaptive Information Retrieval System. SIGIR Forum, XVI, 1. Summer, 1981, 132-144.
7. Korfhage, R. R. Unified Models for Profile-Query Interaction in Information Retrieval. SMU Technical Report CSE-85-16, Department of computer Science and Engineering, Southern Methodist University, Dallas, Texas. October, 1985.
8. Korfhage, R. R. Query Enhancement by User Profiles. Proc. ACM/BCS Symposium on Research & Development in Information Retrieval, 1984, 111-121.
9. Liu, Jung. A distance approach toward an Ideal Information Retrieval System. M. S. Thesis, 1982. Southern Methodist university, Dallas, Texas.
10. Rich, E. A. Building and Exploiting User Models. Ph.D. Thesis, 1979 Carnegie-Mellon University, Pittsburgh, Pennsylvania.
11. Rich, E. A. Users Are Individual: Individualizing User Models. Int. J. Machine Studies 18. 1983, 199-214.
12. Salton, G. The Use of Extended Boolean Logic in Information Retrieval. SIGMOD Record, Vol.14, No.2, 1984, 277-285.
13. Salton, G and McGill, M. J., Introduction to Modern Information Retrieval. Mcgraw-Hill, New York, 1983.
14. Smith, R. G. and et al. A Model for Learning Systems. IJCAI 5, 1977.
15. Van Melle, William. A Domain-Independent System Aids in Constructing Knowledge-Based Construction Program. Tech. Report STAN-CS-80-820, Stanford University, 1980.
16. Wasserman, K. Physical Object Representation and Generalization: A Survey of Programs for Semantic-Based Natural Language Processing. The AI Magazine, Vol. 6, No. 4. Winter, 1985, 28-42.
17. Winston, P. H. Representing Commonsense Knowledge. In Artificial Intelligence. Addison Wesley, Reading, Massachusetts, 1984.