METIORE: A Personalized Information Retrieval System

David Bueno[1], Amos A. David[2]

[1] Department of Languages and Computer Science, University of Málaga, 29071, Málaga, Spain.
bueno@lcc.uma.es
[2] LORIA, BP 239, 54506 Vandoeuvre, FRANCE.
adavid@loria.fr

Abstract. The idea of personalizing the interactions of a system is not new. With stereotypes the users are grouped into classes where all the users in a class have similar characteristics. Personalization was therefore not on individual basis but on a group of users. Personalized systems are also used in Intelligent Tutoring Systems (ITS) and in information filtering. In ITS, the pedagogical activities of a learner is personalized and in information filtering, the long-term stable information need of the user is used to filter incoming new information. We propose an explicit individual user model for representing the user's activities during information retrieval. One of the new ideas here is that personalization is really individualized and linked with the user's objective, that is his information need. Our proposals are implemented in the prototype METIORE for providing access to the publications in our laboratory. This prototype was experimented and we present in this paper the first results of our observation.

1 Introduction

The idea of personalizing the interactions of a system is not new. One of the approaches is the concept of stereotype [16; 17]. With stereotypes the users are grouped into classes where all the users in a class have similar characteristics. Systems interaction and response are based on the knowledge of the characteristics of the users in a class. Personalization was therefore not on an individual basis but on a group of users. This approach was however refined to take into consideration information that is specific to a particular user. This concept has been implemented in many systems in many application areas. In the area of Intelligent Help System, the concept was used in the Unix Consultant, KNOME [8] for the use of Unix. KNOME provides answers according to the stereotype where the system had placed that user after a few interactions. Other personalized systems in the area of ITS are ELM-ART [6; 21] and InterBook [5; 7]. A lot of work has also been done in the area of Information Filtering, from personalized newspaper [3; 9], news filters [4] or e-mail filtering.

The techniques used in Information Filtering can be easily adopted in Information Retrieval because these two fields are very close as [13] said, “The information filtering problem can be seen as a information retrieval process, where the user has a number of long-term queries”. Most of our ideas have therefore been used either in tutoring or in information filtering systems.

Our proposals have been implemented in a prototype called METIORE. The system allows users to access the bibliographic references of research publications available at the library of the computer laboratory center LORIA, France. In the following sections, we first present the functional characteristics of METIORE followed by a description of our user model. We have also carried out an experimentation of the prototype. Our observation during this experimentation is presented in section 4 before presenting a brief conclusion and our prospective. Most of the examples we give in this paper are from METIORE.

2 The Functional Characteristics of METIORE

For a better understanding of how personalization is done in METIORE, we first present in this section the main functional characteristics of the system.

2.1 The User’s objective

When the user launches the system we ask him to specify his *objective* for the session. The use of the system for a given objective is considered a session. Also in WebWatcher [1] the concept of *objective* (also called goal) is used. In their approach, the keywords of the goal are used to look for the links that the system will propose to the user as interesting. For us the objective is the expression of the user’s information need formulated in natural language. Presently, we do not process the text in natural language to extract the keywords even though it constitutes our research objective in order to calculate similarities between objectives. The objective is presently used to group the set of queries, concepts and decisions that the user makes on the system having the objective in mind. Our hypothesis is that grouping the user’s interactions into objectives will help the user to find information in his history and help the system to build a specific model of the user from one or more sessions. Our approach of personalization is based on this concept of objective. Every aspect of the system is organized around the system’s proposals and the user’s interactions.

This concept is very important to us since we believe that every user has a minimum knowledge of his information needs before attempting to use an IRS. This knowledge can of course be improved with the use of the system and with a consequence on the user’s ability to express his information need through the system’s interaction. For us, the queries do not necessarily express all the information needs of the user but rather his approach towards solving the problem of his information needs.

2.2 The Search Functions

For the user to obtain solutions to his information needs, we provide several search functions in the system. The search interface allows the user to make simple or complex queries. Search queries are defined using the attributes of a publication such as title, authors, year, keywords, etc. One of the problems in IRS is that information retrieval is content based. This means that a user must have minimum information on what he is looking for. In order to ease this form of query, we provide search characteristics that allow the user to discover the contents of any attribute of the publications. For example, the user can specify only one attribute and the system will provide the values used for the attribute as well as the associated references.

The equation of a query is *(attribute1 [attribute2 [,attribute3]]; {constraints})* where *attribute1*, *attribute2*, *attribute3* are any of the attributes of a publication and *constraints* are conditions to be satisfied by the publications. For example *attribute1=author, attribute2=author, Author=John* provides the list of co-authors and their number of publications in which one of the authors is John. The attributes and the selector of constraints are the same (author, keyword, year, editor...).

2.3 The Results of a Query

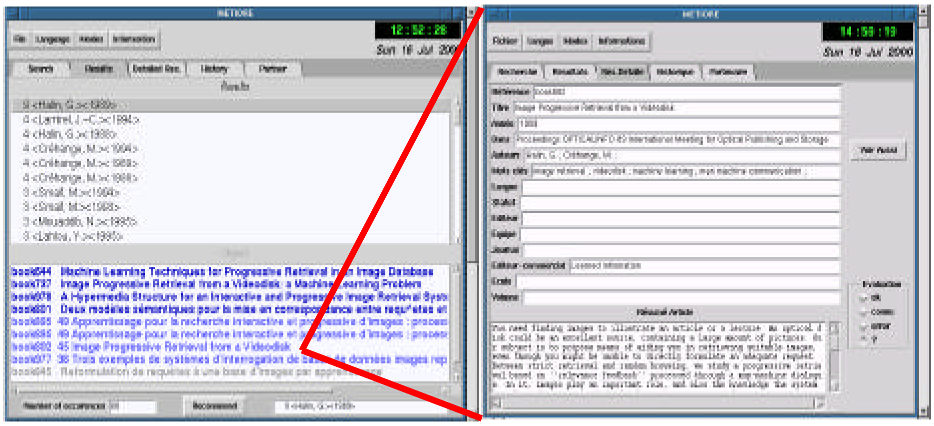

Fig. 1. The two result windows: a) General result with menus in English (left) b) Detailed result with menus in French (right)

The result of a query is displayed in a general result window (Fig. 1.a). It is composed of a list of clusters with their frequencies. For our example (*attribute1=author, attribute2=year; keywords=retrieval)* the result, as commented before, has the frequency and the instance of the two attributes (see Fig. 1.a). For example “9 <Halin G><1989>” means that Halin G had 9 publications in 1989. When the user selects one of the clusters, its components are shown in the down side of the window. The list is sorted using the personalization algorithm that will be explained with the user model. After going through the titles of the solutions, the user can select one of them to see it in detail as shown in the window of Fig. 1.b It is in this window where the user can give his feedback that will be used to update the user model. In the next subsections we will explain the meaning of the feedback and the importance of the color codes used for the list of solutions.

Feedback. The user's feedback is based on the various reasons for accepting or refusing a solution. This is different from the classical types of feedback restricted to interesting/not interesting as in [18] or like the proposal in [2] where the possible evaluations are interesting/not interesting or indifferent. In these classical types of feedback the reasons for the user's decisions are not known. The following four types of feedback are implemented presently in our applications:

The solution is relevant (ok). This means that the solution is valid for the user.

The solution is not relevant because the user knows it already (known). This means that the solution is interesting for this objective but already known.

No opinion on the relevance of the solution (?). This evaluation applies to the case where the user is unable to give another judgment on the solution.

The solution is not relevant because it doesn't correspond to the user's objective (wrong). The user judges the solution as irrelevant to his objective.

Also we have:

Unvisited solution (normal). This is the default evaluation for all proposed solution to signify unvisited.

Color Codes. Brusilovsky [5] justified the utilization of link annotation (equivalent to our color codes) through experiments done with their system InterBook. This system is also used in the lisp tutorial ELM-ART [6; 21]. The use of this kind of annotation makes the user feel that there is some extra help in his interactions with the system, even if a “cognitive overhead may distract users from [5]. In our case we also use this kind of annotation for all the documents proposed as solution to a query. We try to give additional and visual help to the user so that he can know the documents that he had already evaluated and how he had evaluated them in relation to the present objective. Document that have never been evaluated by the user are also presented in color code with the degree of relevance calculated by the system.

2.4 The history of the user’s activities

The user can exploit his past history. The concept of history is also presented in [11] but in our model we sort the history by the user’s past objectives and evaluations. The exploitation of the user’s history constitutes for us another category of activity. Through this category, the user can review his past objectives, the associated solutions and his evaluations associated with each solution. This category of activity may be used for several reasons: a) Recollect past solutions b) Modify the evaluations of past solutions, c) Look for already presented objects in order to find related ones c) Analyze past solutions for some past objectives and the current one.

3 The user model

If we want a system to give personalized responses, it’s necessary to have a model that can represent explicitly and individually each user. One of the most difficult tasks is therefore to identify the most important parameters for representing a user and to what extent personalization is envisaged. We present in the next two sections the components of our model and the algorithm that we use to exploit it for personalization.

3.1 The Parameters of the User Model

In the user model of METIORE we have some general information about the user, such as his personal data (name, login, language of the interface, default mode, etc...). To obtain the real user preference for documents in the model we keep for each objective all the documents evaluated with their evaluation. Each document has some features that represent it (keywords, author, year, etc…). They inherit the evaluation of the document that contains them. This information will be very important for the personalization algorithm.

3.2 Algorithm For Personalization

The problem that we want to solve is a problem of classification. When the user makes a query, we want to sort the solution in the order of his preferences, giving the most relevant solution at the beginning. To do that for each document, that is the result of the query, the system will look for the class of evaluation that is more suited to this document. To make this classification, each document must be parameterized. The selection of parameters for the documents is also an important task. Many systems use only keywords but other parameters can be as important to take into account, like the year, the author or even the journal of publication. Lantz [13] has also analyzed this problem in information filtering. In our system, the number of parameters to use can be configured.

In our user model, we keep the documents evaluated (separately for each objective). Also after each evaluation of the user, for each parameter that belongs to the document evaluated, we increment the value of this evaluation by one. In Fig. 2 is shown the way the parameters are represented. For example, an objective could have the attribute *keyword* with the value *UM* and the evaluations *5 3 0 5* that will mean the number of times that some document that contains this keyword has been evaluated as *ok*, *known*, *?*, *wrong* respectively. Using those values, we calculate the *relevance degree* of the document using the algorithm that we explain below.

```
{objective1 {attribute1 {value ev1 ev2 ev3 ev4} {value ...}} ...
            {attributen {...}}} ...
{objectivem {attribute1 {value ev1 ev2 ev3 ev4} {value ...}}...
            {attributen {...}}}
```

Fig. 2. User evaluation history

Many studies have been done to compare the different methods of classification: Neural Network, ID3 and Bayesian. Most of them have a high complexity. However, naive Bayes, even with its simplicity, due to the assumption of independence between parameters, is able to give similar results and sometimes better than the others with a much simpler calculation. Those comparisons are done in [10; 12; 15; 19], and others versus Winnow and Rocchio algorithms can be found in [20].

Our algorithm is based on the Naive Bayes theory using the precedent evaluations of the user. As stated earlier, the result of the algorithm will be the **degree of relevance of an object to the present objective for a user**. In other words, we try to calculate the probability of the user giving a particular type of evaluation for a given object in the context of the present objective. With this objective in mind we use an adaptation of Naive Bayesian Algorithm [13] to our specific objective. The original formula is shown in (1)

$$P(C/V_{1,J_1},\ldots,V_{n,J_n}) = P(C)\prod_{i=1}^{n} Qi(C,J_i) \tag{1}$$

Where

$$Q_i(C,J_i) = \frac{P(V_{i,J_i}|C)}{P(V_{i,J_i})} \tag{2}$$

In (1) and (2), C is one of the possible classes (ok, known, ?, wrong). V_{iJi} is a Boolean variable with value 1 if the current instance has value J_i and P(C) is the probability of the class C. For example, if the keyword retrieval (J_i) appears in the document the algorithm calculates Q(ok, retrieval). Equation (1) is only valid if the possible attributes are independent, as we assume in our case.

The modification of equation (1) that we propose gives similar results but is less restrictive because we use the average of the weights of each attribute as shown in (3). This reduces the impact of probability 0 for new attributes. The results that we obtain will be already normalized (0-1) with the characteristic that equation (4) is true and we will use this value as the percentage of relevance of the results.

$$P'(C/V_{1,J_1},\ldots,V_{n,J_n}) = P(C)\frac{\sum_{i=1}^{n} Qi(C,Ji)}{n} \tag{3}$$

$$\sum_i P'(C_i/V_{1,J_1},\ldots,V_{n,J_n}) = 1 \tag{4}$$

Having those values, we can now show to the user, the list of articles sorted in the next steps.

1 We show the classes in the order of ok, known, ?, wrong
2 Inside each class, we put the documents already evaluated on top of the list
3 Then we put the documents that the algorithm has attached to this class, sorted by the values that they have obtained in (3).

4 Experimentation of METIORE

In this section we will explain our methods of evaluation and the first results that we obtained.

4.1 The Methods

One way of evaluating a system for personalized response is using the data that the system compiles after each of the user's interaction. This means that if the system proposes a solution, we can use as a measure, the percentage of success of this proposal. This kind of evaluation was employed in the personal assistant of Tom Mitchell [14]. We use this method to calculate the accuracy of the results from the point of view of the system. We present some of our hypothesis below. We use a questionnaire and also the information of the system to validate the answers of the users. The hypotheses are:

- The integration of the user's activities and its association with the objective for the calculation of solutions should give the solutions that the user will evaluate as more relevant.
- The exploration of the active history by the user should facilitate information retrieval and provide more relevant solutions
- When the user hasn't a history, the system can give help by the integration of the history of other users.
- The exploitation of the history of other users should also accelerate the access to relevant solutions
- The possibility of the user to give a feedback on the solutions should give a better understanding of his information need.
- The classification of the solutions by kind of evaluation and the association of color codes should facilitate the retrieval of solutions
- The possibility of a cooperative retrieval should help to accelerate the finding of relevant solutions.

4.2 The First Results

Low percentage of success with the system's recommendation does not necessarily signify a failure as indicated by Armstrong [1]: "Even a modest reduction in the number of hyperlinks considered at each page leads to an exponential improvement in the overall search". This sentence can be applied to our system in the sense that even if only a few of the recommendations are really interesting, that is much better than no recommendation at all. But with our experiments we realized that the results were much better than what we expected, as we will see now.

Our experiments have been done at the computer research laboratory LORIA in France. The database that we used was the database of the publications of this laboratory. The people that were involved in the experiment were 20 Ph. D. students and research members of the laboratory. The experimentation was carried out over three months, divided into sessions of about one hour. The experimentation was also carried out in three phases: the phase of explanation of the functional characteristics of the prototype; the phase of independent use where users selected their own objectives and the phase of feedback with questionnaire.

To be able to generate statistics of the predictions of the system after each evaluation of the user, some information is kept by the system. We save the prediction that the system made for each document and the evaluation that the user made. It should be noted that at the beginning, the system hasn't enough information to recommend a document to the user and the system is not able to know if a given document would interest the user. This is when the prediction is considered as **normal**. Even in those cases the documents are relevant in a good number of cases.

predicted\real	ok	known	?	wrong
pok	50,00%	13,33%	15,00%	21,67%
pknown	75,00%	25,00%	0,00%	0,00%
normal	38,89%	38,89%	0,00%	22,22%
p?	33,33%	33,33%	33,33%	0,00%
pwrong	33,33%	0,00%	23,33%	43,33%

	% of evaluations
ok	47%
known	21%
?	13%
wrong	19%

Fig. 3. Summary for all users interactions: a) Predictions vs. Real evaluations b) frequency of evaluations

The most interesting result of the evaluation is the final solution for all users. Fig. 3 presents a tabular representation for all the predictions made by the system and what the users really evaluated. From that we can conclude the following:

The percentage of (ok/pok) is very interesting. 50% of the times that the system proposed a solution as correct, it was really correct for this user. 13,33% of the times that the system predicted that it will interest him and it is true but the user already knew that. That means that the 63,33% of the prediction of the system is correct. We consider this a good result.

The prediction of *known* has a high evaluation as ok 75%. This can be considered reasonable because when a user evaluates a document as known it means that the document interests, and similar documents can also be interesting but perhaps are unknown.

We have also observed during the experimentation that the number of successes increased over time. As we had suspected, the more the user uses the system, thus training the system on his preferences, the better the proposals he receives from the system.

Another observation that could also explain the high percentage of good evaluated solutions is that most of the users accepted the recommendation of the system, and rarely they evaluate the suggested ones as wrong. This action can be a little dangerous at the beginning when the system doesn't have a lot of information about the user and the recommendations may not be accurate. That's why we have thought to put a threshold, such as a minimum number of evaluated solutions before which the system won't give any advice. This could be also configured.

The development of the questionnaire has also passed by a process of refinement through some initial tests. From the observation of the results of the questionnaires and the observation of the experiments we have also observed the following:

Nearly all the users were satisfied with the solutions that the system gave them and they felt that the system had learnt their objective. In particular, some users at the end of the evaluation asked for automatic recommendation without making any query

We thought at the beginning that giving the feedback on all proposed documents would be a task that the users wouldn't like. Many of them had commented that giving feedback to the system is the only way to get the most correct assistance. So they didn't hesitate to do it.

Mostly the regular users have used the active history and they found it an interesting tool. But the first time users (or users of only a few sessions) couldn't appreciate its utility.

5 Conclusions

We have presented in this paper how personalization is carried out through the use of a user model. We also presented the algorithm used for this personalization. The prototype METIORE, in which our proposals have been implemented, was briefly presented and our observations during the experimentation to evaluate the performance of the system were summarized.

We noticed that the first phase of the experimentation in which we explained the functional characteristics of the prototype was decisive. We thought that we would need thirty minutes to explain them but an average of ten to fifteen minutes was sufficient. Unfortunately, this observation cannot be generalized because we had users that are already familiar with the user of graphic user interface. We therefore intend to continue our experimentation in a university library where many users are not familiar with the use of graphic user interface.

We have observed that the algorithm for personalization was efficient and simple to implement. We have tested it on a database with about 5000 bibliographic references. We intend to test the system on a more important database and to compare it with other algorithms as the pure Naïve Bayes or random recommendations.

The objective provided by the user has not been fully exploited. We intend to apply natural language processing techniques on them in order to calculate the degree of similarity between two objectives. This approach can be used by the system for first time or casual users. We also intend to provide an interface to navigate through the objectives for long-term users that may have a long list of past objectives. The objectives are presently listed simply in the order in which they are given. Finally, we are making the system available on the network of our laboratory for the researchers to use for information access.

6 References

1. Armstrong, R., Freitag, D., Joachims, T., & Mitchell, T. (1995). "Webwatcher: A learning apprentice for the world wide web". AAAI Spring Symposium on Information Gathering from Heterogeneous Dis tributed Environments
2. Benaki, E., Karkaletsis, V. A., & Spyropoulos, C. D. (1997). "Integrating User Modeling Into Information Extraction: The UMIE Prototype". UM'97 . URL= http://um.org.
3. Billsus, D., & Pazzani, M. (1999). "A Hybrid User Model for News Story Classification". Proceedings of the Seventh International Conference on User Modeling (UM '99) Banff, Canada
4. Billsus, D., & Pazzani, M. (1999). "A Personal News Agent that Talks, Learns and Explains". Proceedings of the Third International Conference on Autonomous Agents (Agents '99) . URL= http://www.ics.uci.edu/~pazzani/Publications/agents99-news.pdf.
5. Brusilovsky, P., & Eklund, J. (1998). "A Study of User Model Based Link Annotation in Educational Hypermedia". Journal of Universal Computer Science, 4(4), 429-448
6. Brusilovsky, P., Schwarz, E., & Weber, G. (1996). "ELM -ART: An intelligent tutoring system on World Wide Web". Intelligent Tutoring Systems (pp. 261-269). BerlinIn Frasson, C., Gauthier, G., & Lesgold, A. (Eds.) Springer Verlag
7. Brusilovsky, P. (1996). "Methos and thecniques of adaptative hypermedia". UMUAI, 6(2-3), 87-129
8. Chin, D. N. (1989). "KNOME: Modeling What the User Knows in UC". User Models in Dialog Systems,
9. Kamba T. , & Bharat K. (1996). "An Interactive, Personalized Newspaper on WWW.". Multimedia Computing and Networking California
10. Keogh, E., & Pazzani, M. (1999). "Learning augmented Bayesian classifiers: A comparison of distribution-based and classification-based approaches.". Uncertainty 99, 7th. Int'l Workshop on AI and Statistics, (pp. 225-230). Ft. Lauderdale, Florida
11. Kobsa, A., Nill, A., & Dietmar Müller. (1996). "KN-AHN: An Adaptative Hypertext Client of the User Modeling System BGP-MS". Review of Information Science 1(1). URL= http://www.inf-wiss.uni-konstanz/RIS.
12. Kononenko, I. (1990). "Comparison of Inductive and Naive Bayesian Learning Approaches to Automatic Knowledge Acquisition". Current Trends in Knowledge Adquisition, 190-197
13. Lantz, A., & Kilander F. (1995). "Intelligent Filtering; Based on Keywords Only? ". Computer Human Interaction (CHI'95)
14. Mitchell, T., Caruana, R., McDermott, J., & Zabowski D. (1994). " Experience With a Learning Personal Assistant". Communications of the ACM, 37(7)
15. Mitchell, T. M. (1997). "Machine Learning". The McGraw-Hill Companies, Inc.
16. Rich E. (1979). "User Modeling via Stereotypes". International Journal of Cognitive Science, 3, 329-354
17. Rich, E. (1983). "Users are individuals: individualizing user models". Int. J. Man-Machine Studies, 18, 199-214
18. Schwab I. , & Pohl W. (1999). "Learning Information Interest from Positive Examples". User Modeling (UM'99)
19. Singh, M., & Provan, G. M. (1996). "Efficient learning of selective Bayesian network classifiers". . Proceedings of the 13th International Conference on Machine Learning
20. Versteegen, L. (2000). "The Simple Bayesian Classifier as a Classification Algorithm". URL= http://www.cs.kun.nl/nsccs/artikelen/leonv.ps.Z.
21. Weber, G., & Specht, M. (1997). "User modeling and adaptive navigation support in WWW-based tutoring systems". (pp. 289-300). Wien Springer-Verlag