

Conference Proceedings

1991 IEEE International Conference on Systems, Man, and Cybernetics

"Decision Aiding for Complex Systems"

October 13-16, 1991

***Omni Charlottesville Hotel
and the
University of Virginia
Charlottesville, Virginia***



IEEE

Volume 2

91CH3067 - 6

A Neural Network Approach for User Modeling

Qiyang Chen and A. F. Norcio
Department of Information Systems
University of Maryland
Baltimore, MD 21228-5398

Abstract—A neural network approach to user modeling is proposed in the context of information retrieval. Users' level of expertise and their inquiry interests underlying the goals are considered as two major factors affecting information provision. A prototype system, UM-net is presented for modeling user's domain experiences and the inquiry interests and tailoring the descriptions about software packages provided to a user. The paper concludes with some thoughts on further research.

I. INTRODUCTION

It is necessary that an interactive system has certain perspectives about its user community. Traditionally, they are oriented toward generic user characteristics, and encoded procedurally throughout several components of the system. However, three important concepts that characterize current user modeling research have been emphasized [3, 5-7, 9]:

1) Individualization: A user model should contain characteristics that are not only common to a class of users but also unique to each specific user.

2) Knowledge sources: Generally, a user model should take into account following sources of knowledge, which are related to the tasks a user is performing:

- . user's general knowledge or task background.
- . user's beliefs, goals and plans .
- . user's cognitive limitation and preference.

3) Knowledge representation: Knowledge about a user is represented declaratively in a separate module for multiple uses, rather than distributed procedurally throughout the system. The modeling system should provide the procedures for data abstraction that supports the representation.

Incorporating user modeling into a interface has many advantages that vary among applications. Two aspects of these advantages are as follows:

1) Cooperative and intelligent interaction: User modeling enables system to tailor its response individually to user. The dialogue concerns the specific

features of each individual user with respect to tasks. Thus, it exhibits a more cooperative mode. Modeling a user's background, beliefs and goals enables a system to provide some additional information or avoid redundancy and incomprehensibility in its response. In addition, it may allow the system to detect and correct a user's misconception in the interaction [8]. This underlies an intelligent dialogue behavior, which sometimes seems not necessarily "friendly".

2) System efficiency and effectiveness: Since a user model may have extensive default knowledge or assumptions about a user and the task, the interaction may be better focused. In addition, the decision made by system can also be enhanced in terms of modeling process that guides the system to provide most appropriate information regarding user's specific characteristics, goals, his knowledge deficiencies and mistaken tendencies [6].

II. AN ARTIFICIAL NEURAL NETWORK APPROACH TO USER MODELING

In this paper, we propose an artificial neural network (ANN) approach to user modeling in information retrieval system that tailors the appropriate information provided to a user. In our approach, the collection of a user's information is conceptually regarded as an individual "user image ". We suggest that ANN is an appropriate alternative to process this image for its strong capability of pattern classification with fault tolerance, graceful degradation and signal enhancement [10]

In addition, with its strong capability of learning, an ANN can deduce the relationships between the input and output in the absence of knowledge. In our approach, the historical information collected in the "log" of the sessions can be used as training data. The trained network may be retrained if necessary.

We suggest that this approach provides a technique to overcome some of the inherent shortcomings in traditional rule based methods for dealing with incomplete information and experience-based learning.

We have been implementing this approach in a prototype system: UM-net, a system that generates descriptions of DSS tools. In following sections we will briefly introduce the design of UM-net. After discussing each system component and the problems intended to be solved, we conclude with a discussion on some further research directions.

A. Application Domain

We focus on two factors affecting a response to a user in the context of information seeking process: user's level of expertise and their interests in an inquiry. Since the information stored about a subject/object may be very large and organized in many ways of abstraction, it is necessary to select a subset of data that may be most appropriate to satisfy user's request. Among all the characteristics, user's expertise in relevant field and interests underlying an inquiry are most fundamental factors for determining which kind of and how much of information that should be provided.

The application here is a software information retrieval system developed in our department, which stores various information about DSS tools and other software packages. Descriptions in software information data base are organized into frames that form a generalization hierarchy. Each software package is described in terms of its functionality, applications, operation properties and commercial information. Each frame in this hierarchy contributes the information it contains to its super frame. We suggest that user's level of expertise mainly affects the amount or the degree of information specialization; and a user's interests or goals mainly affect the type of information. Thus, the determination of the type of information involves the some kind of "width-search" and the determination of the degree of specialization implies "depth- search" in the generalization hierarchy. That is, tailoring a description becomes a procedure of two-dimensional search which, in our approach, is guided by neural network output.

B. System Architecture

The architecture of UM-net is shown in figure 1. The goal of UM-net is to model a user's level of expertise and interests to support tailoring the description provided to a user. It mainly consists of three modules:

- . Dialogue Manager (DM) that acts as a front-end of system, monitors sessions and coordinates the other system components.

- . Neural Network (NN) module, a set of feedforward networks trained by back propagation algorithm [10]. recognize the user's level of expertise and their intention to determine the amount and types of information provided.

- . Description generator (DG) that retrieves the software data base, according to the output of NN, and generates the descriptions for an inquiry.

UM-net works on three data bases:

- . User Profile(UP) recording most recent information about a user who has accessed the system, including all inputs and outputs of NN.

- . Session Log (SL) containing all historical information about sessions held by users, including the identifications of frames that contain information provided to users.

- . Training Set (TS), a subset of SL which is sampled by DM.

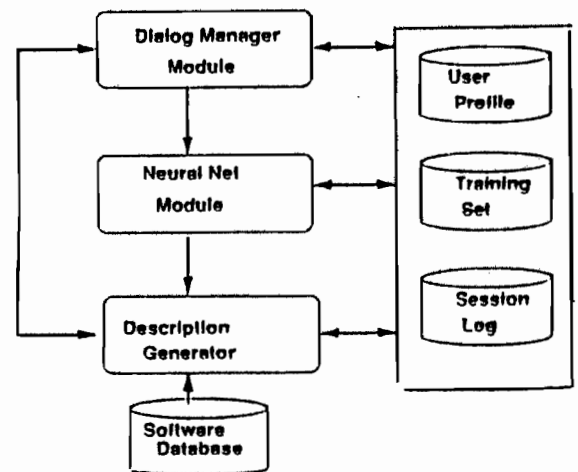


Figure 1: The Architecture of UM-net

C. Neural Network Processing

The neural network module (NN) is used to process "user image" which is characterized by two aspects: user domain knowledge and user's interests in information inquiry. NN is mainly composed of three sub-modules, each perform different task of pattern classification. Figure 2 shows the schematic diagram of NN.

1) Expertise analysis module(EA), a three layer network, consists of 22 input units, four hidden and four output units. Its function is to analyze the pattern of user's domain knowledge and to specify the implied level of expertise in several relevant fields. The original data for input is about user's engagement in following categories:

- . Programming, database application and system operation: total time (in typical work months) of engagement in above activities.

- . Application of decision models: total time of using modeling techniques in problem solving. The techniques are popular procedures summarized in [4].

- . Application of DSS software package, the total time of using the packages or others.

- . Position description that is most relevant to DSS applications, the total of time of engagement.

The set of activations of EA output (denoted by E_o) indicates the user's level of expertise in four aspects: computer literacy, DSS application, management and capability of problem solving in general with respect to decision making.

This characterization of input and output is based on our investigation and analysis on the casual relationship between users' domain experiences and their level of expertise. In the context of DSS application, we restrict the network processing to the user's characteristics that most likely reflect user's domain expertise in above four aspects that most likely affect the specialization of information provided to a user.

2) Interests analysis module (IA), a two layer network, consists of 20 input and four output units. It is used to analyze the important entries of users' inquiries and determine the user's underlying intentions.

The input of IA, which comes from a user's inquiry, contains four sets of input values:

- . model-oriented inquiry: built-in functions and their related procedures,

- . application-oriented inquiry: application and examples;

- . operation-oriented inquiry: features of manipulation language and installation requirements, and

- . commercial-oriented inquiry: information about the versions, prices, vendors, etc.

The set of activations of IA output (denoted by I_o) indicates a user's different kinds of implied intention, namely, to understand

- . technical features about system functionalities,

- . application domains and relevant instances,

- . operational features about a software, and

- . purchase information.

It should be noted that the complete information about one of user's intentions may not be determined exclusively from a single corresponding input set. The entries of inquiries in different input sets may be correlated. The distribution of user's intentions should be reasoned from the pattern of the inquiries.

3) Description analysis module (DA). It is a five-layer network that includes 12 input, 4 output units, and three hidden layers that each consists of four processing elements. The inputs are from three sources: outputs of both EA and IA, and a subset of input in IA (denoted by I_I):

$$I_I = \{ x_i \mid x_i = \max \{ x_j \mid x_j \in C_i \}, i = 1, 2, 3, 4 \} \quad (1)$$

where C_i is a set of input values corresponding to one of input categories in IA. This set indicates the user's most imperative request within each inquiry category.

The three hidden layers are introduced to simulate the sequences of inference performed by human intermediary in information advising activities [2]. The input of first layer is the output of EA. Its function is to integrate a user's expertise through several domain and properly weight the influences among domain expertise. The second hidden layer connects both previous layer and partial input of IA (I_I). Conceptually, this layer establishes the relation between user's background and their inquiries. The third layer receives the activation of second layer and the output of IA which indicates the potential goals underlying a user's inquiry. This layer, together with the output layer produces the final result of the neural network processing. The output of NN (denoted by V_D), called description index, is an indication on how wide and deep a description should be with respect to software's functionality, application features,

operational and commercial properties, which are represented by four units. V_D is used to guide the description generation. The distribution of output values in V_D reflects the user's different intention regarding software information, and the magnitude of an output value indicates how specific a description should be.

This three module neural network structure provides a way to model different types of information about user explicitly, and then to synthesize the results and produce guidance for information provision. Therefore it serves not only for user modeling but also information retrieval. Generally, subdividing a neural network processing system (NN) into modules may bring following advantages:

- 1) Multiple uses, each module can be utilized separately for different purposes,
- 2) Flexible training, the training process could be conducted for either single module or the whole system, which gives more flexibilities to system adaptation, and
- 3) More clear insight on knowledge representation and inference in neural network system which may enhance the capability of explanation.

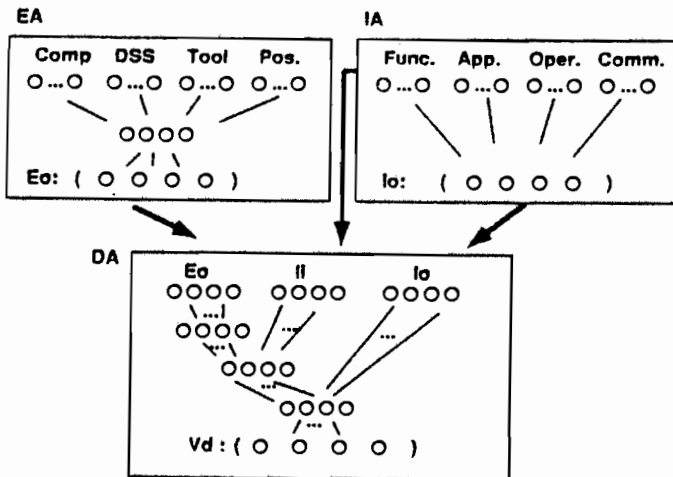


Figure 2 : The schematic diagram of NN

D. Description Generation

A database retrieval module, Description Generator (DG), is used to search and tailor the descriptions stored in software information database. The inputs of DG are the description index (V_D) and the entries of user's original inquiry. The strategy for generating a description can be illustrated as follows:

. Locating the frames regarding user's original inquiry. A user's inquiry may not directly related to certain frames in the generalization hierarchy in software database, but it falls into one or more of the four categories (i.e. software functionality, applications, operational features and commercial information). DG analyzes into which category a user's inquiry falls and locates the a general frame related to that category. This general frame is at the top level of a generalization hierarchy. The location of frame is a rule based process.

. For each located frame, calculating the depth of search:

$$h_i = \text{int}(x_i * d_i), x_i \in V_D, 0 < x_i < 1 \quad (2)$$

where x_i is one of network output values corresponding the category this frame belongs to; d_i is a constant that indicates the depth of a local generalization hierarchy; h_i is an integer indicating the maximum depth of the search within the subtree rooted by this frame.

. For each subtree rooted by located frame, searching its subnodes/frames with the restriction that the maximum searching distance between the currently searched subnodes/frame and the root frame is no larger than h_i . Then organizing the searched frames into a temporary file submitted to Dialogue Manager (DM) for display, and inserting the frame identifications into session log database (SL).

E. Dialogue Management

Dialogue Manager (DM) plays some major roles as follows:

- 1) Identification and transformation of user inputs: DM receives users' inputs with respect to their domain experiences and information inquiries, and translates these inputs into numerical representation. If a user is known to the system, DM retrieves the user profile database (UP) for obtaining the information about the user's background and the level of expertise that was the output of NN. Otherwise, for the first-time users, they are asked a set of questions

regarding the domain experiences in several aspects. DM allows user to ignore the preliminary interview to enter their inquiry directly. In this situation, a set of default values on user's level of expertise are established. The strategies of setting default value are based on the assumption that for each aspect of a software package, the more detailed the inquiry is, the more experiences a user might have in relevant domain. A user may also only enter the most representative information about his background in each category in terms of his own judgement, rather than answer all questions. For example, if a user has already indicated experiences in Ada, then the system does not need to know his experiences in Basic for determining the level of expertise in programming. Since the neural network processing is robust in tolerating incomplete input, it can produce correct classification without complete input as long as the input captures the key information. This property could facilitate default reasoning in which the stereotype based production systems are usually used [1, 5, 9].

2) Information display: DM displays descriptions stored in the frames that have been retrieved by DG for current inquiry. The sequence of display is in the order that the frames corresponding to higher index value in V_D and appearing at higher level of hierarchy are displayed first.

3) Maintenance of databases: DM maintains three databases: SL, TS and UP (see figure 1). It provides the interface to access and update these databases. In run time, it updates UP according to the changes of user background and NN output, inserts the inputs/outputs of NN and the original entries of user inquiries into SL. The system performance is monitored in terms of the accuracy of targeting (see next section). When it shows that the networks need to be retrained, DM creates a new training sample databases

F. Training Strategies

Most of the data for initially training NN comes from subjective analysis, which may not correctly reflect the real world situation. However, it is not critical that the accurate data describing the user characteristics and inquiry interests be available at beginning because of the capability of learning in neural network. In this approach, a trained network

can be retrained when systematic observation shows that the output (V_D) does not serve description generation well. The data collected for retraining NN are from following sources.

1) Recalculated description index (V_D') During each session, a user can control the searches within the subtrees defined by V_D and located frames in terms of either stopping display description or conducting further search beyond the scope of that subtree. When user stops display, system records current depth of search (denoted as h_i'), and assumes that h_i' is the appropriate depth for current search. By using equation (2) the index value x_i is recalculated. Thus forms a new description index (V_D'), which is regarded as the appropriate indication for the description generation.

2) Updated EA outputs (E_o') If V_D' is generated, it implies that E_o may not be correct. Then system asks the modification from user on E_o for which a reference between linguistic variables and numerical representation is provided to user. The updated output from EA (E_o') can be used as training data.

3) Updated IA outputs (I_o') The process of generating I_o' is similar to that of E_o' .

These three types of data, together with user's original input are stored in SL database and are selected by DM to form a part of training set. Another part of training data comes from the previous training data set which are unchanged and consistent with these new training data. Two factors determine when to trigger a retraining process:

. Time period. After a period of running, the network may need to be retrained due to some changes such as new patterns of user background or inquiry interests emerge.

. Accuracy of targeting. If the accuracy rate of targeting a set of frames/descriptions indicated by V_D is so low that too much V_D' occur, then networks should be retained. DM dynamically provides the statistical data concerning these two factors.

III. CONCLUSION

In this paper, we proposed an ANN approach to user modeling. In particular, we emphasized that user's level of expertise and inquiry interests underlying the goals are two major factor affecting information provision. These two factors can be better recognized by using ANN technique for its capability

of pattern classification and learning. We believe that this approach can be extended to other information retrieval environments. However, some relevant issues need to be further explored.

1) Although we have examined two major factors affecting information provided to user, some other individual characteristics about a user still play important roles in information seeking process, such as a users' beliefs regarding system functions and types of information provided, user's cognitive limitation, and preference for the ways of description presentations, etc.. In this study, the individualization is primarily realized in the numerical differences between network outputs, which might not be adequate to model all necessary information about a user.

2) The changes of attributes in a user model is another problem in both our approach and other stereotype based user modeling systems. The attributes assigned to a user model may not be the most representative for the current user community. On the other hand, during run time some new attributes may be found more useful and need to be involved in neural network processing. It may cause the changes of network architecture and new training process is also required. Thus increases the complexity of system implementation and affects system efficiency.

3) It is necessary to evaluate dynamically the effect of user modeling on the overall performance of the system. Neural network based learning process should be contingent on this evaluation. Ideally, the statistical methods using historical data of interactions should be built-in to support the dynamic evaluation so that the evaluation could be more accurate and objective.

4) It is acknowledged that ANN cannot accomplish all tasks of user modeling in the sense that the input/output data need to be transferred; some default reasonings are still necessary; and the results need to be explained. These processes still require a symbolic reasoning mechanism (e.g. production system) as an important part in user modeling. Although a neural network is powerful for incomplete information processing, how to integrate the numerical and symbolic processing together to handle other types of uncertainties, which may also occur in user modeling, effectively is still an open question.

1. Brajnik, G., Guida, G., and Tasso, C., "User modeling in expert man-machine interfaces: A case study in intelligent information retrieval," *IEEE Transactions on Systems, Man, and Cybernetics* 20(1), pp. 166-185 (1990).
2. Brooks, H. M., Daniels, P. J., and Belkin, N. J., "Problem description and user models: Developing an intelligent interface for document retrieval systems," *Proceedings of Advances in Intelligent retrieval Informatics*, pp. 191-214 (1985).
3. Daniels, P. J., "Cognitive models in information retrieval - an evaluative review," *Journal of Documentation* 42, pp. 272-304 (1986).
4. David, M. W., *Applied Decision Support*, Prentice-Hall, Englewood Cliffs, NJ (1988).
5. Finin, T. W., "GUMS - A general user modeling shell," in *User Models in Dialogue Systems*, ed. A. Kobsa and W. Wahlster, Springer-Verlag, Berlin (1989).
6. Norcio, A. F., "The software engineering of adaptive human-computer interfaces," *Proceedings of the 1989 IEEE International Conference on Systems, Man, and Cybernetics* (1989).
7. Norcio, A. F. and Stanley, J., "Adaptive human-computer interfaces: A literature survey and perspective," *IEEE Transactions on Systems, Man, and Cybernetics* 19(2) (1989).
8. Quilici, A., "Detecting and responding to plan-oriented misconceptions," in *User Models in Dialogue Systems*, ed. A. Kobsa and W. Wahlster, Springer-Verlag, Berlin (1989).
9. Rich, E., "Users are individuals: Individualizing user models," *International Journal of Man-Machine Studies* 18(3), pp. 199-214 (1983).
10. Rumelhart, D. E. and McClelland, J. L., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations*, The MIT Press, Cambridge, MA (1986).

REFERENCES