

MENU INTERFACES TO EXPERT SYSTEMS:
OVERVIEW AND EVALUATION.

Galina Datskovsky

CUCS-168-85

Menu Interfaces to Expert Systems : Overview and Evaluation.

By Galina Datskovsky
Columbia university, New York.

In order for a human expert to be able to answer a person's question s/he often has to carry out extensive dialogs with that person in order to gather information about that person's needs. Extensive interaction and clarification are also needed for expert systems. One way expert systems communicate with their users is via a menu interface. To gather information the system poses a question and the user is given a number of possible answers and s/he then proceeds to choose the one that s/he thinks best corresponds with the correct answer. For example, in MYCIN the choice is very often between a yes and a no, as shown below:

```
system: Is PT248 a compromised host (e.g. alcoholic..)
user:   yes (or no)
```

A menu system is very easy to use. It certainly insures that a system can handle everything a person can input. There is a finite number of predetermined answers to each question, and for each of these answers the system knows exactly what to do. There are, however several problems with such menu interfaces. A person is very limited in choice of input. If none of the choices provided by the system are adequate, the user can not just give an arbitrary answer, however more satisfactory it may be. Moreover, since a menu in effect spans out a tree with many paths, a set of multiple choices a user sees at any given point depends on answers to previous questions. Therefore, if none of the choices presented to users satisfy their needs, they may end up answering the wrong set of questions and going down the wrong path and may find it difficult to back up to the point where the wrong choice was made. Sometimes, in order to answer a question the user may need some extra information from the system, but there is no facility for him to ask for that information at an arbitrary point. Although the system may be able to provide such information, in order to acquire it the user would have to choose a totally different sequence of multiple choice answers. These are just some of the problems of using menu interfaces to expert systems.

We feel that it is important for the user to have more flexibility and control in order to arrive at the answers to his/her queries in the most informative and least time consuming way. We are working on building a natural language front end for expert systems which will be as domain independent as possible. However, before starting work on such a project it is certainly important to analyze existing expert systems in order to better understand what the problems with such systems really are, and whether the addition of a natural language interface will solve any of these problems.

In this report we analyze some transcripts of interaction with MYCIN. MYCIN is a backward chaining expert system; it bases all its questions on the medical knowledge that it has. Here is an outline of the way the system works. The system first collects data from the physician, intelligently asking the relevant questions and then prescribes treatment based on its analysis of the physician's data. An example of a session between MYCIN and a physician is shown below, with our own commentary on the interaction in italics:

A 52 year old male with a history of salmonella sepsis earlier this year, now readmitted for elective resection of aortic aneurysm; ruptured septic aneurysm found at surgery and 3 of 3 blood cultures grew gramneg rods out the same day.

```
-----PATIENT-248-----
1) Patient's name:
** PT248
2) Age:
** 52 YEARS
3) Sex: ** MALE
4) Race:
** CAUCASIAN
5) Are there any cultures for Pt248 which may be related to the present
illness, and from which organisms have been grown successfully in the
microbiology laboratory?
** YES
```

Here the physician has a choice of a yes/no answer. If the answer is yes a certain path is taken and questions are asked about the organism. If the answer is no then a different path is taken at this point.

```
-----CULTURE-1-----
6) From what site was the specimen for CULTURE-1 taken?
** BLOOD
7) Please give the date and time when the blood culture was obtained.
   (mo/da/yr time)
** 21-Oct-75 11:57AM
The first organism isolated from the blood culture will be referred to
```

as: -----ORGANISM-1-----

The way organism identity is established is very tedious. If the physician were asked to enter the information about an organism s/he would probably give the same information without going through all the steps.

- 8) Please enter the laboratory-reported identity of ORGANISM-1, including species or subtype, if known:
 ** UNKNOWN
- 9) The stain (Gram, India ink, or Ziehl-Neelsen acid-fast) of ORGANISM-1:
 ** GRAMNEG
- 10) Is ORGANISM-1 a rod or coccus (etc.):
 ** ROD
- 11) What is the form of the individual organisms (e.g. Lancet-shaped for cocci, fusiform for rods, etc.)?
 ** UNKNOWN
- 12) Did the Gramneg rod from the blood culture (ORGANISM-1) grow aerobically (i.e., in the aerobic bottle or on the aerobic plate)?
 ** YES
- 13) Did the Gramneg rod from the blood culture (ORGANISM-1) grow anaerobically (i.e., in the anaerobic bottle or on the anaerobic plate)?
 ** YES
- 14) Any other organisms isolated from the blood culture?
 ** NO
- 15) Are there any other cultures from which organisms have been successfully isolated in the laboratory?
 ** YES

As illustrated by the example above, at every point a physician has a limited number of answers to a question that he must choose from in order to proceed with the diagnostic session. There is a choice of a few answers to any given question. The questions that determine the actual path of the system are mostly yes/no questions. A different path might have been taken if a different response were given. For example, if the response to question 5) was *no*, as it was in the case of patient 629, the next question would have been:

6). Are there any pending cultures for Pt629?

The dialog between the system and the physician is rather long and tedious. Time and effort could be saved if the physician were able to provide more information. For example, a short case history is provided with most cases. If the program were to parse that information, time would be saved on answering certain questions. Also, as I already mentioned in the above example, a lot of time would be saved if the physician just entered the identity of organisms, without

having to go through all the questions. (This would correspond to giving additional information ahead of time).

The program also has a question answering facility which can be invoked in one of two places. The facility may be used during a consultation session. The physician may at any point ask a HOW or WHY question, like 'How did you decide to treat with drug x' or 'Why did you ask about condition y', and the program will trace its steps through the inference tree. One can also ask questions after the consultation or load a special question answering facility during a consultation. This facility allows one to ask questions about different parts of the consultation session, not just simply HOW's and WHY's. This is a somewhat limited natural language facility. It allows only certain types of questions and the answers are generally just a dump of the appropriate rules together with justification for why the rules fired. An example of a question answering session is shown below:

system: Do you wish to continue with the regular QA module?

user: YES

system: Please ask questions now. When you are finished and want to begin a new consultation or sign off the terminal, enter the word STOP.

Type HELP for the list of question options.

user: Why did you want to know whether the patient is a compromised host?

system: The following rules used:
 whether Pt629 is a compromised host

343 80 53

The parameter most frequently concluded by these rules was:
 whether antimicrobial therapy is not recommended for the item

MYCIN is a very well structured system and one may argue that the need for natural language is not at all apparent in a system like this, yet physicians will not readily use the system. I feel that a few modifications would make the system a great deal better.

1. When the system answers the HOW and WHY questions it simply prints out the rules without much explanation. It would be nice if the rules were rephrased. That is, a brief English explanation of the rule would greatly enhance the system's explanation facility.
2. A lot of information which the program asks for is already given in the

case history of a patient. If the program parsed at least that paragraph a lot of routine questions could have been avoided.

3. There is no way for the physician to volunteer any information in a totally menu driven system. A piece of information which s/he may feel is important for the diagnosis may never be known by the program.

A natural language interface would allow the user to volunteer information. It would also allow the physician to skip the parts s/he thinks are unimportant. It may be able to deduce answers to some of the questions from previous answers. The question answering mode can also be greatly enhanced. Instead of just printing out rule numbers, it would be nice if the system were able to give some sort of explanation. However, the natural language must be somewhat restrictive as well. It should keep in line with the domain in order for the system to be able to handle most user queries, and the system should also guide and restrict the user to some extent. This restrictiveness, however, should be as transparent to the user as possible. Use of such semi restrictive natural language interface would greatly enhance the performance of the system.

Even though MYCIN is a very well constructed system, it may become more usable with the addition of a natural language front end. We plan to do similar analysis of several other expert systems and then use the results to build a fairly domain independent natural language front end for a class of expert systems.

References

- [1] Barr, A. and Feigenbaum, E.A.
The Handbook of Artificial Intelligence Volume II.
William Kaufman, Inc, Los Altos, California, 1982.
- [2] Davis, R.
*Knowledge Acquisition in Rule-Based Systems-Knowledge About
Representation as a Basis for System Construction and Maintenance.*
Academic Press, 1978, .
- [3] Davis, R. and Lenat, D.B.
Knowledge-Based Systems in Artificial Intelligence.
McGraw-Hill, New York, 1983.
- [4] transcripts of interaction with the MYCIN program.