

NATURAL LANGUAGE INTERFACES TO EXPERT SYSTEMS

Galina Datskovsky

CUCS-169-85

Natural language interfaces to Expert systems.

By Galina Datskovsky
Columbia University, New York.

Currently, a very popular way for expert systems to 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.

There are 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, going down the wrong path and may find it difficult to back up to the point where the wrong choice was made. Very often, in order to answer a question the user may need some extra information from the system, but because of the lack of a natural language facility he can not simply ask for it. Expert systems are becoming more and more complex and the limited menu interface is simply inadequate.

To solve this important set of problems, we are working on a natural language interface. This interface will alleviate the problems mentioned. It will allow the user to input additional information and allow the user to ask for information at any point. It will also allow a user to stop pursuing a path at any time and let him pursue a different path. In general it is very 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. Our goal is also to make the natural language system transportable from one expert system to another. Some changes will have to be made when the system is moved from domain to domain: for example, the vocabulary will have to be modified. Our goal is to keep these changes to a minimum.

We have begun our work by building a natural language front end to a tax expert system. This expert system contains a number of agents, some that gather information, some that fill out forms and others that answer questions in various tax domains. As a first step we are building an interface to the agent that determines whether one can claim someone as a dependent. This system is an ideal test bed for several reasons. The existing interface to this system is a menu interface. There is a lot of branching: i.e., depending on the question and the answer, the user will end up going down different paths in the system. An example is given below:

From the previous set of menu questions the system has found out that the user is trying to claim a child as a dependent:

```
system: Is your child:
        1. over 19    2. under 19
user:    2.
```

Now the system knows that one of the tests for eligibility is met and it does not have to ask whether the child is a student or not. If the answer were 1, then next question would have been: Is your child a full time student?

Often a lot of information is carried in the question. For example, "Can I claim my 20 year old son if he is in college?" From the question we can infer that the dependent is a student and a direct descendant of the user, therefore three of the five eligibility tests are met and at least three of the menu questions that would have been asked can be omitted.

Often the user requires some additional information as shown in the example below:

user: How much of my father's support do I have to provide in order to claim him as my dependent?

Using the menu interface, in order to acquire this information the user must go through a set of menu questions, even though many of these questions are irrelevant. However, since the information is immediately available in the system, the natural language interface will be able to provide it right away, without asking additional questions.

In order to develop these facilities, we are addressing the following questions: How much does the system have to learn about a new domain? How does it learn about the underlying expert system structure? How do we map the parsed sentences into system usable form? How do we make the semantics and the dictionary general enough so that the system is easily transportable from one domain to another? These are certainly difficult questions to resolve, but answering them will enhance the fields of expert systems and natural language and will bring us much closer to having intelligent systems.

References

- [1] Barr, A. and Feigenbaum, E.A.
The Handbook of Artificial Intelligence Volume II.
William Kaufman, Inc, Los Altos, California, 1982.
- [2] Clancey, W.
The Epistemology of a Rule-Based Expert System - a Framework for
Explanation.
Artificial Intelligence 20 , 1983.
- [3] Datskovsky, G.
Menu Interfaces to Expert Systems: Evaluation and Overview.
Technical Report, Columbia University, New York, 1984.
- [4] Davis, R.
*Knowledge Acquisition in Rule-Based Systems-Knowledge About
Representation as a Basis for System Construction and Maintenance.*
Academic Press, 1978, .
- [5] Davis, R. and Lenat, D.B.
Knowledge-Based Systems in Artificial Intelligence.
McGraw-Hill, New York, 1983.
- [6] Pollack, M., Herschberg J. and Webber, B.
User Participation in the Reasoning Process of Expert Systems.
AAAI , 1982.