Determining The Level of Expertise of a User of a Question Answering System

Cecile Paris

March, 1984

CUCS-115-84

This research was supported in part by the Defense Advanced Research Projects Agency under contract N00039-82-C-0427.

Determining the Level of Expertise of a User of a Question Answering System

Abstract

An intelligent question answering program should be able to tailor its answer to the user. Some factors entering this tailoring process include the level of expertise of the user, his goals for using the system, the discourse structure, and the user type. The decision on how sophisticated and detailed an answer should be is based, in part, on how much the user knows about the domain in question. In this paper, we are mainly concerned with determining the level of expertise of the user. We will show how a generalization based memory can be used in this process.

1. Introduction

When engaged in conversation, people try to speak in such a way that their interlocutors will understand them. Speakers choose carefully both the content of their utterances and the words they use. (Note that this process is not always totally conscious). One important decision is how sophisticated and detailed an utterance should be. Level of detail will depend in part on how knowledgeable a person thinks their interlocutor is. For example, the explanation of how a car engine functions will be different depending on whether it is aimed at a child or an adult, a musician or a mechanical engineering student.

A question answering program will also need the capability to tailor its answers to

the user. In order to do so, it must maintain a model of the user [Rich 79; Allen and Perrault 78; Appelt 81]. In this paper, we are mainly concerned with determining the level of expertise of the user.

2. Our domain: RESEARCHER

In order to study the level of expertise of the user, we need an application involving a complex database that is detailed enough so that answers can be given at different levels of complexity.

At Columbia University, we are developing a program, RESEARCHER, that reads, remembers, and generalizes from patent abstracts written in English [Lebowitz 83a]. The abstracts describe complex objects in some detail. Therefore, the memory resulting from reading many abstracts necessarily contains a large amount of information about each of the objects described in the patents.

The following patent illustrates how much information is available for a sample object:

> A hard fixed head disc drive assembly having a rotating record disc with a transducer cooperating with the surface of the disc. The transducer is mounted on a carriage which has three spaced, grooved bearings, 2 of which are received by a fixed cylindrical track, the third bearing engages a spring loaded cylindrical track which urges said first 2 bearings against said fixed track, whereby the carriage is centered on said tracks for movement therealong radially of said disc surface.

Given this information, we can imagine several possible answers, varying widely in detail, for a single question. For example, consider the question Q1 and the two answers A1 and A2:

Q1: What is a head assembly made of?

A1: A head assembly has a rotating record disc, a transducer, and a carriage for the transducer.

A2: A head assembly has a disc, a transducer, and a carriage for the transducer; the carriage has grooved bearings which are used to engage a cylindrical track to center the disc on some tracks.

The first answer is rather general, and it would be appropriate in several situations: if the question was asked at the beginning of a question-answering session; if the program had already established that the user did not know much about disc drives and should not be flooded with details that would not be understood; or if the user was only interested in getting a general picture of a head assembly.

On the other hand, we can also imagine cases where the second answer would be preferred over the first one: if the question arose during the session after a general picture of a head assembly had already been given; if the program decided that the user was somewhat knowledgeable about disc drives (and so could understand why the carriage should have grooved bearings); or if we know that the user is really interested in all the parts of a head assembly. In such a case, we don't want to require the user to ask many questions in order to get the desired information. Rather, we want to recognize that the user wants detailed information and provide it immediately. We see that the amount of detail can vary with the knowledge the user has about the domain. The need to determine the level of expertise of the user thus arises.

3. Determining expertise

Intuitively, a person who talks only in very general terms about an object probably does not know much about it. Quite the contrary, knowledge about an *obscure* part of an object usually indicates that a person has some expertise about the domain.

As an example, consider the question Q2.

Q2: How are the 3 spaced bearings of a disc drive head assembly connected with the tracks ?

It is certainly not obvious to everyone that a head assembly should include three spaced bearings and therefore we can assume that the user has some knowledge about head assemblies. '3 spaced bearings' constitute obscure parts of a head assembly, and the fact that the user knows about them indicates a certain level of expertise in the domain.

3.1 Role of the memory organization

How can we determine that a part is obscure? This is where the memory organization can play a role. Memory in RESEARCHER is organized in terms of generalizations [Lebowitz 83a; Lebowitz 83b]. As it reads patents, RESEARCHER looks for similarities among the objects described, so that it can make generalizations from them, and organize its memory around those generalizations.

The resulting memory is largely hierarchical, consisting basically of several trees of specific generalizations and individual instances occurring at the leaves. The top

node in the generalization tree contains information common to all instances of that generalization, while instance nodes only contain information special to that instance. This complex, hierarchical structure helps us determine what an obscure part is. When a user is knowledgeable about a part used only deep in the generalization tree (the extreme case being a part which is particular only to an instance at the bottom of the tree), then we can assume expertise in that subdomain. On the other hand, if only parts at a top level node of the generalization tree are mentioned (i.e. the user knows only about information common to a whole class of objects), then we are probably dealing with a novice in that particular subdomain. This technique is helpful in determining the level of expertise of the user in any application using a hierarchical knowledge base.

We have specified a method for determining whether an object is obscure or not. But we need to be careful in applying it for a distinction between *knowing* (as in "How many bearings are they?", where the user knows what bearings are), and merely *mentioning* an item (as in "What is a bearing?", where, obviously, the user does not know what a bearing is) must be made. Furthermore, a term may be misused (as in "Which filters have flying heads", when a filter cannot have a flying head).

3.2 Radius of expertise

Expertise in a subpart of a domain does not necessarily imply expertise in the whole domain. As an example, a user could be an expert in the domain of disc drives without knowing much about the computers that use them (or vice versa: he could know about computers without really knowing much about disc drives). Therefore, we introduce the idea of a *radius of expertise*. When a user mentions a part deep in generalization tree, we infer expertise with respect to that part; we can also establish expertise with respect to a number of objects related to that part. We can thus define an area around the specified particular part of the generalization tree mentioned that corresponds to the person's domain of expertise. Outside this area, the user is not considered to be an expert, unless that fact has previously been established. This idea of radius of expertise relates in some ways to Grosz's use of global focus [Grosz 77]. In the same way as Grosz's global focus guides the interpretation of a dialogue, the radius of expertise can guide the amount of information to give a user as an answer.

3.3 User type

Finally, we note that the system may be able to identify the type of the user, which gives a priori information on the probable level of expertise.

We have identified three plausible categories of potential users for RESEARCHER:

- Inventors who invented a device and want to check what has already been patented in the domain of their inventions. Inventors are experts in their field, but, on the other hand, they may be novices as far as doing a patent search is concerned.

- Lawyers who perform patent searches for their clients. Unlike the inventors, they are experts with respect to doing a patent search, but may be novice with respect to the contents of the patents.

- General users who are people who want to know what kind of information is available from the database. Such a person could be either an expert or a novice.

4. Current status and future work

At this time, we have implemented some simple question/answering routines for RESEARCHER. We are now beginning the incorporation of the analysis described here as part of the system. However, several theoretical questions remain to be analyzed more carefully before the full implementation of the question/answering program can be completed.

Once we have determined the level of expertise of a user, we can use it to include technical details for an expert, and abstract ones for a novice. The question remains as to how to determine exactly what details should be included in the response. We are currently examining how to use the identification of the level of expertise to limit the possible content (i.e., using the radius of expertise to incorporate details at one level in the tree) as well as discourse strategies which dictate exactly how to provide more detail [McKeown 83].

Of the factors we considered including in the user model, the formulation of the use of memory organization is most complete. We expect to do further development of the use of discourse goals, term misuse, and other factors which indicate knowledge level, such as user goals. Part of the work will include how to represent these in the user model and how to update the model as the dialogue progresses.

5. Conclusion

Being able to tailor an answer to a particular user is important when dealing with a complex database, because users have different levels of expertise about the domain and different goals for using the system. We have seen that knowing the level of expertise of the user is necessary. Furthermore, we have shown that the memory organization (which helps us determine the obscurity of a part), the discourse goal, and term misuse are factors contributing to determine that level of expertise. The role of memory organization has been most fully developed at this time, and we are continuing to develop the other factors.

Acknowledgements

We would like to thank Michael Lebowitz and Kathy McKeown for helping in both the research and the writing of this paper.

References

[Allen and Perrault 78]
Allen, J. and Perrault, C. R.
Participating in Dialogues: Understanding via Plan Deduction.
In *Proceedings of the second National Conference of the Canadian Society of Computational Studies of Intelligence*. Canadian Society for Computational Studies of Intelligence, 1978.

[Appelt 81] Appelt, D. E.
Planning Natural Language Utterances to Satisfy Multiple Goals.
PhD thesis, Stanford University, Stanford, Ca, 1981.
also a Technical report No 259, SRI Internatinal, Artificial Intelligence Center, 1982.

[Grosz 77] Grosz, B. J.
The representation and use of focus in dialogue understanding..
Technical Report Technical note 151, Stanford Research Institute, 1977.

[Lebowitz 83a] Lebowitz, M.
RESEARCHER: An overview.
In *Proceedings of the Third National Conference on Artificial Intelligence*. American Association of Artificial Intelligence, Washington, DC, 1983.

[Lebowitz 83b] Lebowitz, M.
Generalization from natural language text.
Cognitive Science 7(1):1 - 40, 1983.

[McKeown 83] McKeown, K. R.
Recursion in Text and its Use in Language Generation.
In *Proceedings of the National Conference on of Artificial Intelligence*. American Association of Artificial Intelligence, Washington, DC, 1983.

[Rich 79] Rich, E. A.
User Modeling Via Stereotypes.
Cognitive Science 3:329 - 354, 1979.