Natural Language for Expert Systems: Comparisons with Database Systems

Kathleen R. McKeown
Department of Computer Science
Columbia University
New York, N.Y. 10027

1 Introduction

Do natural language database systems still provide a valuable environment for further work on natural language processing? Are there other systems which provide the same hard environment for testing, but allow us to explore more interesting natural language questions? In order to answer ***no*** to the first question and ***yes*** to the second (the position taken by our panel's chair), there must be an interesting language problem which is more naturally studied in some other system than in the database system.

We are currently working on natural language for expert systems at Columbia and thus, expert systems provide a natural alternative environment to compare against the database system. The relatively recent success of expert systems in commercial environments (e.g. Stolfo and Vesonder 83, McDermott 81) indicates that they meet the criteria of a hard test environment. In our work, we are particularly interested in developing the ability to generate explanations that are tailored to the user of the system based on the previous discourse. In order to do this in an interesting way, we assume that explanation will be part of natural language dialog with the system, allowing the user maximum flexibility in interacting with the system and allowing the system maximum opportunity to provide different explanations.

The influence of the discourse situation on the meaning of an utterance and the choice of response falls into the category of pragmatics, one of the areas of natural language research which has only recently begun to receive much attention. Given this interesting and relatively new area in natural language research, my goals for the paper are to explore whether the expert system or database system better supports study of the effect of previous discourse on current responses and in what ways.

[1]The work described in this paper is partially supported by ONR grant N00014-82-K-0256.

2 Pragmatics and Databases

There have already been a number of efforts which investigate pragmatics in the database environment. These fall into two classes: those that are based on Gricean principles of conversation and those that make use of a model of possible user plans. The first category revolves around the ability to make use of all that is known in the database and principles that dictate what kind of inferences will be drawn from a statement in order to avoid creating false implicatures in a response. Kaplan (79) first applied this technique to detect failed presuppositions in questions when the response would otherwise be negative and to generate responses that correct the presupposition instead[2]. Kaplan's work has only scratched the surface as there have followed a number of efforts looking at different types of implicatures, the most recent being Hirschberg's (83) work on scalar implicature. She identifies a variety of orderings in the underlying knowledge base and shows how these can interact with conversational principles both to allow inferences to be drawn from a given utterance and to form responses carrying sufficient information to avoid creating false implicatures[3]. Webber (83) has indicated how this work can be incorporated as part of a database interface.

The second class of work on pragmatics and language for information systems was initiated by Allen and Perrault (80), and Cohen (78) and involves maintaining a formal model of possible domain plans, of speech acts as plans, and of plausible inference rules which together can be used to derive a

[2]Kaplan's oft-quoted example of this occurs in the following sequence. If response (B) were generated, the false implicature that CSE110 was given in Spring '77 would be created. (C) corrects this false presupposition and entails (B) at the same time.

A: How many students failed CSE110 in Spring '77?
B: None.
C: CSE110 wasn't given in Spring '77.

[3]For example, knowledge about set membership allows the inference that not all the Bennets were invited to be drawn from response (E) to question (D):

D: Did you invite the Bennets?
E: I invited Elizabeth.

speaker's intended meaning from a question. Their work was done within the context of a railroad information system, a type of database. As with the Gricean-based work, their approach is being carried on by others in the field. An example is the work of Carberry (83) who is developing a system which will track a user's plans and uses this information to resolve pragmatic overshoot. While this work has not been done within a traditional database system, it would be possible to incorporate it if the database were supplemented with a knowledge base of plans.

All of these efforts make use of system knowledge (whether database contents or possible plans), the user's question, and a set of rules relating system knowledge to the question (whether conversational principles or plausible inference rules) to meet the user's needs for the current question. That this work is relatively recent and that there is promising ongoing work on related topics indicates that the database continues to provide a good environment for research issues of this sort.

3 Extended Discourse

What the database work does not address is the influence of previous discourse on response generation. That is, given what has been said in the discourse so far, how does this affect what should be said in response to the current question[4]. Our work addresses these questions in the context of a student advisor expert[5] system. To handle these questions, we first note that being able to generate an explanation (the type of response that is required in the expert system) that is ***tailored*** to a user requires that the system be capable of generating different explanations for the same piece of advice. We have identified 4 dimensions of explanation which can each be varied in an individual response: point of view, level of detail, discourse strategy, and surface choice.

For example, in the student advisor domain, there are a number of different points of view the student can adopt of the process of choosing courses to take. It can be viewed as a state model process (i.e., "***what should be completed at each state in the process?***"), as a semester scheduling process (i.e., "***how can courses fit into schedule slots?***"), as a process of meeting requirements (i.e., "***how do courses tie in with requirement sequencing?***"), or as process of achieving a balanced workload. Given these different points of view, a number of different explanations of the same piece of advice (i.e., ***yes***) can be generated in response to the question, "***Should I take both discrete math and data structures next semester?***":

- **State Model**: Yes, you usually take them both first semester sophomore year.
- **Semester Scheduling**: Yes, they're offered next semester, but not in the spring and you need to get them out of the way as soon as possible.
- **Requirements**: Yes, data structures is a requirement for all later Computer Science courses and discrete math is a co-requisite for data structures.
- **Workload**: Yes, they complement each other and while data structures requires a lot of programming, discrete does not.

To show that the expert system environment allows us to study this kind of problem, we first must consider what the obvious natural language interface for an expert system should look like. Here it is necessary to examine the full range of interaction, including both interpretation and response generation, in order to determine what kind of discourse will be possible and how it can influence any single explanation. A typical expert system does problem-solving by gathering information relevant to the problem and making deductions based on that information. In some cases, that information is gathered from a system environment, while in others, the information is gathered interactively from a user. This paper will be limited to backward chaining systems that gather information interactively as these provide a more suitable environment for natural language (in fact, it is unclear how natural language would be used at all in other systems, except to provide explanations after the system has produced its advice).

In a backward chaining system, the expert system begins by pursuing a goal (for example, to diagnose the patient as having myocardia). To ascertain whether the goal holds or not, the system gathers information from the user often using multiple choice questions to do so. Depending on the answer given to a single question, the system forms a partial hypothesis and asks other questions based on that hypothesis.

If natural language were used in place of such a menu-like interface, the interaction might look somewhat different. Instead of the system beginning by asking questions, the user might initiate interaction by suggesting a plausible goal (for example, a likely diagnosis) supporting it with several justifications. The system could use this information to decide which goal to pursue first and to fill in information which the system would otherwise have

[4]Note that some natural language database systems do maintain a discourse history, but in most cases this is used for ellipsis and anaphora resolution and thus, plays a role in the ***interpretation*** of questions and not in the generation of responses.

[5]This system was developed by a seminar class under the direction of Salvatore Stolfo. We are currently working on expanding the capabilities and knowledge of this system to bring it closer to a eneral roblem solvin sstem Matthews 84.

had to request using menu questions. Alternatively, if the system has several top-level problem solving capabilities (e.g., perform a diagnosis or recommend drug treatment), the user's initial question may indicate which of these problem solving capabilities is being called on. . Again, the system can use this information to avoid asking a question it would normally have to ask. The use of natural language as an "overlay" on an underlying menu system to allow the user to directly state his/her goals, to skip irrelelvant questions, and to provide information to a sequence of menu questions in a single utterance is an issue we are currently exploring at Columbia.

For example, the underlying expert system in the student advisor domain has two possible problem solving capabilities: it can help the student plan next semester's schedule or it can provide information about university courses. Using the menu interface, the system would first query the user to ask which of these is desired. If the student chose to plan the next semester schedule, the system next asks a series of questions to determine the student's year, what courses have already been taken, and what courses the student wants to take. A hypothetical natural language interaction (towards which we are working) is shown below. Note that the system can avoid asking three questions in this case (the required problem solving capability, the student's year, and the courses already taken) as these are all indicated in the first user utterance. In the last system response, the system provides its advice with justification:

1) User: I'm a sophomore and just finished data structures. What courses should I take this spring?
2) System: Have you taken discrete math?
3) User: Yes.
4) System: Are there any courses you'd particularly like to take?
5) User: I'd like to take less programming courses.
6) System: I suggest you take fundamental algorithms and finite math which are both offered next semester You should have finished both courses by the end of your sophomore year and only fundamental algorithms requires programming.

There are a number of ways in which this type of discourse allows us to address our objectives of taking previous discourse into account to generate tailored responses. This discourse segment is clearly concerned with a single purpose which is stated by the user at the beginnning of the session[6] This is the goal that the expert system must pursue and the ensuing discourse is directed at gathering information and defining criteria that are pertinent to this goal. Since the system must ask the user for information to solve the problem, the user is given the opportunity to provide additional relevant information. Even if this information is not strictly necessary for the problem-solving activity, it provides information about the user's plans and concerns and allows the system to select information in its justification which is aimed at those concerns. Thus, in the above example, the system can use the volunteered information that the user is a sophomore and wants to take less programming courses to tailor its justification to just those concerns, leaving out other potentially relevant information.

Is this type of extended discourse, revolving around an underlying goal, possible in the database domain? First, note that extended discourse in a natural language database system would consist of a sequence of *questions* related to the same underlying goal. Second, note that the domain of the database has a strong influence on whether or not the user is likely to have an underlying goal requiring a related sequence of questions. In domains such as the standard suppliers and parts database (Codd 78), it is hard to imagine what such an underlying goal might be. In domains such as IBM's TQA town planning database (Petrick 82), on the other hand, a user is more likely to ask a series of related questions.

Even in domains where such goals are feasible, however, the sequence of questions is only implicitly related to a given goal. For example, suppose our system were a student advisor database in place of an expert system. As in any database system, the user is allowed to ask questions and will receive answers. Extended discourse in this environment would be a sequence of questions which gather the information the user needs in order to solve his/her problem. Suppose the user again has the goal of determining which courses to take next semester. S/he might ask the following sequence of questions to gather the information needed to make the decision:

1. What courses are offered next semester?
2. What are the pre-requisites?
3. Which of those courses are sophomore level courses?
4. What is the programming load in each course?

[6] Over a longer sequence of discourse, more than a single user goal is likely to surface. I am concerned here with *discourse segments* which deal with a sinle or related set of oals.

Although these questions are all aimed at solving the same problem, the problem is never clearly stated. The system must do quite a bit of work in inferring what the user's goal is as well as the criteria which the user has for how the goal is to be satisfied. Furthermore, the user has the responsibility for determining what information is needed to solve the problem and for producing the final solution.

In contrast, in the expert system environment, the underlying expert system has responsibility coming up with a solution to the given problem and thus, the natural language system is aware of information needed to solve that goal. It can use that information to take the responsibility for directing the discourse towards the solution of the goal (see Matthews 84). Moreover, the goal itself is made clear in the course of the discourse. Such discourse is likely to be segmented into discernable topics revolving around the current problem being solved. Note that one task for the natural language system is determining where the discourse is segmented and this is not necessarily an easy task. When previous discourse is related to the current question being asked, it is possible to use it in shaping the current answer. Thus, the expert system does provide a better environment in which to explore issues of user modeling based on previous discourse.

4 Conclusions

The question of whether natural language database systems still provide a valuable environment for natural language research is not a simple one. As evidenced by the growing body of work on Gricean implicature and user modelling of plans, the database environment is still a good one for some unsolved natural language problems. Nevertheless, there are interesting natural language problems which cannot be properly addressed in the database environment. One of these is the problem of tailoring responses to a given user based on ***previous discourse*** and for this problem, the expert system provides a more suitable testbed.

References

(Allen and Perrault 80). Allen, J. F. and C. R. Perrault, "Analyzing intention in utterances," *Artificial Intelligence 15*, 3, 1980.

(Carberry 83). Carberry, S., "Tracking user goals in an information-seeking environment," in *Proceedings of the National Conference on Artificial Intelligence*, Washington D.C., August 1983. pp. 59-63.

(Codd 78). Codd, E. F., et. al., Rendezvous Version 1: An Experimental English-Language Query Formulation System for Casual Users of Relational Databases, IBM Research Laboratory, San Jose, Ca., Technical Report RJ2144(29407), 1978.

(Cohen 78). Cohen, P., On Knowing What to Say: Planning Speech Acts, Technical Report No. 118, University of Toronto, Toronto, 1978.

(Grice 75). Grice, H. P., "Logic and conversation," in P. Cole and J. L. Morgan (eds.) *Syntax and Semantics: Speech Acts*, Vol. 3, Academic Press, N.Y., 1975.

(Hirschberg 83). Hirschberg, J., Scalar quantity implicature: A strategy for processing scalar utterances. Technical Report MS-CIS-83-10, Dept. of Computer and Information Science, University of Pennsylvania, Philadelphia, Pa., 1983.

(Kaplan 79). Kaplan, S. J., Cooperative responses from a portable natural language database query system. Ph. D. dissertation, Univ. of Pennsylvania,Philadelphia, Pa., 1979.

(Matthew 84). Matthews, K. and K. McKeown, "Taking the initiative in problem solving discourse," Technical Report, Department of Computer Science, Columbia University, 1984.

(McDermott 81). McDermott, J., "R1: The formative years," *AI Magazine* 2:21-9, 1981.

(Petrick 82). Petrick, S., "Theoretical /Technical Issues in Natural Language Access to Databases," in *Proceedings of the 20th Annual Meeting of the Association for Computational Linguistics*, Toronto, Ontario, 1982. pp. 51-6.

(Stolfo and Vesonder 82). Stolfo, S. and G. Vesonder, "ACE: An expert system supporting analysis and management decision making," Technical Report, Department of Computer Science, Columbia University, 1982, to appear in *Bell Systems Technical Journal*.

(Webber 83). "Pragmatics and database question answering," in *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, Karlsruhe, Germany, August 1983, pp. 1204-5.