NATURAL-LANGUAGE ACCESS TO DATABASES--THEORETICAL/TECHNICAL ISSUES

Robert C. Moore
Artificial Intelligence Center
SRI International, Menlo Park, CA 94025

I INTRODUCTION

Although there have been many experimental systems for natural-language access to databases, with some now going into actual use, many problems in this area remain to be solved. The purpose of this panel is to put some of those problems before the conference. The panel's motivation stems partly from the fact that, too often in the past, discussion of natural-language access to databases has focused, at the expense of the underlying issues, on what particular systems can or cannot do. To avoid this, the discussions of the present panel will be organized around issues rather than systems.

Below are descriptions of five problem areas that seem to me not to be adequately handled by any existing system I know of. The panelists have been asked to discuss in their position papers as many of these problems as space allows, and have been invited to propose and discuss one issue of their own choosing.

II QUANTITY QUESTIONS

Database query languages typically provide some means for counting and totaling that must be invoked for answering "how much" or "how many" questions. The mapping between a natural-language question and the corresponding database query, however, can differ dramatically according to the way the database is organized. For instance, if DEPARTMENT is a field in the EMPLOYEE file, the database query for "How many employees are in the sales department?" will presumably count the number of records in the EMPLOYEE file that have the appropriate value for the DEPARTMENT field. On the other hand, if the required information is stored in a NUMBER-OF-EMPLOYEES field in a DEPARTMENT file, the database query will merely return the value of this field from the sales department record. Yet a third case will arise if departments are broken down into, say, offices, and the number of exployees in each office is recorded. Then the database query will have to *total* the values of the NUMBER-OF-EMPLOYEES field in all the records for offices in the sales department. In each case, the English question is the same, but the required database query is radically different. Is there some unified framework that will encompass all these cases? Is this a special case of a more general phenomenon?

III TIME AND TENSE

This is a notorious black hole for both theoretical and computational linguistics, but, since many databases are fundamentally historical in character, it cannot really be circumvented. There are many problems in this general area, but the one I would suggest is how to handle, within a common framework, both concepts defined with respect to points in time and concepts defined with respect to intervals. The location of an object is defined relative to a point; it makes sense to ask "Where was the Kennedy at 1800 hours on July 1, 1980?" The distance an object has traveled, however, is defined solely over an interval; it does *not* make sense to ask "How far did the Kennedy sail at 1800 hours on July 1, 1980?" Or, to turn things around, "How far did the Kennedy sail during July 1982?" has only a single answer (for the entire interval)--but "Where was the Kennedy during July 1982?" may have many different answers (in the extreme case, one for each point in the interval). Must these queries be treated as two completely distinct types, or is there a unifying framework for them? If they are treated separately, how can a system recognize which treatment is appropriate?

The fact that any interval contains an infinite number of points creates a special problem for the representation of temporal information in databases. Typically, information about a time-varying attribute such as location is stored as samples or snapshots. We might know the position of a ship once every hour, but obviously we cannot have a record in an extensional database for *every* point in time. How then are we to handle questions about specific points in time not stored in the database, or questions that quantify over periods of time? (E.g., "Has the Kennedy ever been to Naples?") Interpolation naturally suggests itself, but is it really appropriate in all cases?

IV QUANTIFYING INTO QUESTIONS

Normally, most of the inputs to a system for natural-language access to databases will be questions. Their semantic interpretation, however, is not yet completely understood. In particular, quantifiers in questions can cause special problems. In speech act theory, it is generally assumed that a question can be analyzed as a having a propositional content, which is a description, and an illocutionary force, which is a request to enumerate the entities that satisfy the description. Questions such as "Who manages each department?" resist this simple analysis, however. If "each" is to be analyzed as a universal quantifier (as in "Does each department have a manager?"), then its scope, in some sense, must be wider than that of the indicator of the sentence's illocutionary force. That is, what the question actually means is "For each department, who manages the department?" If we to try to force the quantifier to be part of the description of the entities to be enumerated, we seem to be asking for a single manager who manages every department--i.e., "Who is the manager such that he manages each department?" The main issues are: What would be a suitable representation for the meaning of this sort of question, and what would be the formal semantics of that representation?

V QUERYING SEMANTICALLY COMPLEX FIELDS

Natural-language query systems usually assume that the concepts represented by database fields will always be expressed in English by single words or fixed phrases. Frequently, though, a database field will have a complex interpretation that can be interrogated in many different ways. For example, suppose a college admissions office wants to record which applicants are children of alumni. This might be indicated in the database record for each applicant by a CHILD-OF-ALUMNUS field with the possible values T or F. If this field were queried by asking "Is John Jones a child of an alumnus?" then "child of of an alumnus" could be treated as if it were a fixed phrase expressing a primitive predicate. The difficulty is that the user of the system might just as well ask "Is one of John Jones's parents an alumnus?" or "Did either parent of John Jones attend the college?" Can anything be done to handle cases like this, short of treating an entire question as a fixed form?

VI MULTIFILE QUERIES

All the foregoing examples involve questions that can be answered by querying a single file. In a multifile database, of course, questions will often arise that require information from more than one file, which raises the issue of how to combine the information from the various files involved. In database terms, this often comes down to forming the "join" of two files, which requires deciding what fields to compute the join over. In the LADDER system developed at SRI, as well as in a number of other systems, it was assumed that for any two files there is at most a single pair of fields that is the "natural" pair of fields to join. For instance, in a SHIP file there may be a CLASS field containing the name of the class to which a ship belongs. Since all ships in the same class are of the same design, attributes such as length, draft, speed, etc., may be stored in a CLASS file, rather than being given separately for each ship. If the system knows that the natural join between the two files is from the CLASS field of the SHIP file to the CLASSNAME field of the CLASS file, it can retrieve the length of a particular ship by computing this join.

The scheme breaks down, however, when there is more than one natural join between two files, as would be the case if there were a PORT file and fields for home port, departure port, and destination port in the SHIP file. This is sometimes called the "multipath problem." Is there is a solution to this problem in the general case? If not, what is the range of special cases that one can reasonably expect to handle?