TRANSPORTABLE NATURAL-LANGUAGE INTERFACES: PROBLEMS AND TECHNIQUES

Barbara J. Grosz

Artificial Intelligence Center
SRI International, Menlo Park, CA 94025

Department of Computer and Information Science[1]
University of Pennsylvania, Philadephia, PA 19104

I OVERVIEW

I will address the questions posed to the panel from within the context of a project at SRI, TEAM [Grosz, 1982b], that is developing techniques for transportable natural-language interfaces. The goal of transportability is to enable nonspecialists to adapt a natural-language processing system for access to an existing conventional database. TEAM is designed to interact with two different kinds of users. During an acquisition dialogue, a database expert (DBE) provides TEAM with information about the files and fields in the conventional database for which a natural-language interface is desired. (Typically this database already exists and is populated, but TEAM also provides facilities for creating small local databases.) This dialogue results in extension of the language-processing and data access components that make it possible for an end user to query the new database in natural language.

A major benefit of using natural language is that it shifts onto the system the burden of mediating between two views of the data--the way in which it is stored (the "database view") and the way in which an end user thinks about it (the "user's view"). Basically, database access is done in terms of files, records, and fields, while natural-language expressions refer to the same information in terms of entities and relationships in the world.

In my discussion, I will assume the use of a general grammar of English rather than a semantic grammar, and also that the interpretation of queries will pass through an intermediate stage in which a database-independent representation of the meaning of the query is derived before constructing the formal database query. This is because systems based on semantic grammars amalgamate information about language, about the domain, and about the database in ways that make it difficult to transfer those systems to new databases. I will use the term "conceptual schema" to refer to the internal representation of information about the entities in the domain of discourse and the relationships that can hold among them,[2] and "database schema" to refer to the encoding of information about the way concepts in the conceptual schema map onto the structures of the database. In addition, I will use the term "logical form" to refer to the representation of the literal meaning of an expression in the context of an utterance.

The insistence on transportability (which distinguishes TEAM from previous systems such as LADDER [Hendrix et al., 1978], LUNAR [Woods, Kaplan, and Webber, 1972], PLANES [Waltz, 1975], REL [Thompson, 1975], and CHAT [Warren, 1981]) entails two major consequences for the design of a natural-language interface. First, the database cannot be restructured to make the way in which it stores data more compatible with the way in which a user would pose his questions. Second, because the DBE cannot be expected to know about the internal structure of the conceptual schema and the database schema, these must be organized so that the information they encode about any particular database and its corresponding domain can be obtained systematically (and, therefore, automatically).

These differences are crucial to any consideration of the issues before this panel. Although, for any particular database, it may be possible to handcraft solutions to each problem, such an approach is not viable for a transportable system. Handcrafting requires expertise in computational linguistics, knowledge of the internal structures and algorithms used in an interface, and so forth--none of which the DBE can be expected to possess. In addition, interfacing to an existing conventional database introduces many problems caused by the difference between the database view and the end user's view. Many of these problems can be avoided if one is allowed to design the database as well as the natural-language system. However, given the prevalence of existing conventional databases, approaches that make this assumption are likely to have limited applicability in the near future.

Most of the issues the panel has been asked to address arise (or have analogues) in any

---

[1] Currently visiting under the auspices of the Program in Cognitive Science at the University of Pennsylvania.

[2] This schema is a restricted form of the standard AI knowledge base.

application of natural-language processing. In the sections that follow, my objective in discussing each of these issues will be to point out where I see the constraints of the database query task as simplifying the general problem and where, on the other hand, transportability (and the way in which database systems typically structure information and view the world) makes things more difficult. Inevitably, I will be raising at least as many questions as I answer.

II AGGREGATES

It is useful to separate problems involving aggregates into two categories: (1) those that involve mapping from natural-language to logical form, and (2) those that involve translating from logical form into a formal database query. The examples presented to the panel have elements of each of these.

In addressing the question of logical form, I first want to note how similar "how many" and "how much" questions are to other degree questions (e.g., "How tall is John?"). Consider, for example,

(1) James is old./ How old is James?

(2) The department is big./ How big is the department?

(3) The department has many employees./ How many employees does the department have?

(4) The ship is heavy./ How heavy is the ship?

(5) The ship is carrying much coal./ How much coal is the ship carrying?

Hence, it seems that the logical forms for the queries ought to bear a close resemblance. In interpreting degree questions, the language-processing component of TEAM [Grosz et al., 1982a], applies a higher-order degree operator to the predicate that underlies the adjective. For example, the logical form for "How tall is John?" would be

(WHAT H (HEIGHT H) ((DEGREE TALL) JOHN H))

The problem in transferring this treatment to "how many" and "how much" questions is that while adjectives like "heavy" are usually treated as predicates, "many" is usually treated as a quantifier. So, if "how" is treated by uniformly applying some kind of higher-order degree operator, then that operator has to apply to both predicates and quantifiers. Another possibility would be to apply the degree operator to an entire formula, as in

(WHAT H (HEIGHT H) ((DEGREE (TALL JOHN) H))

rather than just to the head of the formula. Whether this can be made to work, however, depends on whether a satisfactory analysis can be provided when the formula consists of more than just a predicate and its arguments.

The problem of an appropriate logical form for these questions is not affected by the need for transportability. However, transportability does make the problem of translating from logical form into a database query more difficult. Fields that store count totals, like NUMBER-OF-EMPLOYEES, are semantically complex in much the same way as the CHILD-OF-ALUMNUS field (the predicate encoded by a count field can be defined in terms of a count operator and the domain entities that are to be counted), and they present similar problems for transportability and database access (see section 5). The question therefore (to which I do not have an answer) is whether this kind of semantically complex field is any simpler to handle than the more general case.

In addition, some ways of storing information about aggregates in these semantically complex fields may require inferences to be drawn to answer certain kinds of queries. For example, if the number of employees in a department must be calculated from the number of employees in each office of the department, answering queries about the number of employees in a department will require reasoning about the part/whole relationship between offices and departments and how the number of employees in a department depends on that relationship. A general treatment of such cases would require both the acquisition of information about the part/whole relationship implicitly encoded in the database, and the ability to infer that (in this case) the count for the whole is the sum of the counts for the parts.

The need for drawing inferences arises with mass fields as well as with count fields. For example, consider a database of ships and their cargoes, with separate entries for the different kinds of cargo a ship is carrying. Then an answer to "How much cargo is the ship carrying?" will require the same kind of totaling operation as does the query about the number of employees in the above example. It may be possible to handle the most straightforward cases of these phenomena by adding special purpose information ("hacks" to compensate for the lack of theorem-proving capabilites) for each operator corresponding to a data access system aggregate function, specifying how it interacts with part/whole relationships (AVERAGE will work differently from TOTAL).

III TIME AND TENSE

The context of database querying does not seem to make questions concerning time and tense any easier than they are for linguistics or philosophy in general; in fact, they are actually more difficult because of the extensional nature of the temporal information stored in a database.

It does not appear useful, even in the database query context, to have different representations for sentences involving concepts related to points in time and those involving intervals. The same natural-language expressions about time may be used to refer to a given time as either a point or an interval. Consider,

(6) How far did the Fox travel yesterday? (yesterday as an interval over which an event extends)

(7) Who was the officer of the day yesterday? (yesterday as a point in a sequence of days)

It is fairly easy to imagine databases against which each of these queries might be posed and, in each case, "yesterday" might correspond either to a single database entry or to a set of entries spanning an interval. Furthermore, the same verb can be used to refer to activities in terms of points or intervals--e.g.,

(8) The ship is sailing to Naples. (interval)

(9) The ship is sailing tomorrow. (point)

--and the same event may be viewed as occurring during an interval or at some single point [Moore, 1981]. (See Prince [1982] for an interesting discussion of the differences between (9) and "The ship sails tomorrow.")

On the issue of interpolation, we should note that questions involving temporal attributes also involve at least one other attribute of an entity (e.g., its location). To handle adequately queries about times not explicitly represented in the database, such factors must be taken into account as the time scale over which an attribute changes (e.g., a ship's position changes more slowly than an airplane's), and whether or not the change is linear. In general, this requires mechanisms for reasoning about temporal relationships and complex events, mechanisms normally absent in database systems. Also note that, even when interpolation is possible, additional mechanisms are needed to handle queries about times beyond the last recorded time. (I have been living in Philadelphia for the last four months, but I will not be two months hence.)

All this suggests that naive interpolation is likely to result in incorrect answers (entities may even have ceased to exist since the last data about them was recorded). I believe it is misleading to provide direct responses involving such interpolation, because the user has no way of knowing that the system's reasoning is only approximate, or knowing on what it has based its answer. If the natural-language interface isolates a user from the manner in which information is stored, it must compensate by furnishing sufficient information in its responses to allow the user to assess their validity. Of course, this is a more general issue than one concerning just time, but the appeal of interpolation (as a simple solution) may mislead us into thinking we can provide the user with an answer that later reflection will reveal as worse than no answer at all.

In an interface designed for a particular database, special purpose routines may be provided that take such factors as time scale into account. The problem is more difficult to deal with for a transportable natural-language interface, but two strategies appear possible. One is to provide the two values of the attribute being queried that correspond to times that bracket the time specified in an actual query. The second is to associate with each attribute-time pairing an interval over which the attribute value can be considered to be constant, as well as possibly a function for interpolating between values and extrapolating from them. The problem for transportability, then, is obtaining the requisite information from the DBE.

IV QUANTIFYING INTO QUESTIONS

The problem of quantifying into questions may have a simpler solution in the database query environment than it does in general. Database queries usually seek an enumeration (as opposed to queries seeking a description, as in "Which woman does every Englishman admire most? His mother." [Engdahl, 1982]). For such cases, it seems possible to analyze a question as a REQUEST to INFORM (an analysis done in [Cohen and Perrault, 1979] to allow planning of questions, taking into account plans and goals of both speakers and hearers), with REQUEST being the illocutionary-force operator. If this is done, a quantifier can outscope the INFORM without outscoping the REQUEST. Thus, the logical form of "Who commands each ship" would be something like

```
(REQUEST (EVERY X (SHIP X)
                  (INFORM "who commands X")))
```

V SEMANTICALLY COMPLEX FIELDS

The predicate represented in a semantically complex field like CHILD-OF-ALUMNUS typically has a definition in terms of simpler concepts, namely an existential quantifier and whatever entity is being quantified over (in this case ALUMNUS). In a nontransportable system, some of the variability of expression that these fields give rise to can be handled by enriching the conceptual schema appropriately (e.g., adding to it the class of alumni). However, as the query "Did either of John Jones's parents attend the college?" illustrates, this by itself is not sufficient in general.

In extreme cases, sophisticated deductive capabilities may be necessary to answer questions that can arise in connection with semantically complex fields. For example, the BLUEFILE database (to which LADDER provided an interface) has a field DOC that records whether or not a ship has a doctor on board. To answer a query like "Is there a doctor within 200 miles of Philadelphia?" requires not only represention of the connection between a positive value in the DOC field and the existence of a doctor, but also the ability to reason that, if a ship that has a doctor on board is within 200 miles of Philadelphia, then the doctor himself is within 200 miles of Philadelphia.

An apparent precondition for the correct treatment of semantically complex fields is that the system should have a richer model of the domain than the model constituted by the database itself. Konolige [1981] suggests one possible approach to this in which a metatheory is employed to describe both the domain of discourse and the information the database contains. Axioms in the metalanguage are used to encode things like the connection between the existence of an alumnus and a particular value in the CHILD-OF-ALUMNUS field.

It does not seem possible to handle a wide variety of semantically complex fields in a transportable system, unless the system is much richer than typical DB systems (in which case much more general knowledge acquisition schemes must be implemented, such as those proposed by Hendrix and Haas [1982], for example). However, transportable systems can provide for a fairly wide range of fixed phrases corresponding to these fields [Grosz et al, 1982b]).

VI MULTIFILE QUERIES

I will address only those aspects of this problem that are directly concerned with interpreting natural-language queries correctly, and not those that are concerned primarily with database access (e.g., ensuring that the fields over which the join must be made possess compatible values). Two basic problems arise in coordinating information from multiple files: (1) determining the relationships among the domains corresponding to the different fields; (2) accounting for the composition of relations across files.

It is relatively straightforward to achieve correctness in (1) even in a transportable system. The composition of relations that are introduced by joins over distinct files presents greater difficulties because natural-language queries may refer only implicitly to the composition. I want to consider two such cases: (1) the use of a field value (or a synonym) to modify a noun phrase (e.g., "Italian ships"), and (2) the use of a field value as a head noun referring to entities possessing that value for the attribute represented by the field (e.g., in a database about cars, "Fords" might refer to those cars with manufacturer=FORD).

In both cases, it may be ambiguous as to exactly what relationship is being expressed. If we restrict natural-language interface systems to handling only isolated queries, the DBE can be asked to eliminate certain of these ambiguities by establishing which fields have values that can be used to modify (or stand alone for) the entities in the database. Thus, for example, a DBE might establish that "Italian ships" will never be used to refer to ships with a port of departure in Italy.

Once discourse contexts are taken into account, the problem becomes more difficult. For any field, it is fairly easy to create a context in which the relation represented by that field can be implicitly expressed by using one of its values as a modifier. For example, following the query "Are there more ships sailing from Italy or France this month?", the query "What cargoes are the Italian ships carrying?" uses "Italian ships" to refer specifically to ships departing from Italy.

VII Acknowledgments

Robert Moore and Bonnie Webber provided many helpful comments on the content and form of this paper. Many of the ideas in it have resulted from discussions among the members of the TEAM project at SRI. The TEAM project is supported by the Defense Advanced Research Projects Agency under Contract N00039-80-C-0645 with the Naval Electronic Systems Command.

REFERENCES

Cohen, P. R. and C. R. Perrault [1979] "A Plan-Based Theory of Speech Acts," Cognitive Science, Vol. 3, No. 3, pp. 177-212 (July-September 1979)

Grosz, B. et al. [1982a] "DIALOGIC: A Core Natural-Language Processing System," to appear in Proceedings of the Ninth International Conference on Computational Linguistics, Prague, Czechoslovakia (July 1982)

Grosz, B. et al. [1982b] "TEAM: A Transportable Natural-Language System," Technical Note No. 263, Artificial Intelligence Center, SRI International, Menlo Park, California (April 1982).

Engdahl, E. [1982] "Constituent Questions, Topicalization, and Surface Structure Interpretation," to appear in Proceedings from the First West Coast Conference on Formal Linguistics, D. Flickinger, M. Macken, and N. Wiegand, eds., Stanford, California (1982).

Hendrix, G. G., et al. [1978] "Developing a Natural Language Interface to Complex Data," ACM Transactions on Database Systems, Vol. 3, No. 2, pp. 105-147 (June 1978).

Hendrix, G. G. and Haas, N. [1982] "Learning by Being Told: Acquiring Knowledge for Information Management," to appear in Machine Learning, R.S. Michalski, J. Carbonell, and T. Mitchell, eds. (Tioga Publishing Co., Palo Alto, California, 1982).

Konolige, K. G. [1981] "A Metalanguage Representation of Relational Databases for Deductive Question-Answering Systems," Proceedings of the Seventh International Joint Conference on Artificial Intelligence, pp. 496-503, Vancouver, British Columbia, Canada (August 24-28, 1981).

Moore, R. C. [1981] "Problems in Logical Form," Proceedings of the 19th Annual Meeting of the Association for Computational Linguistics, pp. 117-124, Stanford University, Stanford, California (June 29-July 1, 1981).

Prince, E. [1982] "The Simple Futurate: Not Simply Progrssive Futurate Minus Progressive," Meeting of the Chicago Linguistics Society, Chicago, Illinois (April 1982).

Thompson, F. B., and B. H. Thompson [1975] "Practical Natural Language Processing: The REL System as Prototype," in Advances in Computers 13, M. Rubinoff and M. C. Yovits, eds. (Academic Press, New York, New York, 1975).

Waltz, D. [1975] "Natural Language Access to a Large Data Base: An Engineering Approach," Advance Papers of the Fourth International Joint Conference on Artificial Intelligence, pp. 868-872, Tbilisi, Georgia, USSR (September 1975).

Warren, D. H. [1981] "Efficient Processing of Interactive Relational Database Queries Expressed in Logic," Proc. Seventh International Conference on Very Large Data Bases, pp. 272-283, Cannes, France (September 1981).

Woods, W. A., R. M. Kaplan, and B. N-Webber [1972] "The Lunar Sciences Natural Language Information System," BBN Report 2378, Bolt Beranek and Newman, Cambridge, Massachusetts (1972).