Proceedings of the 1985 ACM Computer Science Conference—Agenda for Computing Research: The Challenge for Creativity, 1985 March 12-14

THE HUMAN FACTORS OF NATURAL LANGUAGE QUERY SYSTEMS

William C. Ogden

International Business Machines Corporation
Santa Teresa Human Factors Laboratory
San Jose, California, USA.

Understanding the hidden limitations and constraints of the system is the largest potential problem for users of natural language query (NLQ). By their nature, most NLQ systems hide "how it works" because they are intended for users who do not want to know. However, human factors research indicates that when users do not have a good understanding of a system, the behavior of the system becomes unpredictable. For example, consider the two natural language questions: "Which students have more than 20 credits?" and "Which students have more than 20 courses?" Some NLQ systems may be able to answer the first but not the second question and when the user knows that both answers can be derived from the database, the system appears to be inconsistent. To be able to use this type of NLQ system effectively, a user will have to learn the hidden system constraints that produce this type of inconsistency. There are two approaches to minimize the impact of the hidden constraints. One approach is to customize the linguistic coverage of the NLQ system for a particular user population so that most of their questions can be processed correctly. Another approach that is being investigated in our laboratory is to explicitly define a learnable and memorable subset of the language so that the limitations and constraints will no longer be hidden.

The first approach may not guarantee a solution. In order to gain good linguistic coverage of a domain, the system must initially have a powerful enough linguistic capability to be able to represent a deep structure of the processed sentence. In addition, however, the grammar and lexicon needs to be augmented with semantic and pragmatic information about the task domain and linguistic requirement of the users. For each application of a NLQ system a great deal of effort is required to capture, define and enter this information into the system. This requires a user who is knowledgeable about how to acquire this information and how to translate this information into the form required by the NLQ program. Thus, the human factors of the NLQ application depends on the ability of knowledgeable users to initially supply this information to the system as well as their ability to maintain the system as it changes and as the needs of the users change. It is unlikely that this ability will be uniformly present in all environments in which a NLQ system could be applied. Therefore this approach to solving the human factors problem of these systems will likely produce mixed results. This approach, then, may be feasible only in situations where a great deal of work has been done at understanding the task domain such as in the domains in which expert systems have been developed.

The second approach may not be in the spirit of providing unconstrained natural language to end user populations, but it is an approach that promises to be a more general solution to the human factors problems associated with NLQ systems. By restricting users to a memorable subset of natural language we are taking the burden off of the computer system (and the programmers ability to customize it) and are shifting the burden to the user who will have to learn and remember how to restrict their own natural language. However, the burden that we shift to the user should be a light one. Humans are naturally skillful at processing language and in our laboratory we are exploring what kinds of language restrictions are easy for a user to follow.

Our approach is empirical. A general methodology for obtaining a usable subset of English for query could consist of the following steps:

- Determine users' natural form of question asking within a database application domain.
- Select a subset which can be expressed as rules to be learned and followed.
- Test users, identifying which rules can or cannot be easily followed.
- Iterate the previous steps until all rules can be followed and users can still express all required retrieval requests.
- Move to other database applications.

We feel all steps are important. For example, rules not based on users' natural forms of question asking will likely not be successful. Similar-

ly, English subsets based on users' natural forms may not be successful unless the rules are communicable to the user.

A major obstacle to overcome in the study of natural forms of question writing, is to develop a task that does not bias the subjects' natural language. To overcome this problem, some researchers have given subjects open-ended problem statements that require many questions to solve. This is adequate for studying connected discourse, but the experimenter has very little control over the types of questions that can be asked. In our studies, we wanted to be able to ensure that the users would be able to express all of the data retrieval functions that are currently available in existing formal query languages. Thus, we needed to control the types of questions that subjects would be required to enter.

To meet these needs, we presented forms to subjects that contained information that was obtainable from the database. On each form some of the information was missing however, and it was the subject's task to type a question that would retrieve the missing information. The form contained enough context to indicate the retrieval keys but not enough to bias the syntax of the user's question. This technique did, of course, bias the vocabulary the subjects used. However, this bias was in a direction which represented knowledge actual users normally have about the database they use. Thus, query writing performance would not be affected by the subject's inability to think of appropriate task-related questions.

The forms were constructed to represent all of the data retrieval capability contained in a powerful formal query language such as SQL. Thus, they represented questions that were based on a relational database consisting of six tables of information about a hypothetical college, including information about students, faculty, courses, and departments. Therefore, all questions which were represented on the forms had analogous SQL solutions and covered the full range of SQL function.

Our research examined the effects that various sets of restrictions would have on the types of syntactical constructions subjects would use to express the questions indicated on the forms. The first set of studies imposed a vocabulary restriction on subject's responses. They could use only the pre-defined names of the attributes in the database but they had no restriction on how they could combine these words into a sentence. The results showed that performance was very poor. Thus, vocabulary restrictions of the type that are commonly imposed by formal query languages create difficulties for the user.

The second set of studies removed the vocabulary restrictions and showed that a large percentage of the natural queries that our subjects produced could be described with a limited set of grammatical rules. A parser based on these rules was implemented and a simple set of instructions were given to a new set of subjects. These subjects showed that they could follow the instructions and could restrict the grammatical form of their questions to the subset selected. Thus, these results indicated that users would be able to learn to use a natural subset of English for database query when the syntactic rules were exposed.

However, it became clear that even when the subjects could restrict their questions using the syntactic rules of this natural subset, a NLQ system would still require a significant amount semantic and pragmatic knowledge to be able process the questions that were asked. Thus, a large amount of customization would still be required. Therefore, the next phase of experimentation was focused on discovering the types of semantic restrictions that users would be able to learn and remember.

In addition to the syntactic restrictions, new subjects were asked to include more semantic information in their questions. Specifically, they were given a model of the database and asked to include the name or a synonym of the name of the attribute associated with each database value expressed in the query. Thus, instead of asking "What is the major of David Lee," subjects were required to ask "What is the major of the student David Lee." The results showed that users could easily specify the attribute name when selecting on a particular value of that attribute but had difficulty specifying the name when the attribute was used to calculate a value not in the database. For example, user had trouble expressing the concept of a "full class" as a class with "size greater than or equal to limit."

These results suggest that any database query system which is intended to be for general use (i.e. transportable across application domains) will require that it's users have a good understanding of what is in the database so that they will know what attributes of the data they can refer to. This suggests that a well human factored database query systems will exposed in a natural way the underlying structure of the database and then to allow a flexible vocabulary to be used to reference the items in the database that they know about.