Q-TRANS: QUERY TRANSLATION INTO ENGLISH

Eva-Maria M. Mueckstein

IBM Thomas J. Watson Research Center
Yorktown Heights, New York 10598

ABSTRACT

Q-TRANS, which stands for Query-TRANSlation System, translates formal database queries into English to enhance the usability of both natural and formal language database access systems. Q-TRANS is designed for the database query language SQL, whose query expressions serve as an abstract representation from which an English paraphrase is generated. Q-TRANS is also intended to be part of the Transformational Question Answering System (TQA system), which provides a natural language interface for database query, analyzing and ultimately translating the English queries into SQL expressions. The concepts and methods used in Q-TRANS to arrive at a query translation are, however, independent from the TQA system except for compatibility of lexical and grammatical coverage of the paraphrases produced. The paraphrases generated are true translations of the SQL expressions which are the input to Q-TRANS and serve in a sense as deep structures that get mapped into English imperatives. The grammatical English structures Q-TRANS produces obey somewhat conflicting constraints in that they preserve as much of the SQL structure as necessary to reflect the internal logic to the user, and at the same time represent as natural English sentences as possible.

I. PURPOSE AND GOALS OF Q-TRANS

Formal database query languages represent a well-defined, syntactically constrained, yet powerful means of accessing databases. Once users have mastered the syntax and lexicon of such languages, however, they still may not be successful in retrieving the desired data due to the amount of detail that needs to be specified, due to the requirement for exact matching of variables, or due to misconceptions about the contents and structure of the database. Should users be fortunate enough to have a natural language interface for database accessing, different problems arise, caused by the inherently vague and ambiguous nature of natural language. That is, the system's interpretation of English queries in terms of database queries can yield ambiguous or otherwise inappropriate results. In any case, whether a formal database language or English is the basis for communication with the database, users might benefit from an independent means of verifying whether the data retrieved is really what they wanted or asked for.

Q-TRANS has been designed to help users resolve such uncertainties by providing an English paraphrase of the formal query, regardless of whether it was produced by the user directly or indirectly via a natural language front end system. Q-TRANS is currently being developed at IBM Research in conjunction with the Transformational Question Answering System (TQA) (Petrick 1981, Plath 1976). TQA provides a natural language interface to databases, allowing users to pose their queries in English rather than in some formal language. The English input is processed in the TQA system by computing its underlying structure using a lexicon, a transformational grammar, and a transformational parser. The underlying structure is translated into logical form, a relational calculus expression, which is then mapped into SQL. SQL is a relational database language used independently of TQA for accessing, maintaining and updating databases (Astrahan et al. 1976). The output of TQA, the natural language front end, is a formal SQL expression. The latter serves as input to the back end, System R, which executes the SQL query against the database.

The SQL expression also constitutes the only direct link between TQA and Q-TRANS. Q-TRANS translates SQL expressions into English requests without utilizing the original English query or any intermediate analysis produced by TQA. The grammatical and lexical coverage of the paraphrase, however, is designed to be compatible with that of TQA. Given the Q-TRANS paraphrase, users now can check whether their original query was interpreted correctly by the system or phrased correctly by them, or - in the rare event that TQA produces more than one formal query from the English input - they can choose the appropriate one on the basis of the Q-TRANS paraphrases.

The fact that the English paraphrase is directly generated from the formal database query distinguishes this work from most other research on database response generation, which tends to rely on the natural language front end and its analysis and semantic representation of the input query (McKeown 1979, Grishman 1979, Kaplan 1977) or on template matching (Codd 1978, Waltz 1978). Whether the INTELLECT system, which is commercially available, uses either of these methods for its paraphrase is not clear. The paraphrase appears to be produced by simply inserting English-like names for the column names of the database into a sentence skeleton.

II. Q-TRANS PROCESSING

Three major steps are involved in producing the English paraphrase from the SQL expression. First, the incoming SQL expression is parsed by Q-TRANS, producing for the input string a hierarchical description which serves as the underlying structure in the translation process. This is accomplished by using a modified version of the SQL grammar which is provided to every user in the SQL manual. The original grammar is unambiguously context-free, and processing with it is extremely fast. The modifications that have been implemented consist of changes in category names and some regrouping of the original structures in the grammar so as to make them more compatible with the English structures to be generated. The end result after the modification is still unambiguously context-free.

In the second step, a surface English grammatical structure is produced from the SQL parse tree; in other words, the SQL parse tree gets mapped into an English parse tree where categories are (a) renamed, resulting in linguistic categories such as noun phrases and relative clauses; (b) reordered, allowing for post- and prenominal modifications; (c) deleted, suppressing parts of SQL statements such as join conditions; and (d) inserted, providing English function words not present in SQL.

The third step involves the actual translation of the terminal elements in the English parse tree, which are still SQL terminals, and the insertion of lexical items for the function words created in the second step.

The mechanism used to manipulate the SQL parse tree in the second and third steps is defined by a set of formal translation rules (in the form of a Knuth attribute grammar (Knuth 1968)) which are applied using Petrick's Knuth translator (Petrick 1973). The attribute grammar formalism allows for meaning assignments to nodes and rules in context-free grammars. This is done by defining any number of attributes and their respective values for the nonterminals in specific rules. Thus, for Q-TRANS, each nonterminal in the SQL grammar has, for example, a category attribute which has as its value some appropriate grammatical category or string of grammatical categories. These values get passed up and down the parse tree until all SQL categories have a grammatical value.

As for generality, the attributes assigned to each rule and category are database/application independent as are the values for attributes other than the specific column names, which have features and translations that vary from application to application. The Knuth mechanism is powerful enough to provide for the basic operations necessary to produce valid structures and translations, namely deletion, permutation, and insertion even across phrase boundaries.

III. AN EXAMPLE

To illustrate the operation of Q-TRANS, we will start with the example of an English question which belongs to a set of about 1000 queries entered by users of a previous TQA application. The database for that application contained a variety of information about properties in the City of White Plains (Damerau 1981). The main entities which are described in the database by various attributes are lots or parcels. The query reads as follows:

> What parcels in the R5 zone on Stevens St. have greater than 5000 sq. ft. ?

The current TQA system produces the following SQL expression for the query:

```
SELECT UNIQUE A.JACCN, B.PARAREA
FROM ZONEF A, PARCFL B
WHERE A.JACCN = B.JACCN
AND B.STN = 'STEVENS ST'
AND B.PARAREA > 5000
AND A.ZONE = 'R5';
```

To explain briefly, the topic of the query appears in the first line, the SELECT clause in SQL. The noun "parcels" from the input query is instantiated in the database as parcel numbers, which correspond to the first item (A.JACCN) in the SELECT clause. Note that TQA realized that the question also concerns parcel area (B.PARAREA), the second item in the SELECT clause, even though the term does not appear in the original query. Due to the structure of the database, two relations, namely ZONEF and PARCFL, have to be accessed; they are specified in the FROM clause. These two relations get joined in the first statement of the WHERE clause on the column JACCN, using the tuple variables A and B. The next three lines provide the real modifications or restrictions on the topic by specifying actual values for certain column names (STN, PARAREA, and ZONE). From this expression, Q-TRANS - as currently implemented - generates the following translation automatically:

> Find the account numbers and parcel areas for lots that have the street name STEVENS ST, a parcel area of greater than 5000 sq. ft., and zoning code R5.

Obviously, the translation provided by Q-TRANS is somewhat longer than the original input by the user; however, it is also more precise as to the items queried, namely account numbers (JACCN) and parcel areas (PARAREA). While the SQL expression simply lists (1) the names of the columns whose values are to be searched (SELECT clause), (2) what tables have to be accessed (FROM clause), and (3) the conditions and restrictions to be observed in the search (WHERE and AND clauses), Q-TRANS connects (1), (2), and (3) grammatically and logically: the terms in the SELECT clause (1) become the topic (grammatically the direct object) of the imperative verb ("find") and part of the information in (3) is expressed in a relative clause modifying (1).

The connecting element, the antecedent of the relative pronoun, is derived from the FROM clause (2) and represents the "real world" entity to which the tables and their column attributes apply. In this database, almost all tables and attributes relate to lots or parcels. This information is recorded for each table and column name in the Knuth translator, allowing Q-TRANS in this case to generate the prepositional phrase "for lots" modifying the preceding conjoined noun phrase and heading the relative clause that follows. The WHERE clause expresses a join condition which is suppressed in the translation. This is accomplished in the Knuth translator by assigning features to column names identifying them as keys of certain tables. If a WHERE (or AND) clause has column names on either side of an identity equation and if these names are compatible keys for joining, then the translation of the clause is suppressed.

The features that are currently used in the system pertain to (a) grammatical categories for producing the appropriate translations depending on grammatical function; (b) various counters for conjunction treatment; (c) data type information to determine prenominal versus postnominal modifications, singular versus plural usage; (d) code information for triggering translation of codes into English; (e) database related information regarding joins and table membership; and (f) the lexical translations of column names, SQL key words and other inserted function words.

In this particular example, the values for the column names are all in postnominal position. Given no further information other than the English equivalents for the column names, this structure will produce adequate translations. In this case, however, more knowledge is available and the insertion of the "sq. ft." for parcel area is triggered by a feature for that particular column name. The exclusive generation of postnominal qualifications in this example is due to the fact that we are dealing with descriptive attributes (zoning codes and street names) and a quantitative measure (parcel area) only. Whenever there are value specifications for attributes

that are countable, Q-TRANS provides prenominal descriptions. For example, the partial SQL sentence

AND B.JDWL > 25

would result in a translation like

..., more than 25 dwelling units...

If, in addition to the above, information is included concerning appropriate prepositions for linking attributes to entities, a shorter version with prepositional phrases instead of the relative clauses is produced by a prepositional version of Q-TRANS:

Find the account numbers and the parcel areas for lots on STEVENS ST, with a parcel area of greater than 5000 sq. ft., and in zone R5.

IV. PROBLEMS AND FUTURE WORK

While Q-TRANS generates paraphrases for a set of 1000 database queries that were produced by TQA, there are still a few cases which result in unnatural and sometimes misleading paraphrases. For example, harmless appearing questions such as:

How much vacant land in subplanning area 7.80 is not residentially zoned?

can be translated into complex SQL xpressions with query-level embeddings introduced by negated existential quantifiers. The reason for such constructions is the configuration of this particular database, where values for "residentially zoned" constitute a long list, which has to be filtered out in searching for the vacant land in the given subplanning area. The resulting query, containing a subquery with a negated existential quantifier, makes for a long and somewhat awkward abstract representation for the translation. A related problem lies in noun phrase conjunction, as in

In what wards do Smith and Jones both own parcels?

Since there exists no straightforward way in SQL to express conjunctions or groupings in the SELECT clause, TQA has to produce SQL expressions based on self-joins of a relation, listing the topics to be queried for each relation individually.

The difficulties in both examples here stem from the need for the logical rearrangement of the query structure without being too repetitive and without introducing ambiguities in the English paraphrase. Both of these problems will be resolved by extending the currently implemented attribute grammar such that it can compare units across non-adjacent phrases and take action accordingly, using grammatical mechanisms like deletion and pronominalization, as well as referencing techniques required for this particular task. These problems are in the process of being solved in a way designed to guarantee logically accurate translations of SQL, whether TQA-produced or user-produced.

The possible uses of Q-TRANS go beyond the one described in this paper. Two immediate applications are its use (1) for full sentence answerback, which is a straightforward extension of the existing system, since Q-TRANS has been designed in such a way that the output sentences can easily be modified from an imperative to a declarative sentence to provide a complete sentence containing the data retrieved; and (2) for SQL well-formedness checking, e.g. for accurate variable binding and data and function compatibility.

More generally, the approach that has been taken here to generate English (natural language) paraphrases is not restricted to SQL as the formal basis but is applicable to other formal query languages as well. Going a step further than that, any formal language that is analyzable by a context-free grammar can in principle be translated with the described mechanisms into English. One of the requirements and challenges in such an undertaking is the definition of the mapping relationship between the formal concepts and natural language.

REFERENCES

[1] Astrahan, M. M., M. W. Blasgen, D. D. Chamberlin, K. P. Eswaran, J. N. Gray, P.P. Griffiths, W. F. King, R. A. Lorie, J. McJones, J. W. Mehl, G. R. Putzolu, I. L. Traiger, B. W. Wade, V. Watson, "System R: Relational Approach to Database Management". ACM Transactions on Database Systems, Vol. 1, No. 21 June, 1976, pp. 97-137.

[2] Codd, E. F., et al., "Rendezvous Version 1: An Experimental English-language Query Formulation System for Casual Users of Relational Data Bases". IBM Research Report RJ2144(29407), IBM Research Laboratory, San Jose, CA, 1978.

[3] Damerau, F. J., "Operating Statistics for The Transformational Question Answering System". American Journal of Computational Linguistics, Vol. 7, No. 1, January-March 1981, pp. 30-42.

[4] Grishman, R., "Response Generation in Question-Answering Systems". Proc. of the 17th Ann. Mtg. of the ACL, LaJolla, CA, August 1979, pp. 99-102.

[5] Kaplan, S. J., "Cooperative Responses from a Natural Language Data Base Query System: Preliminary Report". Technical Report, Dept. of Computer and Information Science, Moore School, University of Pennsylvania, Philadelphia, PA, 1977.

[6] Knuth, D. E., "Semantics of Context-Free Languages". Mathematical Systems Theory, Vol. 2, No. 2, 1968 pp. 127-145.

[7] McKeown, K. M., "Paraphrasing Using Given and New Information in a Question-Answer System". ACL Proceedings, 1979, pp. 67-72.

[8] Petrick, S. R., "Semantic Interpretation in the Request System". In Computational and Mathematical Linguistics, Proceedings of the International Conference on Computational Linguistics, Pisa, 27/VIII-1/IX 1973, pp. 585-610.

[9] Petrick, S. R., "Field Testing the Transformational Question Answering (TQA) System". Proc. 19th Ann. Mtg. of the ACL, June 1981, pp. 35-36.

[10] Plath, W. J. "REQUEST: A Natural Language Question-Answering System". IBM Journal of Research and Development, Vol. 20, No. 4, July 1976, pp. 326-335.

[11] Waltz, D. L., "An English Language Question Answering System for a Large Relational Database". CACM, Vol. 21 No. 7, July 1978.