FIELD TESTING THE TRANSFORMATIONAL QUESTION ANSWERING (TQA) SYSTEM

S. R. Petrick
IBM T.J. Watson Research Center
PO Box 218 Yorktown Heights, New York 10598

The Transformational Question Answering (TQA) system was developed over a period of time beginning in the early part of the last decade and continuing to the present. Its syntactic component is a transformational grammar parser [1, 2, 3], and its semantic component is a Knuth attribute grammar [4, 5]. The combination of these components provides sufficient generality, convenience, and efficiency to implement a broad range of linguistic models; in addition to a wide spectrum of transformational grammars, Gazdar-type phrase structure grammar [6] and lexical functional grammar [7] systems appear to be cases in point, for example. The particular grammar which was, in fact, developed, however, was closest to those of the generative semantics variety of transformational grammar; both the underlying structures assigned to sentences and the transformations employed to effect that assignment traced their origins to the generative semantics model.

The system works by finding the underlying structures corresponding to English queries through the use of the transformational parsing facility. Those underlying structures are then translated to logical forms in a domain relational calculus by the Knuth attribute grammar component. Evaluation of logical forms with respect to a given data base completes the question-answering process. Our first logical form evaluator took the form of a toy implementation of a relational data base system in LISP. We soon replaced the low level tuple retrieval facilities of this implementation with the RSS (Relational Storage System) portion of the IBM System R [8]. This version of logical form evaluation was the one employed in the field testing to be described. In a more recent version of the system, however, it has been replaced by a translation of logical forms, first to equivalent logical forms in a set domain relational calculus and then to appropriate expressions in the SQL language, System R's high level query language.

The first data base to which the system was applied was one concerning business statistics such as the sales, earnings, number of employees, etc. of 60 large companies over a five-year period. This was a toy data base, to be sure, but it was useful to us in developing our system. A later data base contained the basic land identification records of about 10,000 parcels of land in a city near our research center. It was developed for use by members of the city planning department and (less frequently) other departments to answer questions concerning the information in that file. Our purpose in making the system available to those city employees was, of course, to provide access to a data base of real interest to a group of users and to field test our system by evaluating their use of it. Accordingly, the TQA system was tailored to the land use file application and installed at City Hall at the end of 1977. It remained there during 1978 and 1979, during which time it was used intermittently as the need arose for ad hoc query to supplement the report generation programs that were already available for the extraction of information.

Total usage of the system was less than we had expected would be the case when we made the decision to proceed with this application. This resulted from a number of factors, including a change in mission for the planning department, a reduction in the number of people in that department, a decision to rebuild the office space during the period of usage, and a degree of obsolescence of the data due to the length of time between updates (which were to have been supplied by the planning department). During 1978 a total of 788 queries were addressed to the system, and during 1979 the total was 210. Damerau [9] gives the distribution of these queries by month, and he also breaks them down by month into a number of different categories.

Damerau's report of the gross performance statistics for the year 1978, and a similar, as yet unpublished report of his for 1979, contain a wealth of data that I will not attempt to include in this brief note. Even though his reports contain a large quantity of statistical performance data, however, there are a lot of important observations which can only be made from a detailed analysis of the day-by-day transcript of system usage. An analysis of sequences of related questions is a case in point as is an analysis of the attempts of users to phrase new queries in response to failure of the system to process certain sentences. A paper in preparation by Plath is concerned with treating these and similar issues with the care and detail which they warrant. Time and space considerations limit my contribution in this note to just highlighting some of the major findings of Damerau and Plath.

Consider first a summary of the 1978 statistics:

Total Queries	788	
Termination Conditions:		%
Completed (Answer reached)	513	65.1
Aborted (System crash, etc.)	53	6.7
User Cancelled	21	2.7
Program Error	39	4.9
Parsing Failure	147	18.7
Unknown	15	1.9
Other Relevant Events:		
User Comment	96	12.2
Operator Message	45	5.7
User Message	11	1.4
Word not in Lexicon	119	15.1
Lexical Choice Resolved by User	119	15.1
"Nothing in Data Base" Answer	61	7.7

The percentage of successfully processed sentences is consistent with but slightly smaller than that of such other investigators as Woods [10], Ballard and Bierman [11], and Hershman et al [12]. Extreme care should be exercised in interpreting any such overall numbers, however, and even more care must be exercised in comparing numbers from different studies. Let me just mention a few considerations that must be kept in mind in interpreting the TQA results above.

First of all, our users' purposes varied tremendously from day to day and even from question to question. On one occasion, for example, a session might be devoted to a serious attempt to extract data needed for a federal grant proposal, and either the query complexity might be relatively limited so as to minimize the chance of error, or else the questions might be essentially repetitions of the same query, with minor variations to select different data. On another occasion, however, the session might be a demonstration, or a serious attempt to determine the limits of the system's understanding capability, or even a frivolous query to satisfy the user's curiosity as to the computer's response to a question outside its area of expertise. (One of our failures was the sentence, "Who killed Cock Robin?".)

Our users varied widely in terms of their familiarity with the contents of the data base. None knew anything about the internal organization of information (e.g. how the data was arranged into relations), but some had good knowledge of just what kind of data was stored, some had limited knowledge, and some had no knowledge and even false expectations as to what knowledge was included in the data base. In addition, they varied widely with respect to the amount of prior experience they had with the system. Initially we provided no formal training in the use of the system, but some users acquired significant knowledge of the system through its sustained use over a period of time. Something over half of the total usage was made by the individual from the planning department who was responsible for starting the system up and shutting it down each day. Usage was also made by other members of the planning department, by members of other departments, and by summer interns.

It should also be noted that the TQA system itself did not stay constant over the two-year period of testing. As problems were encountered, modifications were made to many components of the system. In particular, the lexicon, grammar, semantic interpretation rules (attribute grammar rules), and logical form evaluation functions all evolved over the period in question (continuously, but at a decreasing rate). The parser and the semantic interpreter changed little, if any. A rerun of all sentences, using the version of the grammar that existed at the conclusion of the field test program showed that 50 % of the sentences which previously failed were processed correctly. This is impressive when it is observed that a large percentage of the remaining 50 % constitute sentences which are either ungrammatical (sometimes sufficiently to preclude human comprehension) or else contain references to semantic concepts outside our universe of (land use) discourse.

On the whole, our users indicated they were satisfied with the performance of the system. In a conference with them at one point during the field test, they indicated they would prefer us to spend our time bringing more of their files on line (e.g., the zoning board of appeals file) rather than to spend more time

providing additional syntactic and associated semantic capability. Those instances where an unsuccessful query was followed up by attempts to rephrase the query so as to permit its processing showed few instances where success was not achieved within three attempts. This data is obscured somewhat by the fact that users called us on a few occasions to get advice as to how to reword a query. On other occasions the terminal message facility was invoked for the purpose of obtaining advice, and this left a record in our automatic logging facility. That facility preserved a record of all traffic between the user's terminal, the computer, and our own monitoring terminal (which was not always turned on or attended), and it included a time stamp for every line displayed on the users' terminal.

A word is in order on the real time performance of the system and on the amount of CPU time required. Damerau [9] includes a chart which shows how many queries required a given number of minutes of real time for complete processing. The total elapsed time for a query was typically around three minutes (58% of the sentences were processed in four minutes or less). Elapsed time depended primarily on machine load and user behavior at the terminal. The computer on which the system operated was an IBM System 370/168 with an attached processor, 8 megabytes of memory and extensive peripheral storage, operating under the VM/370 operating system. There were typically in excess of 200 users competing for resources on the system at the times when the TQA system was running during the 1978-1979 field tests. Besides queuing for the CPU and memory, this system developed queues for the IBM 3850 Mass Storage System, on which the TQA data base was stored.

Users had no complaints about real time response, but this may have been due to their procedure for handling ad hoc queries prior to the installation of the TQA system. That procedure called for ad hoc queries to be coded in RPG by members of the data processing department, and the turnaround time was a matter of days rather than minutes. It is likely that the real time performance of the system caused users sometimes to look up data about a specific parcel in a hard copy printout rather than giving it to the system. Queries were most often of the type requiring statistical processing of a set of parcels or of the type requiring a search for the parcel or parcels that satisfied given search criteria.

The CPU requirements of the system, broken down into a number of categories, are also plotted by Damerau [9]. The typical time to process a sentence was ten seconds, but sentences with large data base retrieval demands took up to a minute. System hardware improvements made subsequent to the 1978-1979 field tests have cut this processing time approximately in half. Throughout our development of the TQA system, considerations of speed have been secondary. We have identified many areas in which recoding should produce a dramatic increase in speed, but this has been assigned a lesser priority than basic enhancement of the system and the coverage of English provided through its transformational grammar.

Our experiment has shown that field testing of question answering systems provides certain information that is not otherwise available. The day to day usage of the system was different in many respects from usage that results from controlled, but inevitably somewhat artificial, experiments. We did not influence our users by the wording of problems posed to them because we gave them no problems; their requests for information were solely for their own purposes. Our sample queries that we initially exhibited to city employees to indicate the system was ready to be tested were invariably greeted with mirth, due to the improbability that anyone would want to know the information requested. (They asked for reassurance that the system would also answer "real" questions). We also obtained valuable information on such matters as how long users persist in rephrasing queries when they encounter difficulties of various kinds, how successful they are in correcting errors, and what new errors are likely to be made while correcting initial errors. I hope to discuss these and other matters in more detail in the oral version of this paper.

Valuable as our field tests are, they cannot provide certain information that must be obtained from controlled experiments. Accordingly, we hope to conduct a comparison of TQA with several formal query languages in the near future, using the latest enhanced version of the system and carefully controlling such factors as user training and problem statement. After teaching a course in data base management systems at Queens College and the Pratt Institute, and after running informal experiments there comparing students' relative success in using TQA, ALPHA, relational algebra, QBE, and SEQUEL, I am convinced that even for educated, programming-oriented users with a fair amount of experience in learning a formal query language, the TQA system offers significant advantages over formal query languages in retrieving data quickly and correctly. This remains to be proved (or disproved) by conducting appropriate formal experiments.

REFERENCES

[1] Plath, W. J., Transformational Grammar and Transformational Parsing in the Request System, IBM Research Report RC 4396, Thomas J. Watson Research Center, Yorktown Heights, N.Y., 1973.

[2] Plath, W. J., String Transformations in the REQUEST System, American Journal of Computational Linguistics, Microfiche 8, 1974.

[3] Petrick, S. R., Transformational Analysis, Natural Language Processing (R. Rustin, ed.), Algorithmics Press, 1973.

[4] Knuth, D. E., Semantics of Context-Free Languages, Mathematical Systems Theory , II, June 1968 2, pp. 127-145.

[5] 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.

[6] Gazdar, G. J. M., Phrase Structure Grammar, to appear in The Nature of Syntactic Representation , (eds. P. Jacobson and G. K. Pullum), 1979.

[7] Bresnan, J. W. and Kaplan, R. M., Lexical-Functional Grammar: A Formal System for Grammatical Representation, to appear in The Mental Representation of Grammatical Relations (J. W. Bresnan, ed.), Cambridge: MIT Press.

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

[9] Damerau, F. J., The Transformational Question Answering (TQA) System Operational Statistics - 1978, to appear in AJCL, June 1981.

[10] Woods, W. A., Transition Network Grammars, Natural Language Processing (R. Rustin, ed.), Algorithmics Press, 1973.

[11] Biermann, A. W. and Ballard, B. W., Toward Natural Language Computation, AJCL, Vol. 6, No. 2, April-June 1980, pp. 71-86.

[12] Hershman, R. L., Kelley, R. T., and Miller, H. C., User Performance with a Natural Language Query System for Command Control, NPRDC TR 79-7, Navy Personnel Research and Development Center, San Diego, Cal. 92152, January 1979.