

Experience with the Evaluation of Natural Language Question Answerers*

Harry Tennant
Advanced Automation Group
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Urbana, IL 61801

Research in natural language processing could be facilitated by thorough and critical evaluations of natural language systems. Two measurements, conceptual and linguistic completeness, are defined and discussed in this paper. Testing done on two natural language question answerers demonstrated that the conceptual coverage of such systems should be extended to better satisfy the needs and expectations of users.

None of the natural language papers presented at IJCAI77 reported on their programs' performance as language understanders. The reader has very little hope of thoroughly understanding the capabilities of the systems and techniques described.

The most critical dimensions of natural language performance are the range of concepts understood and the accommodation of diverse forms of expression. In this work on evaluation I have concentrated on these two aspects of performance and restricted the study to natural language question answering systems. In the paragraphs that follow, I will describe performance goals and measurement in more detail, then discuss the findings to which their use has led.

1.0 Performance Goals and Measurement

A natural language processor can be seen from two points of view: that of the designer and that of the user. The designer examines the domain of discourse of a particular application and maps out the range of concepts that are within it. The range of concepts is the CONCEPTUAL COVERAGE of the system. The designer must also accommodate the diversity with which the users will generate their statements and requests. The range of linguistic phenomena handled by the system is the LINGUISTIC COVERAGE of the system. The gauge of the success of the designers is how well the coverage has anticipated the needs of the users.

Users see a natural language question answering system and the database to which it interfaces through the perspective of their own needs and habits. They approach a database question answerer expecting it to include certain concepts, and forming utterances in their accustomed ways. The degree to which the concepts expected by a set of users can actually be found in the system's conceptual coverage is the CONCEPTUAL COMPLETENESS of the natural language processor, with respect to the set of users. Similarly, the degree to which the language of a set of users is appropriately analyzed by the system is the LINGUISTIC COMPLETENESS with respect to that set of users.

It is interesting to consider the current description techniques for natural language processors in terms of conceptual and linguistic coverage and completeness. When a paper presents twenty or so questions that are appropriately analyzed, the questions include concepts from the system's conceptual coverage, and their forms and features suggest elements of the system's linguistic coverage. Unfortunately, the conceptual and linguistic coverage of the system is not fully specified by what is found in the examples, and one cannot generalize from them. If no claim is made that the examples were in some sense typical of user inputs, nothing can be inferred about conceptual or linguistic completeness. If the paper goes on to explicitly mention handling features such as ellipsis or pronoun reference a comment is being made about the elements of linguistic coverage. Since this is not being related to the needs of a set of users, it says nothing about linguistic completeness.

*This work was supported by the Office of Naval Research under Contract N00014-75-C-0612.

2.0 Performance Testing

The ideas for testing conceptual and linguistic completeness were exercised on two natural language question answering systems, PLANES[9] and, to a lesser extent, the Automatic Advisor[7]. The performance of PLANES will be described in [8].

One tests the general competence of the natural language system by simply giving database problems to users and letting them attempt to solve them. Conceptual completeness is tested by giving database problems to subjects who are very familiar with the system, eg., the system designers. If a problem can be solved at all, the conceptual coverage must include the concepts indicated by the problem. Linguistic completeness is tested by giving users the problems that were solved in the conceptual completeness test. If a user utterance cannot be handled appropriately by the system, it is probably due to a limitation of linguistic coverage. It could, however, be due to a limitation of the conceptual coverage that was not discovered in the conceptual completeness test. For this reason, tests for conceptual completeness and linguistic completeness are not entirely independent.

3.0 Conceptual Completeness

Much work has been done and discussed in the literature on identifying the elements of linguistic coverage for improved linguistic completeness. The user testing has not led to the discovery of new elements of linguistic coverage.

The most interesting findings were in the area of conceptual completeness. Papers describing natural language processors frequently concentrate on linguistic coverage with the implicit assumption that the database determines the range of conceptual coverage. However, we found that the subjects thought in terms of the activities that the database described. The activities orientation is evidently the one that should be supported for casual users.

The activities orientation is helpful for identifying a broad class of erroneous presuppositions that may be embedded in user's queries. In the question "Which planes that flew on March 2, 1970 had maintenance on March 3?", one presupposition is that there were in fact some planes that flew on March 2. If the database lists the planes that flew on particular days, this would be a Kaplan-type presupposition [2]. However, if there was no

information on flights in the database, it would be a more general form of presupposition.

Another result relating to conceptual completeness was evidence against the common assumption that users' utterances are limited to database queries. Most natural language question answerers rely on the assumption that the user will ask a question, the system will query the database to answer it, then the user will ask another and so on. This is not the way conversations work. In testing PLANES and the Automatic Advisor, users frequently made utterances that were not intended to be interpreted as database queries. Various examples follow.

1. About the database:
 1. "What do you know?"
 2. "Is that everything you know about math 195?"
2. About vocabulary:
 1. "What is a buser?"
 2. "What does howmal mean?"
3. Context setting:
 1. "Now I am talking about the year 1970".
 2. "Year is 1970".
 3. "I am interested in a7's".
4. Reference to discourse objects:
 1. "What was the last plane I talked about?"
 2. "Now combine the two"
 3. "The other ones I just mentioned"
5. Verifying or summarizing results:
 1. "Then infe 210 and infe 211 must be taken at the same time".
 2. "So all I need is junior standing and math 195".
6. Multiple query utterances:
 1. "What parts failed and what were the parts removed or installed".
7. Multiple utterance queries:
 1. "How many aircraft flew more than 10 flights in 1973? in 1972? in 1970? List aircraft by number of flights for 1970."

Each of these examples was actually typed in by users using one of the natural language processors. Many more examples in each of these categories could be imagined. A few additional examples can be found in Malhotra's work [3] on a hypothetical management information system. These examples show clearly that question answerers cannot operate on a one-database-query-per-sentence basis.

A third result was discovered while testing the Automatic Advisor. It pertains to pragmatic relationships between elements of a database

domain.

One of the data domains of the Automatic Advisor is that of the topics that are covered by the engineering courses in the database. The topics covered in the lower level courses were general, those in the upper level courses were more specific. The Automatic Advisor included no pragmatic relationships among the elements of the topics domain. Consequently, when asked for the courses that dealt with computers, it returned one course, the only one which explicitly mentioned "computers" as a topic. It failed to return about ten higher level hardware and software courses and a host of others that made extensive use of computers.

The problem of misleading answers due to missing pragmatic relationships is not peculiar to the Automatic Advisor, but is common to "portable" natural language processors. Harris states that "[the portability of ROBOT] is largely possible because of the way ROBOT makes use of the database as an existing semantic structure" [1]. Indeed, ROBOT, the Automatic Advisor, and other systems are evidence that much semantic and pragmatic information does lie in the structure of the database, but the tests have illustrated the limitations of relying solely on the database for this information. This suggests that conceptual coverage should include the pragmatic relationships between data elements even though these relationships may not be expressed in the database.

4.0 Conclusion

There is a clear need for natural language processors to be evaluated. The measures and measurement techniques described in this paper suggest that meaningful testing can be performed and that its results can promote the engineering of better systems.

The findings discussed above suggest that in addition to the study of the elements of linguistic coverage, attention should be focused on conceptual coverage. Without extending conceptual coverage beyond the limits of the database contents, a natural language question answerer can do little more than a formal query language. One mainstay argument in favor of natural language processors over formal query languages is that the use of natural language processors can be learned quickly and recalled easily after a period of disuse. These claims have not been substantiated, however.

The LADDER system [6] was evaluated [4] as a query language, and users were able to use the system with some facility after "moderate training (approximately 1.5 hours) in LADDER syntax and vocabulary" [4] (they were selected to be familiar with the activities described by the database). The LADDER users were trained because, "LADDER is sufficiently demanding in its syntax and lexicon so that if no training were provided, the outcome could be predicted with certainty -- namely, the technology would be severely limited as a tool for accessing a Naval command and control database." But tests of formal query languages have shown similar performance after familiarization [5].

It seems clear that the role that natural language question answerers should play is in providing the user with more complete conceptual coverage than he can have using a formal query language. The user has gained little if he avoids the rigors of learning a formal query language only to be subjected to the rigors of learning the details of the structure of the database. If he must know the minutia of either, he will not be able to use the system effectively on an occasional basis.

References

- [1] Harris, L. R. "User Oriented Data Base Query With the ROBOT Natural Language Query System." International Journal of Man-Machine Studies. Vol. 9, 1977, pp. 697-713.
- [2] Kaplan, S. J. "On the Difference Between Natural language and High Level Query Languages." ACM78. Washington, D.C., 1978.
- [3] Malhotra, A. "Design Criteria for a Knowledge-Based English Language System for Management: An Experimental Analysis." Ph.D. Thesis, MIT, MAC TR-146, 1975.
- [4] Miller, H. G., R. L. Hershman, R. T. Kelly. "Performance of a Natural Language Query System in a Simulated Command Control Environment." Naval Electronics Systems Command, 1978.
- [5] Reisner, P., R. F. Boyce, D. D. Chamberlin. "Human Factors Evaluation of Two Data Base Query Languages--Square and Sequel." NCC, 1975, pp.447-452.
- [6] Sacerdoti, E. D. "Language Access to Distributed Data With Error Recovery." IJCAI77 1977, pp. 196-202.
- [7] Tennant, H. "The Automatic Advisor." Master's Thesis, University of Illinois at Chicago Circle. 1977.
- [8] Tennant, H. "The Performance of PLANES" (forthcoming).
- [9] Waltz, D. L., B. A. Goodman. "Writing a Natural Language Data Base System." IJCAI77, 1977, pp. 144-150.