What Makes Evaluation Hard?

Harry Tennant
PO Box 225621, M/S 371
Texas Instruments, Inc.
Dallas, Texas 75265

1.0 THE GOAL OF EVALUATION

Ideally, an evaluation technique should describe an algorithm that an evaluator could use that would result in a score or a vector of scores that depict the level of performance of the natural language system under test. The scores should mirror the subjective evaluation of the system that a qualified judge would make. The evaluation technique should yield consistent scores for multiple tests of one system, and the scores for several systems should serve as a means for comparison among systems. Unfortunately, there is no such evaluation technique for natural language understanding systems. In the following sections, I will attempt to highlight some of the difficulties

2.0 PERSPECTIVE OF THE EVALUATION

The first problem is to determine who the "qualified judge" is whose judgements are to be modeled by the evaluation. One view is that he be an expert in language understanding. As such, his primary interest would be in the linguistic and conceptual coverage of the system. He may attach the greatest weight to the coverage of constructions and concepts which he knows to be difficult to include in a computer program.

Another view of the judge is that he is a user of the system. His primary interest is in whether the system can understand him well enough to satisfy his needs. This judge will put greatest weight on the system's ability to handle his most critical linguistic and conceptual requirements: those used most frequently and those which occur infrequently but must be satisfied. This judge will also want to compare the natural language system to other technologies. Furthermore, he may attach strong weight to systems which can be learned quickly, or whose use may be easily remembered, or which takes time to learn but provides the user with considerable power once it is learned.

The characteristics of the judge are not an impediment to evaluation, but if the characteristics are not clearly understood, the meaning of the results will be confused.

3.0 TESTING WITH USERS

3.1 Who Are The Users?

It is surprising to think that natural language research has existed as long as it has and that the statement of the goals is still as vague as it is. In particular, little commitment is made on what kind of user a natural language understanding system is intended to serve. In particular, little is specified about what the users know about the domain and the language understanding system. The taxonomy below is presented as an example of user characteristics based on what the user knows about the domain and the system.

Classes of Users of database query systems

V Familiar with the database and its software
IV Familiar with the database and the interaction language
III Familiar with the contents of database
II Familiar with the domain of application
I Passing knowledge of the domain of application

Of course, as users gain experience with a system, they will continually attempt to adapt to its quirks. If the purpose of the evaluation is to demonstrate that the natural language understanding system is merely useable, adaptation resents no problem. However, if natural language is being used to allow the user to express himself in his accustomed manner, adaptation does become important. Again, the goals of natural language systems have been left vague. Are natural language systems to be 1) immediately useful, 2) easily learned 3) highly expressive or 4) readily remembered through periods of disuse? The evaluation should attempt to test for these goals specifically, and must control for factors such as adaptation.

What a user knows (either through instruction or experience) about the domain, the database and the interaction language have a significant effect on how he will express himself. Database query systems usually expect a certain level of use of domain or database specific jargon, and familiarity with constructions that are characteristic of the domain. A system may perform well for class IV users with queries like,

> 1) What are the NORMU for AAFs in 71 by month?

However, it may fare poorly for class I users with queries like,

> 2) I need to find the length of time that the attack planes could not be flown in 1971 because they were undergoing maintenance. Exclude all preventative maintenance, and give me totals for each plane for each month.

3.2 What Does Success Rate Mean?

A common method for generating data against which to test a system is to have users use it, then calculate how successful the system was at satisfying user needs. If the evaluation attempts to calculate the fraction of questions that the system understood, it is important to characterize how difficult the queries were to understand. For example, twelve queries of the form,

3) How many hours of down time did plane 3 have in January, 1971
4) How many hours of down time did plane 3 have in February, 1971
.
.
.

will help the success rate more than one query like,

5) How many hours of down time did plane 3 have in each month of 1971,

However, one query like 5 returns as much information as the other twelve. In testing PLANES [Tennant, 1981], the users whose questions were understood with the highest rates of success actually had less success at solving the problems they were trying to solve. They spent much of their time asking many easy, repetitive questions and so did not have time to attempt some of the problems. Other users who asked more compact questions had plenty of time to hammer away at the queries that the system had the greatest difficulty understanding.

Another difficulty with success rate measurement is the characteristics of the problems given to users compared to the kind of problems anticipated by the system. I once asked a set of users to write some problems for other users to attempt to solve using PLANES. The problem authors were familiar with the general domain of discourse of PLANES, but did not have any experience using it. The problems they devised were reasonable given the domain, but were largely beyond the scope of PLANES' conceptual coverage. Users had very low success rates when attempting to solve these problems. In contrast, problems that I had devised, fully aware of PLANES' areas of most complete coverage (and devised to be easy for PLANES), yielded much higher success rates. Small wonder. The point is that unless the match between the problems and a system's conceptual coverage can be characterised, success rates mean little.

4.0 TAXONOMY OF CAPABILITIES

Testing a natural language system for its performance with with users is an engineering approach. Another approach is to compare the elements that are known to be involved in understanding language against the capabilities of the system. This has been called "sharpshooting" by some of the implementers of natural language systems. An evaluator probes the system under test to find conditions under which it fails. To make this an organized approach, the evaluator should base his probes on a taxonomy of phenomena that are relevant to language understanding. A standard taxonomy could be developed for doing evaluations.

Our knowledge of language is incomplete at best. Any taxonomy is bound to generate disagreement. However, it seems that most of the disagreements describing language are not over what the phenomena of language are, but over how we might best understand and model those phenomena. The taxonomy will become quite large, but this is only representative of the fact that understanding language is a very complex process. The taxonomy approach faces the problem of complexity directly.

The taxonomy approach to evaluation forces examination of the broad range of issues of natural language processing. It provides a relatively objective means for assessing the full range of capabilities of a natural language understanding system. It also avoids the problems listed above inherent in evaluation through user testing. It does, however, have some unpleasant attributes. First, it does not provide an easy basis for comparison of systems. Ideally an evaluation would produce a metric to allow one to say "system A is better than system B". Appealing as it is, natural language understanding is probably too complex for a simple metric to be meaningful.

Second, the taxonomy approach does not provide a means for comparison of natural language understanding to other technologies. That comparison can be done rather well with user testing, however.

Third, the taxonomy approach ignores the relative importance of phenomena and the interaction between phenomena and domains of discourse. In response to this difficulty, an evaluation should include the analysis of a simulated natural language system. The simulated system would consist of a human interpreter who acts as an intermediary between users and the programs or data they are trying to use. Dialogs are recorded, then those dialogs are analyzed in light of the taxonomies of features. In this way, the capabilities of the system can be compared to the needs of the users. The relative importance of phenomena can be determined this way. Furthermore, users' language can be studied without them adapting to the system's limitations.

The taxonomy of phenomena mentioned above is intended to include both linguistic phenomena and concepts. The linguistic phenomena relate to how ideas may be understood. There is an extensive literature on this. The concepts are the ideas which must be understood. This is much more extensive, and much more domain specific. Work in knowledge representation is partially focused on learning what concepts need to be represented, then attempting to represent them. Consequently, there is a taxonomy of concepts implicit in the knowledge representation literature.

Reference

Tennant, Harry. Evaluation of Natural Language Processors. Ph.D. Thesis, University of Illinois, Urbana, Illiniois, 1981.