SUBJECT-BASED EVALUATION MEASURES FOR INTERACTIVE SPOKEN LANGUAGE SYSTEMS

Patti Price,[1] Lynette Hirschman,[2] Elizabeth Shriberg,[3] Elizabeth Wade[4]

[1]SRI International, 333 Ravenswood Ave., EJ 133, Menlo Park, CA 94306.
[2] MIT Laboratory for Computer Science, Cambridge, MA 02139
[3]University of California at Berkeley, Department of Psychology, Berkeley, CA94720
[4]Stanford University, Department of Psychology, Stanford, CA 94305

ABSTRACT

The DARPA Spoken Language effort has profited greatly from its emphasis on tasks and common evaluation metrics. Common, standardized evaluation procedures have helped the community to focus research effort, to measure progress, and to encourage communication among participating sites. The task and the evaluation metrics, however, must be consistent with the goals of the Spoken Language program, namely interactive problem solving. Our evaluation methods have evolved with the technology, moving from evaluation of read speech from a fixed corpus through evaluation of isolated canned sentences to evaluation of spontaneous speech in context in a canned corpus. A key component missed in current evaluations is the role of subject interaction with the system. Because of the great variability across subjects, however, it is necessary to use either a large number of subjects or a within-subject design. This paper proposes a within-subject design comparing the results of a software-sharing exercise carried out jointly by MIT and SRI.

1. INTRODUCTION

The use of a common task and a common set of evaluation metrics has been a cornerstone of DARPA-funded research in speech and spoken language systems. This approach allows researchers to evaluate and compare alternative techniques and to learn from each other's successes and failures. The choice of metrics for evaluation is a crucial component of the research program, since there will be strong pressure to make improvements with respect to the metric used. Therefore, we must select metrics carefully if they are to be relevant both to our research goals and to transition of the technology from the laboratory into applications.

The program goal of the Spoken Language Systems (SLS) effort is to support human-computer interactive problem solving. The DARPA SLS community has made significant progress toward this goal, and the development of appropriate evaluation metrics has played a key role in this effort. We have moved from evaluation of closed vocabulary, read speech (resource management) for speech recognition evaluation to open vocabulary for spontaneous speech (ATIS).

In June 1990, the first SLS dry run evaluated only transcribed spoken input for sentences that could be interpreted independent of context. At the DARPA workshop in February 1991, researchers reported on speech recognition, spoken language understanding, and natural language understanding results for context-independent sentences and also for pairs of context-setting + context-dependent sentences. At the present workshop, we witness another major step: we are evaluating systems on speech, spoken language and natural language for all evaluable utterances within entire dialogues, requiring that systems handle each sentence in its dialogue context, with no externally supplied context classification information.

2. EVALUATION METHODOLOGY: WHERE ARE WE?

The current measures have been and will continue to be important in measuring progress, but they do not assess the interactive component of the system, a component that will play a critical role in future systems deployed in real tasks. Indeed, some current metrics may penalize systems that attempt to be co-operative (for example, use of the weighted error, see below, and the maximal answer constraints). We propose a complementary evaluation paradigm that makes possible the evaluation of interactive systems. In this section we outline the current state of evaluation methodology and point out some shortcomings.

The current evaluation procedure is fully automated, using a canned corpus as input and a set of canonical database tuples as output reference answers. The evaluation measures the recognition and understanding components of a spoken language system, based on the number of correctly answered, incorrectly answered, and unanswered queries. These are then incorporated into a single number to produce a weighted error: percent "No-Answer" plus twice the percent "Incorrect" (this formulation is equivalent to 1 minus the "Score", where the "Score" is the percent "Correct" minus the percent "Incorrect)."

To date, evaluation has focused on only half of the user-to-system-to-user loop: these measures evaluate how well the system understands what the user says, but they ignore how felicitously the system answers the user. The use of a canned corpus effectively precludes any attempt by the system to direct the course of the conversation, including eliciting missing information from the subject, carrying on a clarification dialogue (for ambiguous or misunderstood input) or providing partial answers. For example, suppose the user says "I want to go to Dallas next Sunday". There are a number of ways for the system to respond:

1. Ask for a place of departure:

   SYSTEM: "Where are you leaving from?"

2. Inform the user that the information is incomplete:

   SYSTEM: "Please give both a place of departure and a destination."

3. Show all flights arriving in Dallas from anywhere:

   SYSTEM: "Here are the flights arriving in Dallas from anywhere."

4. Refuse to answer the question, with some diagnostic message:

   SYSTEM: "I'm sorry, there are too many flights to show; please give me more information."

5. Refuse to answer the question:

   SYSTEM: "Please rephrase your question."

Obviously some of these responses seem more helpful than others (although it is not immediately clear that there is a "best" response). In any case, what the user says next depends on which strategy the system adopts. For example, if the first strategy is chosen, where the system asks the user to specify a departure city, the user might well respond "Boston." This makes sense as an answer to System Response 1, but is not easily interpreted if the system uses System Response 2: "Please give both a place of departure and a destination." Maintaining coherence in dialogue depends on what the system says to the user during data collection. If we continue to use only canned dialogue in evaluation, then we can choose among ways to evaluate systems:

1. Canonical Response: All systems must provide the identical response to each situation;

2. Dialogue Resynchronization: Each system must be able to process the data collection system's response, and resynchronize its state based on what the user received as a response during data collection;

3. Discarding Interactive Dialogue: We throw out of the common evaluation all interactive (particularly mixed initiative) dialogue.

Alternative 1 was rejected as stifling experimentation in building interactive systems. Alternative 2 was described in [6], but some sites have felt that it was too burdensome to implement (multiparty dialogue tracking is not an immediate requirement for our current spoken language systems). We are left with Alternative 3, disallowing interactive dialogue in evaluation. This is cause for concern, since using spoken language systems for interactive problem solving is a shared goal [9]. It also leaves open the possibility that we inadvertently direct our research toward strategies that maximize our current measures but may not translate to any measurable gains when used in real-world applications.

The push toward a single-number, automated, evaluation score has encouraged system builders to adopt strategies that maximize this score. To arrive at a single number, it is necessary to combine right answers, wrong answers, and "no answers". The current formula makes a wrong answer twice as bad as saying "I don't understand." As indicated by Appelt at the October 1991 Spoken Language Interim Meeting, these strategies may not always be the most helpful to a user interacting with the system to accomplish a task. For example, in our current evaluation framework, the answer in the following interchange could be counted as wrong, because the information would be insufficiently restricted:

USER: Show me flights from Boston to Dallas leaving tomorrow on some kind of wide-body airplane

SYSTEM: You want to see flights from Boston to Dallas leaving tomorrow, but I didn't understand the rest of the sentence; here are tomorrow's flights from Boston to Dallas: (*shows the flights but doesn't restrict answer to wide-bodies*).

Similarly, the following interchange would typically be scored correct:

USER: Show me the flights to Boston arriving before 7 a.m.

SYSTEM: Here are the flights from anywhere to Boston arriving before 7 a.m.: (*shows all flights to Boston from anywhere arriving before 7 a.m.*).

Meanwhile the following interaction counts as "unevaluable" because the system takes the initiative, providing a (possibly irreproducible) context necessary to interpret the user's next input:

USER: Show me the flights to Boston arriving before 7 a.m.

SYSTEM: Where are you departing from?

USER: Dallas

SYSTEM: Here are the flights to Boston from Dallas arriving before 7 a.m.:

To move this technology toward realistic applications, we need to focus on building useful systems. This requires that we explore trade-offs between guessing, partial answers, reporting lack of understanding, and having a clarification dialogue with the user. We also need to explore different styles of interaction between system and user. In addition, most aspects of the system interface (particularly, the form of the output) are not being assessed at all using current metrics (e.g., display of information, presence or absence of spoken output, mixed initiative strategies). We need to develop complementary evaluation techniques that allow us to make progress and measure performance on interactive systems, rather than confining ourselves to a metric that may penalize cooperativeness. Further, we need a sanity check on our measures to reassure ourselves that gains we make according to the measures will translate to gains in application areas. The time is right for this next step, now that many sites have real-time spoken language systems.

3. METHODS

We have argued that interactive systems cannot be evaluated solely on canned input; live subjects are required. However, live subjects can introduce uncontrolled variability across users which can make interpretation of results difficult. To address this concern, we propose a within-subject design, in which each subject solves a scenario using each system to be compared, and the scenario order and system order are counterbalanced. However, the within-subject design requires that each subject have access to the systems to be compared, which means that the systems under test must all be running in one place at one time (or else that subjects must be shipped to the sites where the systems reside, which introduces a significant time delay). Given the goal of deployable software, we chose to ship the software rather than the users, but this raises many infrastructure issues, such as software portability and modularity, and use of common hardware and software.

Our original plan was to test across three systems: the MIT system, the SRI system, and a hybrid SRI-speech/MIT-NL system. SRI would compare the SRI and SRI-MIT hybrid systems; MIT would compare the MIT and SRI-MIT hybrids. The first stumbling block was the need to license each system at the other site; this took some time, but was eventually resolved. The next stumbling block was use of site-specific hardware and software. The SRI system used D/A hardware that was not available at MIT. Conversely, the MIT system required a Lucid Lisp license, which was not immediately available to the SRI group. Further, research software typically does not have the documentation, support, and portability needed for rapid and efficient exchange. Eventually, the experiment was pared down to comparing the SRI system and the SRI/MIT hybrid system at SRI. These infrastructure issues have added considerable overhead to the experiment.

The SRI SLS employs the DECIPHER^{tm} speech recognition system [4] serially connected to SRI's Template Matcher system [7,1]. The pruning threshold of the recognizer was tuned so that system response time was about 2.5 times utterance duration. This strategy had the side-effect of pruning out more hypotheses than in the comparable benchmark system, and a higher word error rate was observed as a consequence. The system accesses the relational version of the Official Airline Guide database (implemented in Prolog), formats the answer and displays it on the screen. The user interface for this system is described in [16]. This system, referred to as the SRI SLS, will be compared to the hybrid SRI/MIT SLS. The hybrid system employs the identical version of the DECIPHER recognizer, set at the same pruning threshold. All other aspects of the system differ. In the SRI/MIT hybrid system, the DECIPHER recognition output is connected to MIT's TINA [15] natural-language understanding system and then to MIT software for database access, response formatting, and display. Thus, the experiment proposed here compares SRI's natural language (NL) understanding and response generation with the same components from MIT. We made no attempt to separate the contribution of the NL components from those of the interface and display, since the point of this experiment was to debug the methodology; we simply cut the MIT system at the point of easiest separation. Below, we describe those factors that were held constant in the experiment and the measures to be used on the resulting data.

3.1. Subjects, Scenarios, Instructions

Data collection will proceed as described in Shriberg et al. 1992 [16] with the following exceptions: (1) updated versions of the SRI Template Matcher and recognizer will be used; (2) subjects will use a new data collection facility (the room is smaller and has no window but is acoustically similar to the room used previously); (3) the scenarios to be solved have unique solutions; (4) the debriefing questionnaire will be a merged version of the questions used on debriefing questionnaires at SRI and at MIT in separate experiments; and (5) each subject will solve two scenarios, one using the SRI SLS and one using the SRI/MIT hybrid SLS. Changes from our previous data collection efforts are irrelevant as all comparisons will be made within the experimental paradigm and conditions described here.

MIT designed and tested two scenarios that were selected for this experiment:

SCENARIO A. Find a flight from Philadelphia to Dallas that makes a stop in Atlanta. The flight should serve breakfast. Find out what type of aircraft is used on the flight to Dallas. Information requested: aircraft type.

SCENARIO B. Find a flight from Atlanta to Baltimore. The flight should be on a Boeing 757 and arrive around 7:00 p.m. Identify the flight (by number) and what meal is served

on the flight. Information requested: flight number, meal type.

We will counterbalance the two scenarios and the two systems by having one quarter of the subjects participate in each of four conditions:

1. Scenario A on SRI SLS, then Scenario B on SRI/MIT hybrid SLS
2. Scenario A on SRI/MIT hybrid SLS, then Scenario B on SRI SLS
3. Scenario B on SRI SLS, then Scenario A on SRI/MIT hybrid SLS and
4. Scenario B on SRI/MIT hybrid SLS, then Scenario A on SRI SLS).

A total of 12 subjects will be used, 3 in each of the above conditions. After subjects complete the two scenarios, one on each of the two systems, they will complete a debriefing questionnaire whose answers will be used in the data analysis.

3.2. Measures

In this initial experiment, we will examine several measures in an attempt to find those most appropriate for our goals. One measure for commercial applications is the number of units sold, or the number of dollars of profit. Most development efforts, however, cannot wait that long to measure success or progress. Further, to generalize to other conditions, we need to gain insight into why some systems might be better than others. We therefore chose to build on experiments described in [12] and to investigate the relations among several measures, including:

- User satisfaction. Subjects will be asked to assess their satisfaction with each system (using a scale of 1-5) with respect to the scenario solution they found, the speed of the system, their ability to get the information they wanted, the ease of learning to use the system, comparison with looking up information in a book, etc. There will also be some open-ended questions in the debriefing questionnaire to allow subjects to provide feedback in areas we may not have considered.
- Correctness of answer. Was the answer retrieved from the database correct? This measure involves examination of the response and assessment of correctness. As with the annotation procedures [10], some subjective judgment is involved, but these decisions can be made fairly reliably (see [12] for a discussion on interevaluator agreement using log file evaluation). A system with a higher percentage of correct answers may be viewed as "better." However, other factors may well be involved that correctness does not measure. A correlation of correctness with user satisfaction will be a stronger indication of the usefulness of this measure. Lack of correlation might reveal an interaction with other important factors.
- Time to complete task, as measured from the first push-to-talk until the user's last system action. Once task and subject are controlled, as in the current design, making this measurement becomes meaningful. A system which results in faster completion times may be preferred, although it is again important to assess the correlation of time to completion with user satisfaction.
- User waiting time, as measured between the end of the first query and the appearance of the response. Faster recognition has been shown to be more satisfying [16] and may correlate with overall user satisfaction.
- User response time, as measured between the appearance of the previous response and the push-to-talk for the next answer. This time may include the time the user needs to formulate a question suitable for the system to answer as well as the time it takes the user to assimilate the material displayed on the screen. In any case, user response time as defined here is distinct from waiting time, and is a readily measurable component of time to completion.
- Recognition word error rate for each scenario. Presumably higher accuracy will result in more user satisfaction, and these measures will also allow us to make comparison with benchmark systems operating at different error rates.
- Frequency and type of diagnostic error messages. Systems will typically display some kind of message when it has failed to understand the subject. These can be automatically logged and tabulated.

4. SUMMARY AND DISCUSSION

As pointed out by LTC Mettala in his remarks at this meeting, we need to know more than the results of our current benchmark evaluations. We need to know how changes in these benchmarks will change the suitability of a given technology for a given application. We need to know how our benchmarks correlate with user satisfaction and user efficiency. In a sense, we need to evaluate our evaluation measures.

At this writing, the MIT software has been transferred to SRI, and data collection is about to begin. We find that what began as an exercise in evaluation has become an exercise in software sharing. We do not want to deny the importance of software sharing and its role in strengthening portability. However, the difficulties involved (legal and other paperwork, acquisition of software and/or hardware, extensive interaction between the two sites) are costly enough that we believe we should also consider mechanisms that achieve our goals without requiring exchange of complete systems. Two such possibilities are described below.

Existing logfiles, including standard transcriptions, could be presented to a panel of evaluators for judgments of the appropriateness of individual answers and of the interaction as a whole. In a sense, then, the evaluators would simulate different users going through the same problem solving experience as the subject who generated the logfile. Cross-site variability of subjects used for this procedure could be somewhat controlled by specifying characteristics of these subjects (first time users, 2 hours of experience, daily computer user, etc.). This approach has several important advantages:

- It allows a much richer set of interactive strategies than our current metrics can assess, which can spur research in the direction of the stated program goals.
- It provides an opportunity to assess and improve the correlation of our current metrics with measures that are closer to the views of consumers of the technology, which should yield greater predictive power in matching a given technology to a given application.
- It provides a sanity check for our current evaluation measures, which could otherwise lead to improved scores but not necessarily to improved technology.
- It allows the same scenario-session to be experienced by more than one user, which addresses the subject-variability issue.
- It requires no exchange of software or hardware, and takes advantage of existing data structures currently required of all data collection sites, which means it is relatively inexpensive to implement.

The method however does NOT make use of a strictly within-subject design, i.e., the same subject does not interact with different systems (although the same evaluator would assess different systems). As a result, the logfile evaluation may require use of more subjects, or other techniques for addressing the issue of subject variability.

A live evaluation in which sites would bring their respective systems to a common location for assessment by a panel of evaluators could provide a means for a within-subject design. The solution of having a live test would have benefits similar to those outlined above for the logfile evaluation, but in addition subjects could assess the speed of system response, which the logfile proposal largely ignores. However, it would be more costly to transport the systems and the panel of evaluators than to ship logfiles (although most sites curretnly bring demonstration systems to meetings).

The logfile proposal could be modified to overcome its limited value in assessment of timing (at some additional expense) by the creation of a mechanism that would play back the logfiles using a standard display mechanism and based on the time stamps appearing in the logfiles. This would also open the possibility of having evaluators hear the speech of the subject, rather than just seeing transcriptions.

The costs involved for the use of such measures is negligible given the potential benefits. We propose these methods not as a replacement for the current measures, but rather as a complement to them and as a reality check on their function in promoting technological progress.

Acknowledgment. We gratefully acknowledge support for the work at SRI by DARPA through the Office of Naval Research Contract N00014-90-C-0085 (SRI), and Research Contract N00014-89-J-1332 (MIT). The Government has certain rights in this material. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the government funding agencies. We also gratefully acknowledge the efforts of David Goodine of MIT and of Steven Tepper at SRI in the software transfer and installation. This research was supported by DARPA

References

1. Appelt, D., Jackson, E., and R. Moore, "Integration of Two Complementary Approaches to Natural Language Understanding," *Proc. Fifth DARPA Speech and Natural Language Workshop*, M. Marcus (ed.), Morgan Kaufmann, 1992; this volume.

2. Bates, M., Boisen, S., and J. Makhoul, "Developing an Evaluation Methodology for Spoken Language Systems," pp. 102-108 in *Proc. Third Darpa Speech and Language Workshop*, Morgan Kaufmann, 1990.

3. Bly, B., P. Price, S. Tepper, E. Jackson, and V. Abrash, "Designing the Human Machine Interface in the ATIS Domain," pp. 136-140 *in Proc. Third Darpa Speech and Language Workshop*, Morgan Kaufmann, 1990.

4. Butzberger, J., H. Murveit, M. Weintraub, P. Price, and E. Shriberg, "Modeling Spontaneous Speech Effects in Large Vocabulary Speech Applications," *Proc. Fifth Darpa Speech and Language Workshop*, M. Marcus (ed.), Morgan Kaufmann, 1992; this volume.

5. Hemphill, C. T., J. J. Godfrey, and G. R. Doddington, "The ATIS Spoken Language System Pilot Corpus," pp. 96-101

in Proc. Third Darpa Speech and Language Workshop, Morgan Kaufmann, 1990.

6. Hirschman, L., D. A. Dahl, D. P. McKay, L. M. Norton, L., and M. C. Linebarger, "Beyond Class A: A Proposal for Automatic Evaluation of Discourse," pp. 109-113 *in Proc. Third Darpa Speech and Language Workshop*, Morgan Kaufmann, 1990.

7. Jackson, E., D. Appelt, J. Bear, R. Moore, A. Podlozny, "A Template Matcher for Robust NL Interpretation," pp. 190-194 in *Proc. Fourth DARPA Speech and Natural Language Workshop*, P. Price (ed.), Morgan Kaufmann, 1991.

8. Kowtko, J. C. and P. J. Price, "Data Collection and Analysis in the Air Travel Planning Domain," pp. 119-125 in *Proc. Second Darpa Speech and Language Workshop*, Morgan Kaufmann, 1989.

9. Makhoul, J., F. Jelinek, L. Rabiner, C. Weinstein, and V. Zue, pp. 463-479 in Proc. Second DARPA Speech and Natural Language Workshop, Morgan Kaufmann, 1989.

10. "Multi-Site Data Collection for a Spoken Language System," MADCOW, *Proc. Fifth Darpa Speech and Language Workshop*, M. Marcus (ed.), Morgan Kaufmann, 1992; this volume.

11. Polifroni, J., S. Seneff, V. W. Zue, and L. Hirschman, "ATIS Data Collection at MIT," *DARPA SLS Note 8*, Spoken Language Systems Group, MIT Laboratory for Computer Science, Cambridge, MA, November, 1990.

12. Polifroni, J., Hirschman, L., Seneff, S., and V. Zue, "Experiments in Evaluating Interactive Spoken Language Systems," *Proc. Fifth Darpa Speech and Language Workshop*, M. Marcus (ed.), Morgan Kaufmann, 1992; this volume.

13. Price P., "Evaluation of Spoken Language Systems: The ATIS Domain," pp. 91-95 *in Proc. Third Darpa Speech and Language Workshop*, Morgan Kaufmann, 1990.

14. Ramshaw, L.A. and S. Boisen, "An SLS Answer Comparator," *SLS Note 7*, BBN Systems and Technologies Corporation, Cambridge, MA, May 1990.

15. Seneff, S., Hirschman, L. and V. Zue, "Interactive Problem Solving and Dialogue in the ATIS Domain," pp. 354-359 *in Proc. Fourth Darpa Speech and Language Workshop*, P. Price (ed.), Morgan Kaufmann, 1991.

16. Shriberg, E., E. Wade, and P. Price, "Human-Machine Problem Solving Using Spoken Language Systems (SLS): Factors Affecting Performance and User Satisfaction," *Proc. Fifth Darpa Speech and Language Workshop*, M. Marcus (ed.), Morgan Kaufmann, 1992; this volume.