ISCA Archive
http://www.isca-speech.org/archive

2nd International Conference on
Spoken Language Processing (ICSLP 92)
Banff, Alberta, Canada
October 12-16, 1992

Evaluating Interactive Spoken Language Systems[1]

David Goodine, Lynette Hirschman, Joseph Polifroni, Stephanie Seneff, and Victor Zue[2]
Spoken Language Systems Group
Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

ABSTRACT

As the DARPA spoken language community moves towards developing useful systems for interactive problem solving, we must develop new evaluation metrics to assess whether these systems aid people in solving problems. In this paper, we report on experiments with two new metrics: task completion and logfile evaluation (where human evaluators judge query correctness). In one experiment, we used two variants of our data collection system (with a human transcriber) to compare an aggressive system using robust parsing to a more cautious "full-parse" system. In a second experiment, we compared a system using the human transcriber to a fully automated system using the speech recognizer. There were clear differences in task completion, time to task completion, and number of correct and incorrect answers. These experiments lead us to conclude that task completion and logfile evaluation are useful metrics for evaluating interactive systems.

10.21437/ICSLP.1992-58

INTRODUCTION

To date, common evaluation within the ATIS domain has been performed solely with the Common Answer Specification (CAS) protocol [4], which compares a system's performance to a canonical database answer [1]. The CAS protocol has the advantage that evaluation can be carried out automatically, once the principles for generating the reference answers have been established and a corpus has been annotated accordingly. Since direct comparison across systems can be performed relatively easily with this procedure, we have been able to achieve cross fertilization of research ideas, leading to rapid research progress.

However, the CAS evaluation is quite limited. It evaluates only at the sentence-by-sentence level; it makes no distinction between a partially correct answer and a completely misleading answer; it excludes evaluation of any kind of interaction – mixed initiative queries are specifically not evaluated, because they require local (system-dependent) context beyond the previous user queries. This means that the current evaluation does not allow us to evaluate interactive systems, including the use of mixed initiative, the quality of the response, and error detection and correction.

The focus of this paper is to introduce two new types of metric that allow us to evaluate interactive systems: task completion and logfile evaluation. Task completion looks at solving a problem (how correctly, how fast). Logfile evaluation uses human evaluators to asssess answer correctness in the context of an interactive dialogue. In order to assess the utility of these new metrics, we designed two experiments each comparing two versions of our system. We will first describe our experimental design, and then report on the results of our two experiments.

EXPERIMENTAL DESIGN

To carry out end-to-end evaluation, i.e., evaluation of overall task completion effectiveness, we must be able to determine whether the task has been completed and whether the answer is correct. In a pilot study, intended to help us establish an evaluation protocol for our experiments, we designed two scenarios which had well-defined answers [3]. Using our data collection system (with a human transcriber), we asked subjects to solve the scenario and report their answer. From the logfiles associated with the session scenario, we computed a number of objective measures, including the success of task completion, task completion time, and the total number of queries.

In addition to such quantifiable metrics, it should also be possible to obtain an assessment of the appropriateness of the responses to individual queries. If it can be shown that evaluators are consistent, then human evaluation becomes a viable approach. In the pilot study, we presented entire dialogues to several evaluators, displaying each query-answer pair in order, and requiring them to judge both the clarity of the query (i.e., clear, unclear, or unintelligible) and the correctness of the response (correct, partially correct, incorrect, or "system generated an error message"). Evaluators entered their answers on-line and the results were tabulated automatically. Our analyses, based on data from 7 evaluators, indicate that there was unanimous agreement among the evaluators for 82% of the query pairs, and an additional 10% had only one disagreement. Based on this consistency, we decided to make use of this human evaluation protocol for our experiments.

Each of our two experiments concerned a comparison of two distinct systems, and incorporated a within-subject design with each subject using both systems. In each session, the subject had to solve two pairs of scenarios, with each pair consisting of an "easy" scenario followed by a "hard" scenario. All scenarios had identifiable correct answers. In order to neutralize the effects of the individual scenarios and the order of scenarios, all subjects were presented with the same scenarios, in the same order. Half the subjects used system A for the first two scenarios, followed by System B, and the order was reversed for the other half. Subjects were given no prior training or warm-up exercises.

We modified our standard subject instructions slightly to inform the subject that s/he would be using two distinct systems. The subjects were drawn from the same pool as in our previous data collection efforts, mostly MIT students and staff. Each subject was given a $10 gift certificate for a local store. The subjects were not given any special incentive for getting correct answers, nor were they told that they would be timed. Each subject was asked to fill out a version of our debriefing questionnaire, slightly modified to include a specific question asking the subject which

[1] This research was supported by DARPA under Contract N00014-89-J-1332, monitored through the Office of Naval Research.

[2] Names listed in alphabetic order.

system s/he had preferred.

Writing the scenarios turned out to be nontrivial, and we had to iterate several times on the wording of the scenario descriptions in order to elicit the desired solution to the scenario. Even when we altered the instructions to remind the subjects of what kind of answer they should provide, subjects did not always read or follow the instructions carefully.

To determine the number of queries correctly answered by each system, two system developers independently examined each query/answer pair and judged the answer as correct, partially correct, incorrect, or unanswered, using the logfile evaluation program. They were in complete agreement more than 90% of the time. The cases of disagreement were examined to reach a compromise rating. This provided a quick and reasonably accurate way to assess whether subjects received the information they requested.

We made a number of measurements for each scenario/system as follows:

- Scenario completion time;
- Existence of a reported solution;
- Correctness of the reported solution;
- Number of queries;
- Number of queries answered;
- Number of queries judged to be answered correctly, partially, or incorrectly;
- DARPA score, defined as % Correct (of total) - % Incorrect (of total);
- User satisfaction from debriefing questionnaire.

ROBUST PARSING EXPERIMENT

In the first experiment, we collected data from fifteen subjects, with a wizard typing in subjects' queries, verbatim except for false starts. System A (the full-parse system) used a conservative approach that only answered when it was confident, whereas System B (the robust parse system [5]) used a more aggressive approach that was willing to make mistakes by answering much more often, based on partial understanding. In a previous test, the robust parser had performed significantly better than the other system in terms of the CAS metric[5]. These new metrics, for the most part, exhibited similar trends across all scenarios, as shown in Table 1.

Task Completion The first column of Table 1 shows that the subjects provided an answer in *all* the scenarios when the system was in robust mode, but for only 83% of the scenarios in non-robust mode. For the 5 cases in non-robust mode when users gave up, there was never an incorrectly answered query, but the number of unanswered queries was extremely high. From a problem-solving standpoint, we can tentatively conclude that a system that takes chances and answers more queries seems to be more successful than a more conservative one.

Finding the Correct Solution Our experimental paradigm allowed us to determine automatically, by processing the logfiles, whether the subject solved the scenario correctly, incorrectly, or not at all. A much larger percentage of the scenarios were correctly answered with the robust system than with the non-robust system (90% vs. 70%). Measured in terms of the percent of scenarios correctly solved, the robust system outperformed the non-robust system in all scenarios.

Task Completion Time The task completion time is summarized in the third column of Table 1. The results are somewhat inconclusive, due to a number of factors. Although we were interested in assessing how long it took to solve a scenario, we did not inform our subjects of this. In part, this was because we didn't want to add more stress to the situation. Because subjects were not encouraged to proceed quickly, it may be difficult to draw any conclusions from the results on time-to-completion. Another insidious factor was background network traffic and machine load, factors that would contribute to variations in time-to-completion which we did not control for in these experiments. Average number of queries to completion (next column) appears to be well correlated with task completion time.

Logfile Score The percentages of queries correctly answered, incorrectly answered, and unanswered, and the resulting DARPA score (i.e., % Correct - % Incorrect) are shown in the last four columns of Table 1. The overall ratio of correctly answered queries to those producing no answer was an order of magnitude higher for the robust parser (148:13) than for the non-robust parser (118:125). This was associated with an order-of-magnitude increase in the number of *incorrect* answers: 32 vs. 3 for the non-robust parser. However, the percentage of "no answer" queries seemed to be more critical in determining whether a subject succeeded with a scenario than the percentage of incorrect queries.

Debriefing Questionnaire Each subject received a debriefing questionnaire, which included a question asking for a comparison of the two systems used. Unfortunately, data were not obtained from the first five subjects. Of the ten subjects that responded, five preferred the robust system, one preferred the non-robust system, and the remaining ones expressed no preference.

Difficulty of Scenarios There was a wide variation across subjects in their ability to solve a given scenario, and in fact, subjects deviated substantially from our expectations. Several subjects did not read the instructions carefully and ignored or misinterpreted key restrictions in the scenario. Other subjects had a weak knowledge of air travel, which led to difficulties in solving the scenarios.

The full parser and robust parser showed different strengths and weaknesses in specific scenarios. For example, in Scenario 3, the full parser often could not parse the expression "Boeing 757", but the robust parser had no trouble. This accounts in part for the large "win" of the robust parser in this scenario. Conversely, in Scenario 4, the robust parser misinterpreted expressions of the type "about two hundred dollars", treating "about two" as a time expression. This led the conversation badly astray in these cases, and may acount for the fact that subjects took more time solving the scenario in robust mode. The lesson here is that different scenarios may find different holes in the systems under comparison, thus making the comparison extremely sensitive to the exact choice and wording of the scenarios.

Performance Comparison The robust parser performed better than the non-robust parser on all measures for all scenarios except in Scenario 4. In Scenario 4, the percentage of sessions resulting in a correct solution favored robust parsing (71% vs. 38%), but the robust parser had a longer time to completion and more queries to completion than the non-robust system, as well as a worse DARPA score (51% to 49%). The robust parser gave a greater percentage of correct answers (71% vs. 51%), but its incorrect answers were significant enough (22% to 0%) to reverse

Scenario Number	System	% of Scenarios w/Solution	Solution Correct	Completion Time(s)	Number of Queries	% of Queries Correct	% of Queries Incorrect	% of Queries No Answer	DARPA Score
1	Robust	100	100	215	4.4	94	0	6	94
1	Full	86	71	215	4.7	70	0	30	70
2	Robust	100	88	478	8.6	66	25	8	41
2	Full	86	86	483	10.6	39	4	56	35
3	Robust	100	100	199	4.4	82	15	3	68
3	Full	88	88	376	8.0	42	0	58	42
4	Robust	100	71	719	11.7	71	22	6	49
4	Full	75	38	643	9.8	51	0	49	51
All	Robust	100	90	399	7.2	75	18	6	57
All	Full	83	70	434	8.3	48	1	51	47

Table 1: Mean metrics for robust and full parse systems, shown by scenario

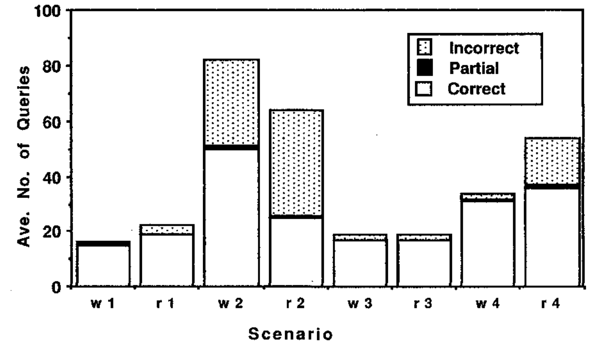

Figure 1: Breakdown of the individual queries for the four scenarios by number correct, partially correct, or incorrect, as judged by two system developers. [w = wizard-mode, r = recognizer-mode]

the outcome for the DARPA score. Thus the DARPA score seems to be correlated with time to completion, but percent of correct answers seems to be correlated with getting a correct solution.

In summary, our data show the following salient trends:

1. Subjects were always able to complete the scenario for the robust system.
2. Successful task completion distinguished the two systems: full parse system succeeded 70% of the time, compared with 90% for the robust system.
3. Percent of correctly answered queries followed the same trend as completion time and number of overall queries; these may provide a rough measure of task difficulty.
4. Users expressed a preference for the robust system.

RECOGNIZER EXPERIMENTS

We next performed another series of experiments using the same paradigm, but introduced a new variable: we compared the case where the system operated completely without the aid of a wizard (recognizer mode [6]) against the case where a wizard typed in verbatim (this time *including* false starts) what the user said. Each system used robust parsing and an aggressive answer strategy, always answering the query, but providing warnings to the effect that it was ignoring certain words because it didn't understand how to use them. This decision was made based on our experience in the previous experiment, where an aggressive strategy proved more successful than a more cautious strategy..

We again had each user solve an easy and a harder scenario on each system, (4 scenarios/user) but we substituted new versions of the hard scenarios, in which the system was set up in an interactive booking mode and the user was explicitly asked to book flights. In fact, for scenario 2 it was almost mandatory that they book flights in order to obtain the information needed to complete the task.

Due to a shortage of time, as well as a number of technical difficulties, we were only able to collect data for ten subjects. However, we believe this is sufficient to produce reliable trends. A plot of several parameters as a function of the eight cases (System A/B vs. Scenario 1 through 4) is shown in Table 2. We were encouraged by the bottom-line result – scenarios were solved correctly 82.5% of the time, overall. Again, we had some problems with users not following instructions; this accounted for three of the incorrect solutions, all of which occurred in wizard mode. Two users in Scenario 4 thought they should be *leaving* around 6 P.M. rather than arriving, and in one case in Scenario 1 a user neglected to ask about the aircraft. Since our point here was not to assess users' ability to follow instructions, we adjusted the data to call these three cases "correctly solved" for purposes of later comparisons, since they did correctly solve the scenario as they defined it.

The results of the logfile evaluation are shown graphically in Figure 1. A quick examination of this figure and Table 2 will reveal that Scenario 2 is out of line with the other scenarios on almost all measures. Scenario 2 required that the user return on the latest possible date, given the restrictions imposed by the cheapest fare. However, the information on the restrictions on the return date could not be easily obtained except by explicitly booking the forward leg. When the user tried to get the information directly, the system did not understand and kept asking for a return date, resulting, in some cases, in a grid-lock situation. As a consequence of this problem, Scenario 2 had many more queries incorrectly answered than did any of the other scenarios.

Figure 2 shows a plot comparing three distinct measures of success – (1) percentage of cases in which the scenario was correctly solved, (2) percentage of individual queries correctly answered, and (3) DARPA score, where the system is penalized for incorrectly answered queries [3]. These three parameters show very similar trends, but the DARPA score clearly overpenalizes systems – percent correct is a better measure to reflect "chances of solving the scenario." However, another factor of concern is the total time it takes to solve the scenario. Scenario 2 took about three times as long on average to solve as the other scenarios, and it also had by far the worst DARPA score. This raises an interesting question:

[3] The data for "scenario solved correctly" counted as correct the three cases where the user misunderstood the instructions.

Scenario Number	System	% of Scenarios w/Solution	Solution Correct	Completion Time(s)	Number of Queries	% of Queries Correct	% of Queries Incorrect	% of Queries No Answer	DARPA Score
1	Wizard	100	80	150	3.2	93.8	0	0	93.8
1	Recog.	100	100	284	4.4	86.4	13.7	0	72.7
2	Wizard	100	80	1012	16.4	61.0	37.8	0	23.2
2	Recog.	80	60	1100	12.8	39.1	59.4	0	-20.3
3	Wizard	100	100	195	3.8	89.5	10.6	0	78.9
3	Recog.	100	100	222	3.8	89.5	10.6	0	78.9
4	Wizard	100	60	353	6.8	91.2	5.9	0	85.3
4	Recog.	100	80	639	10.8	66.7	31.5	0	35.2
All	Wizard	95	80	427	7.5	83.9	13.6	0	57
All	Recog.	100	85	561	8.0	70.4	28.8	0	47

Table 2: Mean metrics for wizard and recognizer mode systems, shown by scenario

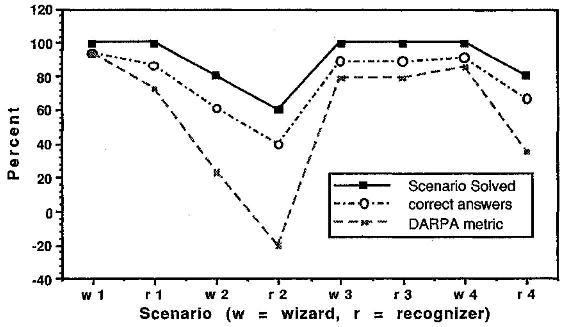

Figure 2: Plots of percentage of scenarios correctly solved, percentage of queries correctly answered, and the Darpa Metric score, by scenario.

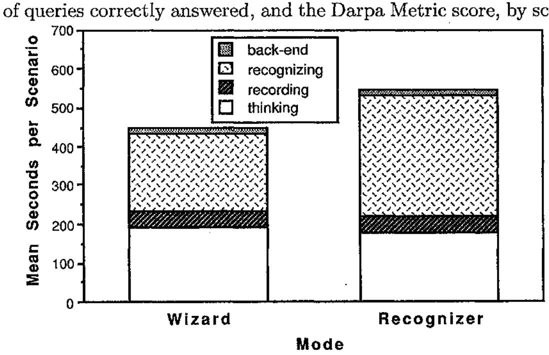

Figure 3: Breakdown of timing information for the wizard-mode vs. recognizer-mode systems.

does it take the user more time to solve the scenario if the system provides incorrect answers to difficult queries than it does if the system refuses to answer such queries. To answer this, a further experiment is needed, in which the only difference between System A and System B is the degree to which the system refuses to answer when it suspects potential misunderstandings.

Figure 3 shows a breakdown of timing information for the recognizer mode versus the wizard mode, pooled over all four scenarios. We divided the time spent into four distinct stages:

1. Thinking: time between when the system displays the table for the previous answer and the user starts recording.
2. Recording: total duration of the user's queries.
3. Recognizing: time the system spent recognizing the sentence [recognizer mode] or time the wizard spent typing the sentence [wizard mode].
4. Back-end: time the system spent retrieving the data and saving out the logfile.

We were surprised that the only significant difference between wizard-mode and recognizer-mode was in the time spent "recognizing." Aside from this obvious difference in processing time, users were able to solve the scenarios equally fast in spite of the errorful recognition. This was an unexpected yet encouraging result, because it indicates that perhaps recognizer errors are not as disruptive of the dialogue as one might expect.

CONCLUSIONS

The results of these experiments are very encouraging. We have shown that it is possible to define quantitative metrics to evaluate the performance of interactive problem-solving systems that are able to distinguish between different systems. There was good correspondence between how effective the system was in helping the user arrive at a correct answer for a given task and metrics such as task completion time, number of queries, and percent of correctly answered queries (based on logfile evaluation). In addition, we have shown that it is feasible to ask human evaluators to judge the quality of system responses to individual queries. Our metrics also indicate that system behavior may not be uniform over a range of scenarios, and may depend on certain peculiarities of a particular scenario. Finally, these experiments demonstrate that naive subjects can successfully use a fully automated spoken language system to solve problems in a limited domain.

REFERENCES

[1] Bates, M., Boisen, S., and Makhoul, J., "Developing an Evaluation Methodology for Spoken Language Systems," *Proc. DARPA Speech and Natural Language Workshop*, pp. 102-108, June, 1990.

[2] Hirschman, L. et al, "Multi-Site Data Collection for a Spoken Language Corpus," *These Proceedings.*

[3] Polifroni, J., Hirschman, L., Seneff, S., and Zue, V., "Experiments on Evaluating Interactive Spoken Language Systems," *Proc. DARPA Speech and Natural Language Workshop*, February 1992.

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

[5] Seneff, S., "A Relaxation Method for Understanding Spontaneous Speech Utterances," *Proc. DARPA Speech and Natural Language Workshop*, February 1992.

[6] Zue, V., Glass, J., Goddeau, D., Goodine, D., Hirschman, L., Philips, M., Polifroni, J., Seneff, S., "The MIT ATIS System: February 1992 Progress Report," *Proc. DARPA Speech and Natural Language Workshop*, February 1992.