Evaluating Response Strategies in a Web-Based Spoken Dialogue Agent

Diane J. Litman
AT&T Labs - Research
180 Park Avenue
Florham Park, NJ 07932 USA
diane@research.att.com

Shimei Pan
Computer Science Department
Columbia University
New York, NY 10027 USA
pan@cs.columbia.edu

Marilyn A. Walker
AT&T Labs - Research
180 Park Avenue
Florham Park, NJ 07932 USA
walker@research.att.com

Abstract

While the notion of a cooperative response has been the focus of considerable research in natural language dialogue systems, there has been little empirical work demonstrating how such responses lead to more efficient, natural, or successful dialogues. This paper presents an experimental evaluation of two alternative response strategies in TOOT, a spoken dialogue agent that allows users to access train schedules stored on the web via a telephone conversation. We compare the performance of two versions of TOOT (literal and cooperative), by having users carry out a set of tasks with each version. By using hypothesis testing methods, we show that a combination of response strategy, application task, and task/strategy interactions account for various types of performance differences. By using the PARADISE evaluation framework to estimate an overall performance function, we identify interdependencies that exist between speech recognition and response strategy. Our results elaborate the conditions under which TOOT's cooperative rather than literal strategy contributes to greater performance.

1 Introduction

The notion of a *cooperative response* has been the focus of considerable research in natural language and spoken dialogue systems (Allen and Perrault, 1980; Mays, 1980; Kaplan, 1981; Joshi et al., 1984; McCoy, 1989; Pao and Wilpon, 1992; Moore, 1994; Seneff et al., 1995; Goddeau et al., 1996; Pieraccini et al., 1997). However, despite the existence of many algorithms for generating cooperative responses, there has been little empirical work addressing the evaluation of such algorithms in the context of real-time natural language dialogue systems with human users. Thus it is unclear under what conditions cooperative responses result in more efficient or efficacious dialogues.

This paper presents an empirical evaluation of two alternative algorithms for responding to *database queries* in TOOT, a spoken dialogue agent for accessing online train schedules via a telephone conversation. We conduct an experiment in which 12 users carry out 4 tasks of varying difficulty with one of two versions of TOOT (*literal* and *cooperative* TOOT), resulting in a corpus of 48 dialogues. The values for a wide range of evaluation measures are then extracted from this corpus. We analyze our data using both traditional hypothesis testing methods and the PARADISE (Walker et al., 1997; Walker et al., 1998) methodology for estimating a performance function. Hypothesis testing shows that while differences among some evaluation measures depend on the response strategy (literal or cooperative), other differences are a function of application task and task/strategy interactions. A PARADISE assessment of the contribution of each evaluation measure to overall performance shows that strategy-dependent dialogue phenomena as well as phenomena associated with speech recognition significantly predict performance. Our results identify the conditions under which TOOT's cooperative response strategy leads to greater agent performance.

2 TOOT

TOOT allows users to access online AMTRAK train schedules via a telephone dialogue, as in Figure 1.[1] (All examples are from the experiment in Section 3.) We have built two versions of TOOT: *literal* TOOT (LT) and *cooperative* TOOT (CT). LT and CT have equivalent functionality, but use different response strategies to present tabular results of web queries in a displayless environment.[2] LT and CT incorporate many of the types of database responses in the

[1] Our domain was chosen to afford future comparison with similar systems, e.g., (Danieli and Gerbino, 1995).

[2] Initiative (Walker et al., 1998) and confirmation (Danieli and Gerbino, 1995) strategies are equivalent in LT and CT, and were chosen to reduce speech recognition errors.

T: Hi, this is the Amtrak train schedule system. Which city do you want to go to?
U: Philadelphia.
T: What city do you want to go to Philadelphia from?
U: New York City.
T: On what day of the week do you want to go to Philadelphia from New York City?
U: On the weekend.
T: What time do you want to leave on the weekend?
U: 4 o'clock pm.
T: Do you want me to find the trains going to Philadelphia from New York City on the weekend around 4 o'clock pm now?
U: Yes.
T: I am going to query the Amtrak website for you. This may take some time if the internet traffic is heavy. Please wait patiently. [30 second pause] *There are 7 trains available around 4 o'clock pm. Here are the first 3 trains. The 1st train leaves at 3:00 pm on Saturday and Sunday, and it takes 1 hour 12 mins. The 2nd train leaves at 3:20 pm every day, and it takes 1 hour 22 mins. The 3rd train leaves at 4:00 pm on Sunday, and it takes 1 hour 12 mins. Do you want to hear the rest of the trains?*
U: No.
T: Do you want to continue and find a new set of trains?
U: No.
T: Thank you for using the Amtrak schedule system. See you next time.

Figure 1: Example dialogue with (literal) TOOT.

literature into relatively literal and cooperative response strategies. (More sophisticated cooperative strategies could be imagined.) When there is *too much information* to present in a single utterance, LT groups the information into units of 3 trains, then presents each unit, as in the italicized portion of Figure 1. In contrast, CT summarizes the range of trains available, then tells the user to either list the trains or further constrain the query. In CT, the italicized portion of Figure 1 would be replaced with the following response:

(1) *There are 7 trains available around 4 o'clock pm. Here is the earliest train we have. The first train leaves at 3:00 pm on Saturday and Sunday, and it takes 1 hour 12 mins. Here is the latest train we have. The seventh train leaves at 5:00 pm on Saturday, and it takes 1 hour 12 mins. Please say "list" to hear trains 3 at a time, or say "add constraint" to constrain your departure time or travel day, or say "continue" if my answer was sufficient, or say "repeat" to hear this message again.*

LT's response incrementally presents the set of trains that match the query, until the user tells LT to stop. Enumerating large lists, even incrementally, can lead to information overload. CT's response is more cooperative because it better respects the resource limitations of the listener. CT presents a subset of the matching trains using a summary response (Pao and Wilpon, 1992), followed by an option to reduce the information to be retrieved (Pieraccini et al., 1997; Goddeau et al., 1996; Seneff et al., 1995; Pao and Wilpon, 1992).

If there is *no information* that matches a query, LT reports only the lack of an answer to the query, as in the following dialogue excerpt:

(2) *There are no trains going to Chicago from Philadelphia on Sunday around 10:30 am. Do you want to continue and find a new set of trains?*

CT automatically relaxes the user's time constraint and allows the user to perform other relaxations:

(3) *There are no trains going to Chicago from Philadelphia on Sunday around 10:30 am. The closest earlier train leaves at 9:28 am every day, and it takes 1 day 3 hours 36 mins. The closest later train leaves at 11:45 am on Saturday and Sunday, and it takes 22 hours 5 mins. Please say "relax" to change your departure time or travel day, or say "continue" if my answer was sufficient, or say "repeat" to hear this message again.*

CT's response is more cooperative since identifying the source of a query failure can help block incorrect user inferences (Pieraccini et al., 1997; Pao and Wilpon, 1992; Joshi et al., 1984; Kaplan, 1981; Mays, 1980). LT's response could lead the user to believe that there are no trains on Sunday.

When there are 1-3 trains that match a query, both LT and CT list the trains:

(4) *There are 2 trains available around 6 pm. The first train leaves at 6:05 pm every day, and it takes 5 hours 10 mins. The second train leaves at 6:30 pm every day, and it takes 2 days 11 hours 30 mins. Do you want to continue and find a new set of trains?*

TOOT is implemented using a platform for spoken dialogue agents (Kamm et al., 1997) that combines automatic speech recognition (ASR), text-to-speech (TTS), a phone interface, and modules for specifying a dialogue manager and application functions. ASR in our platform supports *barge-in*, an advanced functionality which allows users to interrupt an agent when it is speaking.

The dialogue manager uses a finite state machine to implement dialogue strategies. Each state specifies 1) an initial prompt (or response) which the agent says upon entering the state (such prompts often elicit parameter values); 2) a help prompt which the agent says if the user says *help*; 3) rejection prompts which the agent says if the confidence level of ASR is too low (rejection prompts typically ask the user to repeat or paraphrase their utterance); and 4) timeout prompts which the agent says if the user doesn't say anything within a specified time frame (timeout prompts are often suggestions about what to say). A context-free grammar specifies what ASR can recognize in each state. Transitions between states are driven by semantic interpretation.

TOOT's application functions access and process information on AMTRAK's web site. Given a set of constraints, the functions return a table listing all matching trains in a specified temporal interval, or within an hour of a specified timepoint. This table is converted to a natural language response which can be realized by TTS through the use of templates for either the LT or the CT response type; values in the table instantiate template variables.

3 Experimental Design

The experimental instructions were given on a web page, which consisted of a description of TOOT's functionality, hints for talking to TOOT, and links to 4 task pages. Each task page contained a task scenario, the hints, instructions for calling TOOT, and a web survey designed to ascertain the depart and travel times obtained by the user and to measure user perceptions of task success and agent usability. Users were 12 researchers not involved with the design or implementation of TOOT; 6 users were randomly assigned to LT and 6 to CT. Users read the instructions in their office and then called TOOT from their phone. Our experiment yielded a corpus of 48 dialogues (1344 total turns; 214 minutes of speech).

Users were provided with task scenarios for two reasons. First, our hypothesis was that performance depended not only on response strategy, but also on task difficulty. To include the task as a factor in our experiment, we needed to ensure that users executed the same tasks and that they varied in difficulty.

Figure 2 shows the task scenarios used in our experiment. Our hypotheses about agent performance are summarized in Table 1. We predicted that optimal performance would occur whenever the correct task solution was included in TOOT's initial response to a web query (i.e., when the task was easy).

Task 1 (Exact-Match): Try to find a train going to **Boston** from **New York City** on **Saturday** at **6:00 pm**. If you cannot find an exact match, find the one with the **closest** departure time. Write down the **exact departure time** of the train you found as well as the **total travel time**.

Task 2 (No-Match-1): Try to find a train going to **Chicago** from **Philadelphia** on **Sunday** at **10:30 am**. If you cannot find an exact match, find the one with the **closest** departure time. Write down the **exact departure time** of the train you found as well as the **total travel time**.

Task 3 (No-Match-2): Try to find a train going to **Boston** from **Washington D.C.** on **Thursday** at **3:30 pm**. If you cannot find an exact match, find the one **between 12:00 pm and 5:00 pm** that has the **shortest** travel time. Write down the **exact departure time** of the train you found as well as the **total travel time**.

Task 4 (Too-Much-Info/Early-Answer): Try to find a train going to **Philadelphia** from **New York City** on **the weekend** at **4:00 pm**. If you cannot find an exact match, find the one with the **closest departure time**. Please write down the **exact departure time** of the train you found as well as the **total travel time**. ("weekend" means the train departure date includes either Saturday or Sunday)

Figure 2: Task scenarios.

Task 1 (dialogue fragment (4) above) produced a query that resulted in 2 matching trains, one of which was the train requested in the scenario. Since the response strategies of LT and CT were identical under this condition, we predicted identical LT and CT performance, as shown in Table 1.[3]

Tasks 2 (dialogue fragments (2) and (3)) and 3 led to queries that yielded no matching trains. In Task 2 users were told to find the closest train. Since only CT included this extra information in its response, we predicted that it would perform better than LT.

In Task 3 users were told to find the shortest train within a new departure interval. Since neither LT nor CT provided this information initially, we hypothesized comparable LT and CT performance. However, since CT allowed users to change just their departure time while LT required users to construct a whole new query, we also thought it possible that CT might perform slightly better than LT.

Task 4 (Figure 1 and dialogue fragment (1)) led to

[3]Since Task 1 was the easiest, it was always performed first. The order of the remaining tasks was randomized across users.

Task	LT Strategy	CT Strategy	Hypothesis
Exact-Match	Say it	Say it	LT equal to CT
No-Match-1	Say No Match	Relax Time Constraint	LT worse than CT
No-Match-2	Say No Match	Relax Time Constraint	LT equal to or worse than CT
Too-Much-Info/Early-Answer	List 3 then more?	Summarize; Give Options	LT better than CT

Table 1: Hypothesized performance of literal TOOT (LT) versus cooperative TOOT (CT).

a query where the 3rd of 7 matching trains was the desired answer. Since only LT included this train in its initial response (by luck, due to the train's position in the list of matches), we predicted that LT would perform better than CT. Note that this prediction is highly dependent on the database. If the desired train had been last in the list, we would have predicted that CT would perform better than LT.

attribute	value
arrival-city	Philadelphia
depart-city	New York City
depart-day	weekend
depart-range	4:00 pm
exact-depart-time	4:00 pm
total-travel-time	1 hour 12 mins

Table 2: Scenario key, Task 4.

A second reason for having task scenarios was that it allowed us to objectively determine whether users achieved their tasks. Following PARADISE (Walker et al., 1997), we defined a "key" for each scenario using an attribute value matrix (AVM) task representation, as in Table 2. The key indicates the attribute values that must be exchanged between the agent and user by the end of the dialogue. If the task is successfully completed in a scenario execution (as in Figure 1), the AVM representing the dialogue is identical to the key.

4 Measuring Aspects of Performance

Once the experiment was completed, values for a range of evaluation measures were extracted from the resulting data (dialogue recordings, system logs, and web survey responses). Following PARADISE, we organize our measures along four performance dimensions, as shown in Figure 3.

To measure *task success*, we compared the scenario key and scenario execution AVMs for each dialogue, using the **Kappa** statistic (Walker et al., 1997). For the scenario execution AVM, the values for arrival-city, depart-city, depart-day, and depart-range were extracted from system logs of ASR results. The exact-depart-time and total-travel-time were extracted from the web survey. To measure users' *perceptions* of task success, the survey also asked users whether they had successfully **Completed** the task.

- **Task Success:** Kappa, Completed
- **Dialogue Quality:** Help Requests, ASR Rejections, Timeouts, Mean Recognition, Barge Ins
- **Dialogue Efficiency:** System Turns, User Turns, Elapsed Time
- **User Satisfaction:** User Satisfaction (based on TTS Performance, ASR Performance, Task Ease, Interaction Pace, User Expertise, System Response, Expected Behavior, Future Use)

Figure 3: Measures used to evaluate TOOT.

To measure *dialogue quality* or naturalness, we logged the dialogue manager's behavior on entering and exiting each state in the finite state machine (recall Section 2). We then extracted the number of prompts per dialogue due to **Help Requests**, **ASR Rejections**, and **Timeouts**. Obtaining the values for other quality measures required manual analysis. We listened to the recordings and compared them to the logged ASR results, to calculate concept accuracy (intuitively, semantic interpretation accuracy) for each utterance. This was then used, in combination with ASR rejections, to compute a **Mean Recognition** score per dialogue. We also listened to the recordings to determine how many times the user interrupted the agent (**Barge Ins**).

To measure *dialogue efficiency*, the number of **System Turns** and **User Turns** were extracted from the dialogue manager log, and the total **Elapsed Time** was determined from the recording.

To measure *user satisfaction*[4], users responded to the web survey in Figure 4, which assessed their subjective evaluation of the agent's performance. Each question was designed to measure a partic-

[4]Questionnaire-based user satisfaction ratings (Shriberg et al., 1992; Polifroni et al., 1992) have been frequently used in the literature as an external indicator of agent usability.

- Was the system easy to understand in this conversation? (**TTS Performance**)
- In this conversation, did the system understand what you said? (**ASR Performance**)
- In this conversation, was it easy to find the schedule you wanted? (**Task Ease**)
- Was the pace of interaction with the system appropriate in this conversation? (**Interaction Pace**)
- In this conversation, did you know what you could say at each point of the dialogue? (**User Expertise**)
- How often was the system sluggish and slow to reply to you in this conversation? (**System Response**)
- Did the system work the way you expected it to in this conversation? (**Expected Behavior**)
- From your current experience with using our system, do you think you'd use this regularly to access train schedules when you are away from your desk? (**Future Use**)

Figure 4: User satisfaction survey and associated evaluation measures.

ular factor, e.g., **System Response**. Responses ranged over n pre-defined values (e.g., *almost never, rarely, sometimes, often, almost always*), which were mapped to an integer in $1 \ldots n$. Cumulative **User Satisfaction** was computed by summing each question's score.

5 Strategy and Task Differences

To test the hypotheses in Table 1 we use analysis of variance (ANOVA) (Cohen, 1995) to determine whether the values of any of the evaluation measures in Figure 3 significantly differ as a function of response strategy and task scenario.

First, for each task scenario (4 sets of 12 dialogues, 6 per agent and 1 per user), we perform an ANOVA for each evaluation measure as a function of *response strategy*. For Task 1, there are no significant differences between the 6 LT and 6 CT dialogues for any evaluation measure, which is consistent with Table 1. For Task 2, mean **Completed** (perceived task success rate) is 50% for LT and 100% for CT ($p < .05$). In addition, the average number of **Help Requests** per LT dialogue is 0, while for CT the average is 2.2 ($p < .05$). Thus, for Task 2, CT has a better perceived task success rate than LT, despite the fact that users needed more help to use CT. Only the perceived task success difference is consistent with the Task 2 prediction in Table 1.[5] For Task 3, there are no significant differences between LT and CT, which again matches our predictions. Finally, for Task 4, mean **Kappa** (actual task success rate) is 100% for LT but only 65% for CT ($p < .01$).[6] Like Task 2, this result suggests that some type of task success measure is an important predictor of agent performance. Surprisingly, we found that LT and CT did *not* differ with respect to any efficiency measure, in any task.[7]

Next, we combine all of our data (48 dialogues), and perform a two-way ANOVA for each evaluation measure as a function of strategy and task. An *interaction between response strategy and task scenario* is significant for **Future Use** ($p < .03$). For task 1, the likelihood of **Future Use** is the same for LT and CT; for task 2, the likelihood is higher for CT; for tasks 3 and 4, the likelihood is higher for LT. Thus, the results for tasks 1, 2, and 4, but not for Task 3, are consistent with the predictions in Table 1. However, Task 3 was the most difficult task (see below), and sometimes led to unexpected user behavior with both agents. A strategy/task interaction is also significant for **Help Requests** ($p < .02$). For tasks 1 and 3, the number of requests is higher for LT; for tasks 2 and 4, the number is higher for CT.

No evaluation measures significantly differ as a function of *response strategy*, which is consistent with Table 1. Since the task scenarios were constructed to yield comparable performance in Tasks 1 and 3, better CT performance in Task 2, and better LT performance in Task 4, we expected that *overall*, LT and CT performance would be comparable.

In contrast, many measures (**User Satisfaction**, **Elapsed Time**, **System Turns**, **User Turns**, **ASR Performance**, and **Task Ease**) differ as a function of *task scenario* ($p < .03$), confirming that our tasks vary with respect to difficulty. Our results suggest that the ordering of the tasks from easiest to most difficult is 1, 4, 2, and 3,[8] which is consistent with our predictions. Recall that for Task 1, the initial query was designed to yield the correct train for both LT and CT. For tasks 4 and 2, the initial query was designed to yield the correct train for only one agent, and to require a follow-up query for the other.

[5] However, the analysis in Section 6 suggests that **Help Requests** is not a good predictor of performance.

[6] In our data, actual task success implies perceived task success, but not vice-versa.

[7] However, our "difficult" tasks were not that difficult (we wanted to minimize subjects' time commitment).

[8] This ordering is observed for all the listed measures except **User Turns**, which reverses tasks 4 and 1.

For Task 3, the initial query was designed to require a follow-up query for both agents.

6 Performance Function Estimation

While hypothesis testing tells us how each evaluation measure differs as a function of strategy and/or task, it does not tell us how to tradeoff or combine results from multiple measures. Understanding such tradeoffs is especially important when different measures yield different performance predictions (e.g., recall the Task 2 hypothesis testing results for **Completed** and **Help Requests**).

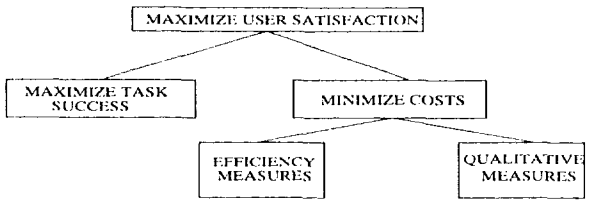

Figure 5: PARADISE's structure of objectives for spoken dialogue performance.

To assess the relative contribution of each evaluation measure to performance, we use PARADISE (Walker et al., 1997) to derive a performance function from our data. PARADISE draws on ideas in multi-attribute decision theory (Keeney and Raiffa, 1976) to posit the model shown in Figure 5, then uses multivariate linear regression to estimate a quantitative performance function based on this model. Linear regression produces coefficients describing the relative contribution of predictor factors in accounting for the variance in a predicted factor. In PARADISE, the success and cost measures are predictors, while user satisfaction is predicted. Figure 3 showed how the measures used to evaluate TOOT instantiate the PARADISE model.

The application of PARADISE to the TOOT data shows that the only significant contributors to **User Satisfaction** are **Completed** (Comp), **Mean Recognition** (MR) and **Barge Ins** (BI), and yields the following performance function:

$$\text{Perf} = .45\mathcal{N}(Comp) + .35\mathcal{N}(MR) - .42\mathcal{N}(BI)$$

Completed is significant at $p < .0002$, **Mean Recognition**[9] at $p < .003$, and **Barge Ins** at $p < .0004$; these account for 47% of the variance in **User Satisfaction**. $\mathcal{N}$ is a Z score normalization function (Cohen, 1995) and guarantees that the coefficients directly indicate the relative contribution of each factor to performance.

Our performance function demonstrates that TOOT performance involves task success and dialogue quality factors. Analysis of variance suggested that task success was a likely performance factor. PARADISE confirms this hypothesis, and demonstrates that perceived rather than actual task success is the useful predictor. While 39 dialogues were perceived to have been successful, only 27 were actually successful.

Results that were not apparent from the analysis of variance are that **Mean Recognition** and **Barge Ins** are also predictors of performance. The mean recognition for our corpus is 85%. Apparently, users of both LT and CT are bothered by dialogue phenomena associated with poor recognition. For example, system misunderstandings (which result from ASR misrecognitions) and system requests to repeat what users have said (which result from ASR rejections) both make dialogues seem less natural.

While barge-in is usually considered an advanced (and desirable) ASR capability, our performance function suggests that in TOOT, allowing users to interrupt actually degrades performance. Examination of our transcripts shows that users sometimes use barge-in to shorten TOOT's prompts. This often circumvents TOOT's confirmation strategy, which incorporates speech recognition results into prompts to make the user aware of misrecognitions.

Surprisingly, no efficiency measures are significant predictors of performance. This draws into question the frequently made assumption that efficiency is one of the most important measures of system performance, and instead suggests that users are more attuned to both task success and qualitative aspects of the dialogue, or that efficiency is highly correlated with some of these factors.

However, analysis of subsets of our data suggests that efficiency measures can become important performance predictors when the more primary effects are factored out. For example, when a regression is performed on the 11 TOOT dialogues with perfect **Mean Recognition**, the significant contributors to performance become **Completed** ($p < .05$), **Elapsed time** ($p < .04$), **User Turns** ($p < .03$) and **Barge Ins** ($p < 0.0007$) (accounting for 87% of the variance). Thus, in the presence of perfect ASR, efficiency becomes important. When a regression is performed using the 39 dialogues where users thought they had successfully completed the task

[9] Since we measure recognition rather than misrecognition, this "cost" factor has a positive coefficient.

(perfect **Completed**), the significant factors become **Elapsed time** ($p < .002$), **Timeouts** ($p < .002$), and **Barge Ins** ($p < .02$) (58% of the variance).

Applying the performance function to each of our 48 dialogues yields a performance estimate for each dialogue. Analysis with these estimates shows no significant differences for mean LT and CT performance. This result is consistent with the ANOVA result, where only one of the three (comparably weighted) factors in the performance function depends on response strategy (**Completed**). Note that for Tasks 2 and 4, the predictions in Table 1 do not hold for *overall* performance, despite the ANOVA results that the predictions do hold for some evaluation measures (e.g., **Completed** in Task 2).

7 Conclusion

We have presented an empirical comparison of literal and cooperative query response strategies in TOOT, illustrating the advantages of combining hypothesis testing and PARADISE. By using hypothesis testing to examine how a set of evaluation measures differ as a function of response strategy and task, we show that TOOT's cooperative and literal responses can both lead to greater task success, likelihood of future use, and user need for help, depending on task. By using PARADISE to derive a performance function, we show that a combination of strategy-dependent (perceived task success) and strategy-independent (number of barge-ins, mean recognition score) evaluation measures best predicts overall TOOT performance. Our results elaborate the conditions under which TOOT's response strategies lead to greater performance, and allow us to make predictions. For example, our performance equation predicts that improving mean recognition and/or judiciously restricting the use of barge-in will enhance performance. Our current research is aimed at automatically adapting dialogue behavior in TOOT, to increase mean recognition and thus overall agent performance (Walker et al., 1998).

Future work utilizing PARADISE will attempt to generalize our results, to make a more predictive model of agent performance. Performance function estimation needs to be done iteratively over different tasks and dialogue strategies. We plan to evaluate additional cooperative response strategies in TOOT (e.g., intensional summaries (Kalita et al., 1986), summarization and constraint elicitation in isolation), and to combine TOOT data with data from other agents (Walker et al., 1998).

8 Acknowledgments

Thanks to J. Chu-Carroll, T. Dasu, W. DuMouchel, J. Fromer, D. Hindle, J. Hirschberg, C. Kamm, J. Kang, A. Levy, C. Nakatani, S. Whittaker and J. Wilpon for help with this research and/or paper.

References

J. Allen and C. Perrault. 1980. Analyzing intention in utterances. *Artificial Intelligence*, 15.

P. Cohen. 1995. *Empirical Methods for Artificial Intelligence*. MIT Press, Boston.

M. Danieli and E. Gerbino. 1995. Metrics for evaluating dialogue strategies in a spoken language system. In *Proc. AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*.

D. Goddeau, H. Meng, J. Polifroni, S. Seneff, and S. Busayapongchai. 1996. A form-based dialogue manager for spoken language applications. In *Proc. ICSLP*.

A. Joshi, B. Webber, and R. Weischedel. 1984. Preventing false inferences. In *Proc. COLING*.

J. Kalita, M. Jones, and G. McCalla. 1986. Summarizing natural language database responses. *Computational Linguistics*, 12(2).

C. Kamm, S. Narayanan, D. Dutton, and R. Ritenour. 1997. Evaluating spoken dialog systems for telecommunication services. In *Proc. EUROSPEECH*.

S. Kaplan. 1981. Appropriate responses to inappropriate questions. In A. Joshi, B. Webber, and I. Sag, editors, *Elements of Discourse Understanding*. Cambridge University Press.

R. Keeney and H. Raiffa. 1976. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. Wiley.

E. Mays. 1980. Failures in natural language systems: Applications to data base query systems. In *Proc. AAAI*.

K. McCoy. 1989. Generating context-sensitive responses to object related misconceptions. *Artificial Intelligence*, 41(2).

J. Moore. 1994. *Participating in Explanatory Dialogues*. MIT Press.

C. Pao and J. Wilpon. 1992. Spontaneous speech collection for the ATIS domain with an aural user feedback paradigm. Technical report, AT&T.

R. Pieraccini, E. Levin, and W. Eckert. 1997. AMICA: The AT&T mixed initiative conversational architecture. In *Proc. EUROSPEECH*.

J. Polifroni, L. Hirschman, S. Seneff, and V. Zue. 1992. Experiments in evaluating interactive spoken language systems. In *Proc. DARPA Speech and NL Workshop*.

S. Seneff, V. Zue, J. Polifroni, C. Pao, L. Hetherington, D. Goddeau, and J. Glass. 1995. The preliminary development of a displayless PEGASUS system. In *Proc. ARPA Spoken Language Technology Workshop*.

E. Shriberg, E. Wade, and P. Price. 1992. Human-machine problem solving using spoken language systems (SLS): Factors affecting performance and user satisfaction. In *Proc. DARPA Speech and NL Workshop*.

M. Walker, D. Litman, C. Kamm, and A. Abella. 1997. PARADISE: A general framework for evaluating spoken dialogue agents. In *Proc. ACL/EACL*.

M. Walker, D. Litman, C. Kamm, and A. Abella. 1998. Evaluating spoken dialogue agents with PARADISE: Two case studies. *Computer Speech and Language*.