Learning Optimal Dialogue Strategies: A Case Study of a Spoken Dialogue Agent for Email

Marilyn A. Walker
walker@research.att.com
ATT Labs Research
180 Park Ave.
Florham Park, NJ 07932

Jeanne C. Fromer
jeannie@ai.mit.edu
MIT AI Lab
545 Technology Square
Cambridge, MA, 02139

Shrikanth Narayanan
shri@research.att.com
ATT Labs Research
180 Park Ave.
Florham Park, NJ 07932

Abstract

This paper describes a novel method by which a dialogue agent can learn to choose an optimal dialogue strategy. While it is widely agreed that dialogue strategies should be formulated in terms of communicative intentions, there has been little work on automatically optimizing an agent's choices when there are multiple ways to realize a communicative intention. Our method is based on a combination of learning algorithms and empirical evaluation techniques. The learning component of our method is based on algorithms for reinforcement learning, such as dynamic programming and Q-learning. The empirical component uses the PARADISE evaluation framework (Walker et al., 1997) to identify the important performance factors and to provide the performance function needed by the learning algorithm. We illustrate our method with a dialogue agent named ELVIS (EmaiL Voice Interactive System), that supports access to email over the phone. We show how ELVIS can learn to choose among alternate strategies for agent initiative, for reading messages, and for summarizing email folders.

1 Introduction

This paper describes a novel method by which a dialogue agent can learn to choose an optimal dialogue strategy. The main problem for dialogue agents is deciding *what* information to communicate to a hearer and *how* and *when* to communicate it. For example, consider one of the strategy choices faced by a spoken dialogue agent that accesses email by phone. When multiple messages match the user's query, e.g. *Read my messages from Kim*, an email agent must choose among multiple response strategies. The agent might choose the Read-First strategy in D1:

(D1) A: In the messages from Kim, there's 1 message about "Interviewing Antonio" and 1 message about "Meeting Today." The first message is titled, "Interviewing Antonio." It says, "I'd like to interview him. I could also go along to lunch. Kim."

D1 involves summarizing all the messages from Kim, and then taking the initiative to read the first one. Alternate strategies are the Read-Summary-Only strategy in D2, where the agent provides information that allows users to refine their selection criteria, and the Read-Choice-Prompt strategy in D3, where the agent explicitly tells the user what to say in order to refine the selection:

(D2) A: In the messages from Kim, there's 1 message about "Interviewing Antonio" and 1 message about "Meeting Today."

(D3) A: In the messages from Kim, there's 1 message about "Interviewing Antonio" and 1 message about "Meeting Today." To hear the messages, say, "Interviewing Antonio" or "Meeting."

Decision theoretic planning can be applied to the problem of choosing among strategies, by associating a utility U with each strategy (action) choice and by positing that agents should adhere to the Maximum Expected Utility Principle (Keeney and Raiffa, 1976; Russell and Norvig, 1995),

> **Maximum Expected Utility Principle:** An optimal action is one that maximizes the expected utility of outcome states.

An agent acts optimally by choosing a strategy a in state S_i that maximizes $U(S_i)$. But how are the utility values $U(S_i)$ for each dialogue state S_i derived?

Several reinforcement learning algorithms based on dynamic programming specify a way to calculate $U(S_i)$ in terms of the utility of a successor state S_j (Bellman, 1957; Watkins, 1989; Sutton, 1991; Barto et al., 1995). Thus if we know the utility for

the final state of the dialogue, we can calculate the utilities for all the earlier states. However, until recently there as been no way of determining a performance function for assigning a utility to the final state of a dialogue.

This paper presents a method based on dynamic programming by which dialogue agents can learn to optimize their choice of dialogue strategies. We draw on the recently proposed PARADISE evaluation framework (Walker et al., 1997) to identify the important performance factors and to provide a performance function for calculating the utility of the final state of a dialogue. We illustrate our method with a dialogue agent named ELVIS (EmaiL Voice Interactive System), that supports access to email over the phone. We test alternate strategies for agent initiative, for reading messages, and for summarizing email folders. We report results from modeling a corpus of 232 spoken dialogues in which ELVIS conversed with human users to carry out a set of email tasks.

2 Method for Learning to Optimize Dialogue Strategy Selection

Our method for learning to optimize dialogue strategy selection combines the application of PARADISE to empirical data (Walker et al., 1997), with algorithms for learning optimal strategy choices. PARADISE provides an empirical method for deriving a performance function that calculates overall agent performance as a linear combination of a number of simpler metrics. Our learning method consists of the following sequence of steps:

- Implement a spoken dialogue agent for a particular domain.
- Implement multiple dialogue strategies and design the agent so that strategies are selected randomly or under experimenter control.
- Define a set of dialogue tasks for the domain, and their information exchange requirements. Represent these tasks as attribute-value matrices to facilitate calculating task success.
- Collect experimental dialogues in which a number of human users converse with the agent to do the tasks.
- For each experimental dialogue:
  - Log the history of the state-strategy choices for each dialogue. Use this to estimate a state transition model.
  - Log a range of quantitative and qualitative cost measures for each dialogue, either automatically or with hand-tagging.
  - Collect user satisfaction reports for each dialogue.
- Use multivariate linear regression with user satisfaction as the dependent variable and task success and the cost measures as independent variables to determine a performance equation.
- Apply the derived performance equation to each dialogue to determine the utility of the final state of the dialogue.
- Use reinforcement learning to propagate the utility of the final state back to states S_i where strategy choices were made to determine which action maximizes $U(S_i)$.

These steps consist of those for deriving a performance function (Section 3), and for using the derived performance function as feedback to the agent with a learning algorithm (Section 4).

3 Using PARADISE to Derive a Performance Function

3.1 ELVIS Spoken Dialogue System

ELVIS is implemented using a general-purpose platform for spoken dialogue agents (Kamm et al., 1997). The platform consists of a speech recognizer that supports barge-in so that the user can interrupt the agent when it is speaking. It also provides an audio server for both voice recordings and text-to-speech (TTS), an interface between the computer running ELVIS and the telephone network, a module for application specific functions, and modules for specifying the application grammars and the dialogue manager. Our experiments are based on modifications to the dialogue manager as described below.

The dialogue manager is based on a state machine. Each state specifies transitions to other states and the conditions that license these transitions, as well as a grammar for what the user can say. State definitions also include the specification of agent *prompts* in terms of templates, with variables that are instantiated each time the state is entered. Prompts include: (1) an initial prompt, which the agent says upon entering the state (this may include a response to the user's current request); (2) a help prompt which the agent says if the user says *help*; (3) multiple rejection prompts which the agent says if the speech recognizer confidence is too low to continue without more user input; (4) multiple time-out prompts which the agent produces if the user doesn't say anything.

Each of these specifications is affected by the agent's dialogue strategy. An agent's dialogue strategy is implemented as a combination of the prompts

that are played to the user and the state transitions that the agent makes in response to the user's utterance. In particular, alternative prompts can be specified for all types of prompts (initial, help, rejection and timeout) to provide alternate dialogue strategies in each state. We implemented alternate strategies for reading messages, for initiative and for summarization.

The alternate strategies for reading messages are Read-First, Read-Summarize-Only and Read-Choice-Prompt as in dialogues D1, D2, and D3.

The alternate strategies for initiative are a system-initiative (SI) strategy and a mixed-initiative (MI) strategy. In the SI strategy, the system maintains control of the dialogue and the user must specify one query parameter at a time, as in D4.

(D4) A: Hi, Elvis here. You have 5 new and 0 unread messages in your inbox. Say Read, Summarize, or say Help for more options.
U: Read.
A: Select by Sender, Subject or in order of arrival?
U: Sender.
A: Which Sender?
U: Kim.
A: In the messages from Kim, there's 1 message about "Interviewing Antonio" .....

In the MI strategy, the system never tells the user what to do, and the user can take control of the dialogue and specify exactly what they want to do in a single utterance, as in D5.

(D5) A: Hi, Elvis here. I've got your mail.
U: Read me my messages from Kim.

We also implemented three alternate summarization strategies as responses to a user request to *Summarize my messages.* The basis for these alternates is ELVIS's decision about which message attributes to mention in the summary. The Summarize-Choice strategy in D6 asks the user to specify which of the relevant attributes to summarize by.

(D6) A: Summarize by subject, by sender, or both?
U: Subject.
A: In your top level inbox, there's 1 message about "Lunch," 1 message about "Interviewing Antonio," 1 message about "Call Me Tomorrow," 1 message about "Evaluation Group Meeting," and 1 message about "Agent Personality."

The Summarize-Both strategy in D7 uses all attributes in the summary:

(D7) A: In your top level inbox, from Kim, there's 1 message about "Lunch." From Michael, there's 1 message about "Evaluation group meeting." From Noah, there's 1 message about "Call Me Tomorrow" and 1 message about "Interviewing Antonio." And from Owen, there's 1 message about "Agent Personality."

In the Summarize-System strategy in D8, ELVIS summarizes by subject or by sender based on the current context. For instance, if the user is in the top level inbox, ELVIS will summarize by sender, but if the user is situated in a folder containing messages from Noah, ELVIS will summarize by subject, as a summary by sender would provide no new information.

(D8) A: In your top level inbox, there's 1 message from Kim, 2 messages from Noah, 1 message from Michael, and 1 message from Owen.

Transitions between states are driven by the user's conversational behavior, such as whether s/he says anything and what s/he says, the semantic interpretation of the user's utterances, and the settings of the agent's dialogue strategy parameters.

3.2 Experimental Design

Experimental dialogues were collected via two experiments in which users (AT&T summer interns and MIT graduate students) interacted with ELVIS to complete three representative application tasks that required them to access email messages in three different email inboxes. In the second experiment, users participated in a tutorial dialogue before doing the three tasks. The first experiment varied initiative strategies and the second experiment varied the presentation strategies for reading messages and summarizing folders. In order to have adequate data for learning, the agent must explore the space of strategy combinations and collect enough samples of each combination. In the second experiment, we parameterized the agent so that each user interacted with three different versions of ELVIS, one for each task. These experiments resulted in a corpus of 108 dialogues testing the initiative strategies, and a corpus of 124 dialogues testing the presentation strategies.

Each of the three tasks were performed in sequence, and each task consisted of two scenarios. Following PARADISE, the agent and the user had to exchange information about criteria for selecting messages and information within the message body in each scenario. Scenario 1.1 is typical.

- 1.1: You are working at home in the morning and plan to go directly to a meeting when you go into

work. Kim said she would send you a message telling you where and when the meeting is. Find out **the Meeting Time** and **the Meeting Place**.

Scenario 1.1 is represented in terms of the attribute value matrix (AVM) in Table 1. Successful completion of a scenario requires that all attribute-values must be exchanged (Walker et al., 1997). The AVM representation for all six scenarios is similar to Table 1, and is independent of ELVIS's dialogue strategy.

attribute	actual value
Selection Criteria	Kim V Meeting
Email.att1	10:30
Email.att2	2D516

Table 1: Attribute value matrix instantiation, Key for Email Scenario 1.1

3.3 Data Collection

Three different methods are used to collect the measures for applying the PARADISE framework and the data for learning: (1) All of the dialogues are recorded; (2) The dialogue manager logs the agent's dialogue behavior and a number of other measures discussed below; (3) Users fill out web page forms after each task (task success and user satisfaction measures). Measures are in **boldface** below.

The dialogue recordings are used to transcribe the user's utterances to derive performance measures for speech recognition, to check the timing of the interaction, to check whether users barged in on agent utterances (**Barge In**), and to calculate the elapsed time of the interaction (**ET**).

For each state, the system logs which dialogue strategy the agent selects. In addition, the number of timeout prompts (**Timeout Prompts**), **Recognizer Rejections**, and the times the user said *Help* (**Help Requests**) are logged. The number of **System Turns** and the number of **User Turns** are calculated on the basis of this data. In addition, the recognition result for the user's utterance is extracted from the recognizer and logged. The transcriptions are used in combination with the logged recognition result to calculate a concept accuracy measure for each utterance.[1] Mean concept accuracy is then calculated over the whole dialogue and used as a Mean Recognition Score **MRS** for the dialogue.

The web page forms are the basis for calculating Task Success and User Satisfaction measures. Users reported their perceptions as to whether they had completed the task (**Comp**),[2] and filled in an AVM with the information that they had acquired from the agent, e.g. the values for Email.att1 and Email.att2 in Table 1. The AVM matrix supports calculating **Task Success** objectively by using the Kappa statistic to compare the information in the AVM that the users filled in with an AVM key such as that in Table 1 (Walker et al., 1997).

In order to calculate User Satisfaction, users were asked to evaluate the agent's performance with a user satisfaction survey. The data from the survey resulted in user satisfaction values that range from 0 to 33. See (Walker et al., 1998) for more details.

3.4 Deriving a Performance Function

Overall, the results showed that users could successfully complete the tasks with all versions of ELVIS. Most users completed each task in about 5 minutes and average κ over all subjects and tasks was .82. However, there were differences between strategies; as an example see Table 2.

Measure	SYSTEM (SI)	MIXED (MI)
Kappa	.81	.83
Comp	.83	.78
User Turns	25.94	17.59
System Turns	28.18	21.74
Elapsed Time (ET)	328.59 s	289.43 s
MeanRecog (MRS)	.88	.72
Time Outs	2.24	4.15
Help Requests	.70	.94
Barge Ins	5.2	3.5
Recognizer Rejects	.98	1.67
User Satisfaction	26.6	23.7

Table 2: Performance measure means per dialogue for Initiative Strategies

PARADISE provides a way to calculate dialogue agent performance as a linear combination of a number of simpler metrics that can be directly measured such as those in Table 2. Performance for any (sub)dialogue D is defined by the following equation:

$$\text{Performance} = (\alpha * \mathcal{N}(\kappa)) - \sum_{i=1}^{n} w_i * \mathcal{N}(c_i)$$

[1] For example, the utterance *Read my messages from Kim* contains two concepts, the *read* function, and the *sender:kim* selection criterion. If the system understood only that the user said *Read*, concept accuracy would be .5.

[2] *Yes,No* responses are converted to *1,0*.

where α is a weight on κ, c_i are the cost functions, which are weighted by w_i, and $\mathcal{N}$ is a Z score normalization function (Walker et al., 1997; Cohen, 1995). The Z score normalization function ensures that, when the weights α and w_i are solved for, that the magnitude of the weights reflect the magnitude of the contribution of that factor to performance. The performance function is derived through multivariate linear regression with **User Satisfaction** as the dependent variable and all the other measures as independent variables (Walker et al., 1997). See Table 2. In the ELVIS data, an initial regression over the measures in Table 2 suggests that **Comp**, **MRS** and **ET** are the only significant contributors to **User Satisfaction**. A second regression including only these factors results in the following equation:

$$\text{Performance} = .21*Comp + .47*MRS - .15*ET$$

with **Comp** (t=2.58, p =.01), **MRS** (t =5.75, p =.0001) and **ET** (t=-1.8, p=.07) significant predictors, accounting for 38% of the variance in R-Squared (F (3,104)=21.2, p <.0001). The magnitude of the coefficients in this equation demonstrates the performance of the speech recognizer (**MRS**) is the most important predictor, followed by users' perception of Task Success (**Comp**) and efficiency (**ET**). In the next section, we show how to use this derived performance equation to compute the utility of the final state of the dialogue.

4 Applying Q-learning to ELVIS Experimental Data

The basic idea is to apply the performance function to the measures logged for each dialogue D_i, thereby replacing a range of measures with a single performance value P_i. Given the performance values P_i, any of a number of automatic learning algorithms can be used to determine which sequence of action choices (dialogue strategies) maximize utility, by using P_i as the utility for the final state of the dialogue D_i. Possible algorithms include Genetic Algorithms, Q-learning, TD-Learning, and Adaptive Dynamic Programming (Russell and Norvig, 1995). Here we use Q-learning to illustrate the method (Watkins, 1989). See (Fromer, 1998) for experiments using alternative algorithms.

The utility of doing action a in state S_i, $U(a, S_i)$ (its Q-value), can be calculated terms of the utility of a successor state S_j, by obeying the following recursive equation:

$$U(a, S_i) = R(S_i) + \sum_j M_{ij}^a \max_{a'} U(a', S_j)$$

where $R(S_i)$ is a reward associated with being in state S_i, a is a strategy from a finite set of strategies A that are admissable in state S_i, and M_{ij}^a is the probability of reaching state S_j if strategy a is selected in state S_i.

In the experiments reported here, the reward associated with each state, $R(S_i)$, is zero.[3] In addition, since reliable a priori prediction of a user action in a particular state is not possible (for example the user may say *Help* or the speech recognizer may fail to understand the user), the state transition model M_{ij}^a is estimated from the logged state-strategy history for the dialogue.

The utility values can be estimated to within a desired threshold using Value Iteration, which updates the estimate of $U(a, S_i)$, based on updated utility estimates for neighboring states, so that the equation above becomes:

$$U_{n+1}(a, S_i) = R(S_i) + \sum_j M_{ij}^a \max_{a'} U_n(a', S_j)$$

where $U_n(a, S_i)$ is the utility estimate for doing a in state S_i after n iterations. Value Iteration stops when the difference between $U_n(a, S_i)$ and $U_{n+1}(a, S_i)$ is below a threshold, and utility values have been associated with states where strategy selections were made. After experimenting with various threshholds, we used a threshold of 5% of the performance range of the dialogues.

The result of applying Q-learning to ELVIS data for the initiative strategies is illustrated in Figure 1. The figure plots utility estimates for SI and MI over time. It is clear that the SI strategy is better because it has a higher utility: at the end of 108 training sessions (dialogues), the utility of SI is estimated at .249 and the utility of MI is estimated at -0.174.

TYPE	STRATEGY	UTILITY
Read	Read-First	.21
	Read-Choice-Prompt	.07
	Read-Summarize-Only	.08
Summarize	Summarize-System	.162
	Summarize-Choice	-0.03
	Summarize-Both	.09

Table 3: Utilities for Presentation Strategy Choices after 124 Training Sessions

The SI and MI strategies affect the whole dialogue; the presentation strategies apply locally and

[3] See (Fromer, 1998) for experiments in which local rewards are nonzero.

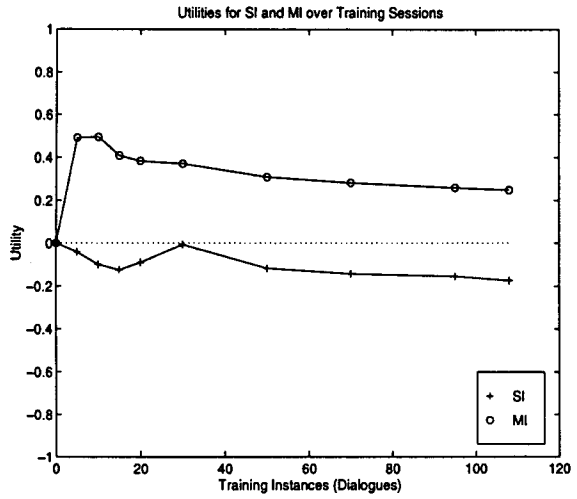

Figure 1: Results of applying Q-learning to System-Initiative (SI) and Mixed-Initiative (MI) Strategies for 108 ELVIS Dialogues

can be actived in different states of the dialogue. We examined the variation in a strategy's utility at each phase of the task, by representing the task as having three phases: no scenarios completed, one scenario completed and both scenarios completed. Table 3 reports utilities for the use of a strategy after one scenario was completed . The policy implied by the utilities at other phases of the task are the same. See (Fromer, 1998) for more detail.

The Read-First strategy in D1 has the best performance of the read strategies. This strategy takes the initiative to read a message, which might result in messages being read that the user wasn't interested in. However since the user can barge-in on system utterances, perhaps little is lost by taking the initiative to start reading a message. After 124 training sessions, the best summarize strategy is Summarize-System, which automatically selects which attributes to summarize by, and so does not incur the cost of asking the user to specify these attributes. However, the utilities for the Summarize-Choice strategy have not completely converged after 124 trials.

5 Conclusions and Future Work

This paper illustrates a novel technique by which an agent can learn to choose an optimal dialogue strategy. We illustrate our technique with ELVIS, an agent that supports access to email by phone, with strategies for initiative, and for reading and summarizing messages. We show that ELVIS can learn that the System-Initiative strategy has higher utility than the Mixed-Initiative strategy, that Read-First is the best read strategy, and that Summarize-System is the best summary strategy.

Here, our method was illustrated by evaluating strategies for managing initiative and for message presentation. However there are numerous dialogue strategies that an agent might use, e.g. to gather information, handle errors, or manage the dialogue interaction (Chu-Carroll and Carberry, 1995; Danieli and Gerbino, 1995; Hovy, 1993; McKeown, 1985; Moore and Paris, 1989). Previous work in natural language generation has proposed heuristics to determine an agent's choice of dialogue strategy, based on factors such as discourse focus, medium, style, and the content of previous explanations (McKeown, 1985; Moore and Paris, 1989; Maybury, 1991; Hovy, 1993). It should be possible to test experimentally whether an agent can automatically learn these heuristics since the methodology we propose is general, and could be applied to any dialogue strategy choice that an agent might make.

Previous work has also proposed that an agent's choice of dialogue strategy can be treated as a stochastic optimization problem (Walker, 1993; Biermann and Long, 1996; Levin and Pieraccini, 1997). However, to our knowledge, these methods have not previously been applied to interactions with real users. The lack of an appropriate performance function has been a critical methodological limitation.

We use the PARADISE framework (Walker et al., 1997) to derive an empirically motivated performance function, that combines both subjective user preferences and objective system performance measures into a single function. It would have been impossible to predict a priori which dialogue factors influence the usability of a dialogue agent, and to what degree. Our performance equation shows that both dialogue quality and efficiency measures contribute to agent performance, but that dialogue quality measures have a greater influence. Furthermore, in contrast to assuming an a priori model, we use the dialogues from real user-system interactions to provide realistic estimates of M_{ij}^{a}, the state transition model used by the learning algorithm. It is impossible to predict a priori the transition frequencies, given the imperfect nature of spoken language understanding, and the unpredictability of user be-

havior.

The use of this method introduces several open issues. First, the results of the learning algorithm are dependent on the representation of the state space. In many reinforcement learning problems (e.g. backgammon), the state space is pre-defined. In spoken dialogue systems, the system designers construct the state space and decide what state variables need to be monitored. Our initial results suggest that the state representation that the agent uses to interact with the user may not be the optimal state representation for learning. See (Fromer, 1998). Second, in advance of actually running learning experiments, it is not clear how much experience an agent will need to determine which strategy is better. Figure 1 shows that it took no more than 50 dialogue samples for the algorithm to show the differences in convergence trends when learning about initiative strategies. However, it appears that more data is needed to learn to distinguish between the summarization strategies. Third, our experimental data is based on short-term interactions with novice users, but we might expect that users of an email agent would engage in many interactions with the same agent, and that preferences for agent interaction strategies could change over time with user expertise. This means that the performance function might change over time. Finally, the learning algorithm that we report here is an *off-line* algorithm, i.e. the agent collects a set of dialogues and then decides on an optimal strategy as a result. In contrast, it should be possible for the agent to learn *on-line*, during the course of a dialogue, if the performance function could be automatically calculated (or approximated). We are exploring these issues in ongoing work.

6 Acknowledgements

G. Di Fabbrizio, D. Hindle, J. Hirschberg, C. Kamm, and D. Litman provided assistance with this research or paper.

References

A.G. Barto, S. J. Bradtke, and S. P. Singh. 1995. Learning to act using real-time dynamic programming. *Artificial Intelligence Journal*, 72(1-2):81–138.

R. E. Bellman. 1957. *Dynamic Programming*. Princeton University Press, Princeton, N.J.

A. W. Biermann and Philip M. Long. 1996. The composition of messages in speech-graphics interactive systems. In *Proc. of the 1996 International Symposium on Spoken Dialogue*, pp. 97–100.

J. Chu-Carroll and S. Carberry. 1995. Response generation in collaborative negotiation. In *Proc. of the 33rd Annual Meeting of the ACL*, pp. 136–143.

P. R. 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. of the 1995 AAAI Spring Symposium on Empirical Methods in Discourse*, pages 34–39.

J. C. Fromer. 1998. Learning optimal discourse strategies in a spoken dialogue system. Technical Report Forthcoming, MIT AI Lab M.S. Thesis.

E. H. Hovy. 1993. Automated discourse generation using discourse structure relations. *Artificial Intelligence Journal*, 63:341–385.

C. Kamm, S. Narayanan, D. Dutton, and R. Ritenour. 1997. Evaluating spoken dialog systems for telecommunication services. In *EUROSPEECH 97*.

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

E. Levin and R. Pieraccini. 1997. A stochastic model of computer-human interaction for learning dialogue strategies. In *EUROSPEECH 97*.

M.T. Maybury. 1991. Planning multi-media explanations using communicative acts. In *Proc. of the Ninth National Conf. on Artificial Intelligence*, pages 61–66.

K. R. McKeown. 1985. Discourse strategies for generating natural language text. *Artificial Intelligence*, 27(1):1–42, September.

J. D. Moore and C. L. Paris. 1989. Planning text for advisory dialogues. In *Proc. 27th Annual Meeting of the ACL*.

S. Russell and P. Norvig. 1995. *Artificial Intelligence: A Modern Approach*. Prentice Hall, N.J.

R. S. Sutton. 1991. Planning by incremental dynamic programming. In *Proc. Ninth Conf. on Machine Learning*, pages 353–357. Morgan-Kaufmann.

M. A. Walker, D. Litman, C. Kamm, and A. Abella. 1997. PARADISE: A general framework for evaluating spoken dialogue agents. In *Proc. of the 35th Annual Meeting of the ACL*, pp. 271–280.

M. Walker, J. Fromer, G. Di Fabbrizio, C. Mestel, and D. Hindle. 1998. What can I say: Evaluating a spoken language interface to email. In *Proc. of the Conf. on Computer Human Interaction (CHI 98)*.

M. A. Walker. 1993. *Informational Redundancy and Resource Bounds in Dialogue*. Ph.D. thesis, University of Pennsylvania.

C. J. Watkins. 1989. *Models of Delayed Reinforcement Learning*. Ph.D. thesis, Cambridge University.