

9

Learning Optimal Discourse Strategies in a Spoken Dialogue System

by

Jeanne C. Fromer

B.A., Barnard College, Columbia University (1996)

Submitted to the Department of Electrical Engineering and Computer Science

in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

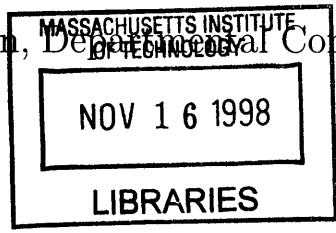
September 1998

© Massachusetts Institute of Technology 1998. All rights reserved.

Author
Department of Electrical Engineering and Computer Science
August 7, 1998

Certified by
Robert C. Berwick
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by
Arthur C. Smith
Chairman, Departmental Committee on Graduate Students



ENG

Learning Optimal Discourse Strategies in a Spoken Dialogue System

by

Jeanne C. Fromer

Submitted to the Department of Electrical Engineering and Computer Science
on August 7, 1998, in partial fulfillment of the requirements for the degree of
Master of Science

Abstract

Participants in a conversation can often realize their conversational goals in multiple ways by employing different *discourse strategies*. For example, one can usually present requested information in various ways; different presentation methods are preferred and most effective in varying contexts. One can also manage conversations, or assume *initiative*, to varying degrees by directing questions, issuing commands, restricting potential responses, and controlling discussion topics in different ways.

Agents that converse with users in natural language and possess different discourse strategies need to choose and realize the optimal strategy from competing strategies. Previous work in natural language generation has selected discourse strategies by using heuristics based on discourse focus, medium, style, and the content of previous utterances. Recent work suggests that an agent can *learn* which strategies are optimal. This thesis investigates the issues involved with learning optimal discourse strategies on the basis of experience gained through conversations between human users and natural language agents.

A spoken dialogue agent, ELVIS, is implemented as a testbed for learning optimal discourse strategies. ELVIS provides telephone-based voice access to a caller's email. Within ELVIS, various discourse strategies for the distribution of initiative, reading messages, and summarizing messages are implemented. Actual users interact with discourse strategy-based variations of ELVIS. Their conversations are used to derive a dialogue performance function for ELVIS using the PARADISE dialogue evaluation framework. This performance function is then used with reinforcement learning techniques, such as adaptive dynamic programming, Q-learning, temporal difference learning, and temporal difference Q-learning, to determine the optimal discourse strategies for ELVIS to use in different contexts. This thesis reports and compares learning results and describes how the particular reinforcement algorithm, local reward functions, and the system state space representation affect the efficiency and the outcome of the learning results. This thesis concludes by suggesting how it may be possible to automate online learning in spoken dialogue systems by extending the presented evaluation and learning techniques.

Thesis Supervisor: Robert C. Berwick

Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

First and foremost, I would like to thank Marilyn (Lyn) Walker for her tremendously insightful guidance and gentle nudges over the past two years.

Special thanks to Bob Berwick for his support during the past two years.

Everyone that I have interacted with at AT&T Labs during the past few years has been extremely supportive of my research and has helped in the construction and debugging of ELVIS. I specifically want to thank Alicia Abella, Bruce Buntschuh, Giuseppe (Pino) di Fabrizio, Don Hindle, Julia Hirschberg, David Johnson, Candy Kamm, Diane Litman, Craig Mestel, Dana Miller, Shri Narayanan, Larry Rabiner, Peter Selfridge, Loren Terveen, Steve Whittaker, Jay Wilpon, and whomever I have inadvertently omitted from this list. I appreciate all of the resources that AT&T has provided for me.

Thanks to Patrick Winston, who taught me about the presentation of technical ideas.

Thanks to my family, for always encouraging me academically.

I would like to thank Mark (Meir) Zelcer, who has consistently encouraged me to stop procrastinating and has inspired all sorts of ideas that I never could have stumbled upon on my own.

I sincerely appreciate the regular and summer employees of AT&T Labs, the MIT students, and miscellaneous friends who took the time to participate in the ELVIS experiments, after I cornered them and pitifully begged for their assistance.

This work is supported by a National Science Foundation Graduate Research Fellowship. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 13 |
| 1.1 | Overview of the Problem | 13 |
| 1.2 | Overview of the Thesis | 15 |
| 2 | ELVIS, a Telephone-Based Email Agent | 17 |
| 2.1 | ELVIS Components and Design | 17 |
| 2.1.1 | General-Purpose Framework | 19 |
| 2.1.2 | Application-Specific Component | 20 |
| 2.2 | ELVIS Functionality | 20 |
| 2.2.1 | Wizard of Oz | 22 |
| 2.2.2 | System Functionality | 22 |
| 2.3 | Different Discourse Strategies | 23 |
| 2.3.1 | Initiative Strategies | 24 |
| 2.3.2 | Summarization Strategies | 25 |
| 2.3.3 | Read Strategies | 26 |
| 2.4 | ELVIS State Space Representation | 27 |
| 3 | Experimenting with ELVIS | 30 |
| 3.1 | Data Collection Methods | 30 |
| 3.1.1 | Task Scenarios | 31 |
| 3.1.2 | Dialogue Recordings | 32 |
| 3.1.3 | Dialogue Manager Logs | 32 |
| 3.1.4 | User Surveys | 33 |

| | | |
|----------|---|-----------|
| 3.2 | Experimental Results | 34 |
| 3.2.1 | Initiative Strategy Experimental Results | 34 |
| 3.2.2 | Presentation Strategies Experimental Results | 35 |
| 3.3 | Deriving a Performance Function | 35 |
| 4 | Learning Optimal Discourse Strategies | 39 |
| 4.1 | Reinforcement Learning | 39 |
| 4.1.1 | Overview | 40 |
| 4.1.2 | Algorithms | 40 |
| 4.1.3 | Partially Observable Markov Decision Problems | 43 |
| 4.2 | Local Reinforcement | 44 |
| 4.2.1 | Qualitative Approach | 44 |
| 4.2.2 | Using the Performance Function | 45 |
| 4.3 | State Space Representation | 46 |
| 4.3.1 | Separating States by Prompt Type | 46 |
| 4.3.2 | Strategy Branching | 47 |
| 4.3.3 | Representing Task Progress | 49 |
| 4.4 | The State Transition Model | 50 |
| 4.5 | Learning Results | 51 |
| 4.5.1 | Initiative Strategies | 51 |
| 4.5.2 | Summarization Strategies | 56 |
| 4.5.3 | Read Strategies | 65 |
| 5 | Related Work | 75 |
| 5.1 | Spoken Dialogue Systems | 75 |
| 5.1.1 | Previous Systems for Accessing Email | 76 |
| 5.1.2 | Initiative | 76 |
| 5.2 | Natural Language Content Planning | 77 |
| 5.3 | Adaptive Agents | 77 |
| 5.4 | Reinforcement Learning | 78 |
| 5.4.1 | State Space Representation | 78 |

| | | |
|----------|--|------------|
| 5.4.2 | Different Applications | 80 |
| 6 | Conclusions and Future Work | 82 |
| 6.1 | Results | 82 |
| 6.2 | Future Work | 83 |
| 6.2.1 | Automating Online Learning of Optimal Discourse Strategies . | 84 |
| A | ELVIS States and State Transitions | 86 |
| B | Experimental Inbox Folders | 90 |
| B.1 | Inbox Folder for Tutorial | 90 |
| B.2 | Inbox Folder for Task 1 | 92 |
| B.3 | Inbox Folder for Task 2 | 95 |
| B.4 | Inbox Folder for Task 3 | 98 |
| C | Experimental Task Scenarios | 102 |
| C.1 | Task Scenario for Tutorial | 102 |
| C.2 | Task Scenario for Task 1 | 102 |
| C.3 | Task Scenario for Task 2 | 103 |
| C.4 | Task Scenario for Task 3 | 104 |
| D | Experimental User Survey | 106 |
| E | Data Tables | 109 |
| | Bibliography | 123 |

List of Figures

| | | |
|------|---|----|
| 2-1 | A Conversation with ELVIS | 18 |
| 2-2 | ELVIS Architecture | 19 |
| 2-3 | ELVIS Functionality | 22 |
| 2-4 | Dialogue 1: System-Initiative Initiative Strategy | 24 |
| 2-5 | Dialogue 2: Mixed-Initiative Initiative Strategy | 25 |
| 2-6 | Dialogue 3: Summarize-System Summarization Strategy | 26 |
| 2-7 | Dialogue 4: Summarize-Choice Summarization Strategy | 26 |
| 2-8 | Dialogue 5: Summarize-Both Summarization Strategy | 27 |
| 2-9 | Dialogue 6: Read-First Read Strategy | 27 |
| 2-10 | Dialogue 7: Read-Summarize-Only Read Strategy | 28 |
| 2-11 | Dialogue 8: Read-Choice-Prompt Read Strategy | 28 |
| 2-12 | Annotation of Utterances 14-21 in Figure 2-1 | 29 |
| | | |
| 3-1 | Collected Experimental Measures | 31 |
| 3-2 | Typical Subtask for the ELVIS Experiments | 31 |
| 3-3 | PARADISE's Structure of Objectives for Spoken Dialogue Performance | 36 |
| | | |
| 4-1 | A Graph with a Reentrant Node | 47 |
| 4-2 | Figure 4-1 Transformed to Remove Reentrancy | 48 |
| 4-3 | Simplified Representation of Successfully Completed ELVIS Interaction | 49 |
| 4-4 | Figure 4-3 Transformed to Show Task Progress | 49 |
| 4-5 | Results of ADP and Q-Learning without Local Reward for Initiative Strategies | 52 |
| 4-6 | Results of ADP Learning with Local Reward for Initiative Strategies | 53 |

| | | |
|------|--|----|
| 4-7 | Results of Q-Learning with Local Reward for Initiative Strategies . . . | 54 |
| 4-8 | Comparison of Learning Algorithms and Local Reinforcement Use for System-Initiative | 54 |
| 4-9 | Comparison of Learning Algorithms and Local Reinforcement Use for Mixed-Initiative | 55 |
| 4-10 | Results of ADP and Q-Learning without Local Reward for Summa- rization Strategies at Task Progress Level 1 | 56 |
| 4-11 | Results of ADP and Q-Learning without Local Reward for Summa- rization Strategies at Task Progress Level 2 | 57 |
| 4-12 | Results of ADP and Q-Learning without Local Reward for Summa- rization Strategies at Task Progress Level 3 | 58 |
| 4-13 | Results of ADP and Q-Learning with Local Reward for Summarization Strategies at Task Progress Level 1 | 58 |
| 4-14 | Results of ADP and Q-Learning with Local Reward for Summarization Strategies at Task Progress Level 2 | 59 |
| 4-15 | Results of ADP and Q-Learning with Local Reward for Summarization Strategies at Task Progress Level 3 | 59 |
| 4-16 | Comparison of Learning Algorithms and Local Reinforcement Use for Summarize-Both at Task Progress Level 1 | 60 |
| 4-17 | Comparison of Learning Algorithms and Local Reinforcement Use for Summarize-Both at Task Progress Level 2 | 61 |
| 4-18 | Comparison of Learning Algorithms and Local Reinforcement Use for Summarize-Both at Task Progress Level 3 | 61 |
| 4-19 | Comparison of Learning Algorithms and Local Reinforcement Use for Summarize-Choice at Task Progress Level 1 | 62 |
| 4-20 | Comparison of Learning Algorithms and Local Reinforcement Use for Summarize-Choice at Task Progress Level 2 | 62 |
| 4-21 | Comparison of Learning Algorithms and Local Reinforcement Use for Summarize-Choice at Task Progress Level 3 | 63 |

| | | |
|------|---|----|
| 4-22 | Comparison of Learning Algorithms and Local Reinforcement Use for Summarize-System at Task Progress Level 1 | 63 |
| 4-23 | Comparison of Learning Algorithms and Local Reinforcement Use for Summarize-System at Task Progress Level 2 | 64 |
| 4-24 | Comparison of Learning Algorithms and Local Reinforcement Use for Summarize-System at Task Progress Level 3 | 64 |
| 4-25 | Results of ADP and Q-Learning without Local Reward for Read Strategies at Task Progress Level 1 | 65 |
| 4-26 | Results of ADP and Q-Learning without Local Reward for Read Strategies at Task Progress Level 2 | 66 |
| 4-27 | Results of ADP and Q-Learning without Local Reward for Read Strategies at Task Progress Level 3 | 67 |
| 4-28 | Results of ADP and Q-Learning with Local Reward for Read Strategies at Task Progress Level 1 | 68 |
| 4-29 | Results of ADP and Q-Learning with Local Reward for Read Strategies at Task Progress Level 2 | 68 |
| 4-30 | Results of ADP and Q-Learning with Local Reward for Read Strategies at Task Progress Level 3 | 69 |
| 4-31 | Comparison of Learning Algorithms and Local Reinforcement Use for Read-Choice-Prompt at Task Progress Level 1 | 70 |
| 4-32 | Comparison of Learning Algorithms and Local Reinforcement Use for Read-Choice-Prompt at Task Progress Level 2 | 70 |
| 4-33 | Comparison of Learning Algorithms and Local Reinforcement Use for Read-Choice-Prompt at Task Progress Level 3 | 71 |
| 4-34 | Comparison of Learning Algorithms and Local Reinforcement Use for Read-First at Task Progress Level 1 | 71 |
| 4-35 | Comparison of Learning Algorithms and Local Reinforcement Use for Read-First at Task Progress Level 2 | 72 |
| 4-36 | Comparison of Learning Algorithms and Local Reinforcement Use for Read-First at Task Progress Level 3 | 72 |

| | |
|--|----|
| 4-37 Comparison of Learning Algorithms and Local Reinforcement Use for Read-Summarize-Only at Task Progress Level 1 | 73 |
| 4-38 Comparison of Learning Algorithms and Local Reinforcement Use for Read-Summarize-Only at Task Progress Level 2 | 73 |
| 4-39 Comparison of Learning Algorithms and Local Reinforcement Use for Read-Summarize-Only at Task Progress Level 3 | 74 |

List of Tables

| | | |
|-----|--|-----|
| 2.1 | Functional Analysis of Wizard of Oz Dialogues | 21 |
| 3.1 | Attribute-Value Matrix Instantiation for Subtask in Figure 3-2 | 32 |
| 3.2 | Performance Measure Means per Dialogue for Initiative Strategies | 34 |
| 3.3 | Performance Measure Means per Dialogue for Summarization and Read Strategies | 35 |
| 4.1 | Final Utilities from ADP and Q-Learning for Summarization Strategies | 60 |
| 4.2 | Summarization Strategy Preferences Learned through ADP and Q-Learning | 60 |
| 4.3 | Final Utilities from ADP and Q-Learning for Read Strategies | 66 |
| 4.4 | Read Strategy Preferences Learned through ADP and Q-Learning | 69 |
| C.1 | Attribute-Value Matrix Key for Subtask 0.1 | 103 |
| C.2 | Attribute-Value Matrix Key for Subtask 0.2 | 103 |
| C.3 | Attribute-Value Matrix Key for Subtask 1.1 | 104 |
| C.4 | Attribute-Value Matrix Key for Subtask 1.2 | 104 |
| C.5 | Attribute-Value Matrix Key for Subtask 2.1 | 105 |
| C.6 | Attribute-Value Matrix Key for Subtask 2.2 | 105 |
| C.7 | Attribute-Value Matrix Key for Subtask 3.1 | 105 |
| C.8 | Attribute-Value Matrix Key for Subtask 3.2 | 105 |
| E.1 | Utilities for Initiative Strategies Learned via ADP | 109 |
| E.2 | Utilities for Initiative Strategies Learned via Q-Learning | 110 |
| E.3 | Utilities for Summarize-Both Summarize Strategy Learned via ADP | 111 |

| | | |
|------|--|-----|
| E.4 | Utilities for Summarize-Choice Summarize Strategy Learned via ADP | 112 |
| E.5 | Utilities for Summarize-System Summarize Strategy Learned via ADP | 113 |
| E.6 | Utilities for Summarize-Both Summarize Strategy Learned via Q-Learning | 114 |
| E.7 | Utilities for Summarize-Choice Summarize Strategy Learned via Q-Learning | 115 |
| E.8 | Utilities for Summarize-System Summarize Strategy Learned via Q-Learning | 116 |
| E.9 | Utilities for Read-Choice-Prompt Read Strategy Learned via ADP . . | 117 |
| E.10 | Utilities for Read-First Read Strategy Learned via ADP | 118 |
| E.11 | Utilities for Read-Summarize-Only Read Strategy Learned via ADP . | 119 |
| E.12 | Utilities for Read-Choice-Prompt Read Strategy Learned via Q-Learning | 120 |
| E.13 | Utilities for Read-First Read Strategy Learned via Q-Learning | 121 |
| E.14 | Utilities for Read-Summarize-Only Read Strategy Learned via Q-Learning | 122 |

Chapter 1

Introduction

This thesis presents a spoken dialogue system that can learn optimal discourse strategies through interactions with actual users. This thesis first describes the spoken dialogue system, ELVIS, which supports access to email by telephone and then presents our approach to learning, which consists of three steps. Different discourse strategies are implemented in ELVIS for initiative, reading messages, and summarizing messages. Users interact with discourse strategy-based variations of ELVIS in an experimental setting. The experimental results are then used to derive a dialogue performance function. Finally, the performance function is used as the basis for global and local reward in reinforcement learning techniques, which are applied to learn optimal discourse strategies in ELVIS. The results of applying different reinforcement learning algorithms are presented and compared. This thesis shows that the particular reinforcement learning algorithm, the system's state space representation, and the use of local reinforcement significantly affect the outcomes and efficiency of learning.

1.1 Overview of the Problem

Participants in a conversation can often realize their conversational goals in multiple ways by employing different discourse strategies. For example, one can usually present requested information in various ways; different presentation methods are

preferred and most effective in varying contexts. One can also manage conversations, or assume *initiative*, to varying degrees by directing questions, issuing commands, restricting potential responses, and controlling discussion topics in different ways. Agents that converse with users in natural language and possess different discourse strategies need to choose and realize the optimal strategy from competing strategies. Previous work in natural language generation has selected discourse strategies by using heuristics based on discourse focus, medium, style, and the content of previous utterances. Recent work suggests that an agent can *learn* which strategies are optimal. This thesis investigates the issues involved with learning optimal discourse strategies on the basis of experience gained through conversations between human users and natural language agents. Reinforcement learning is particularly suited to learning optimal discourse strategies in spoken dialogue systems because no examples of optimal strategies can be provided to these systems initially since they are unknown. Spoken dialogue systems must learn a predictive model of optimal discourse strategies by exploring examples (i.e., conversations with actual users) and rewarding them.

As a testbed for learning optimal discourse strategies, a spoken dialogue agent, ELVIS, is implemented [Walker *et al.*, 1998a, Walker *et al.*, 1998b, Walker *et al.*, 1997a]. ELVIS provides telephone-based voice access to a caller's email. Within ELVIS, various discourse strategies for the distribution of initiative and presentation of information are implemented. The implemented *initiative strategies* are **System Initiative**, where ELVIS controls the conversation by explicitly prompting the user for information, and **Mixed Initiative**, where ELVIS assumes that the user knows what to say and, by default, the user must direct the conversation. The information *presentation strategies* are for summarizing email messages and for reading messages. The implemented summarization strategies are **Summarize-System**, **Summarize-Choice**, and **Summarize-Both**. The implemented read strategies are **Read-First**, **Read-Summarize-Only**, and **Read-Choice-Prompt**. The different presentation strategies are detailed in Sections 2.3.2 and 2.3.3. ELVIS and the listed initiative and presentation strategies are implemented as a testbed for learning; a robust real-world

version of ELVIS would incorporate more diverse, complex strategies.

Actual users interact with discourse strategy-based variations of ELVIS. Their conversations are used to derive a dialogue performance function for ELVIS using the PARADISE dialogue evaluation framework [Walker *et al.*, 1997b]. This performance function is then used with reinforcement learning techniques, such as adaptive dynamic programming, Q-learning, temporal difference learning, and temporal difference Q-learning, to determine the optimal discourse strategies that ELVIS should use in different contexts. This thesis reports and compares the results of learning and describes how the particular reinforcement algorithms, the presence of local reward functions, and the system state space representation affect the efficiency and the outcome of the learning results. Learning optimal discourse strategies in a real-world spoken dialogue system requires the automated collection of system performance data during interactions with actual users, and this thesis concludes by describing extensions to the presented evaluation and learning techniques to automate online learning.

1.2 Overview of the Thesis

In Chapter 2, the spoken dialogue agent, ELVIS (**E**mai**L** **V**oice **I**nteractive **S**ystem), is presented. ELVIS supports access to email by telephone and possesses different discourse strategies for the distribution of initiative, summarizing email folders, and reading email messages. These implemented strategies are detailed in Chapter 2.

Chapter 3 describes the experimental design, the methods of data collection, the experimental results, and the derivation of a dialogue performance function from the experimental data.

Reinforcement learning is introduced in Chapter 4. Chapter 4 specifies and compares several reinforcement learning algorithms, discusses local rewards and state space representations, and presents the results of applying learning algorithms to the experimental data described in Chapter 3.

Chapter 5 discusses how the work described in this thesis relates to previous work in spoken dialogue systems, natural language content planning, adaptive agent

systems, and reinforcement learning.

Conclusions, future work, and methods for automating online learning in ELVIS can be found in Chapter 6.

Appendix A delineates the state space representation of ELVIS. Appendix B contains the email message folders used in the ELVIS experiments. The different task scenarios used in the experiments are presented in Appendix C. Appendix D presents the user survey used in the experiments. The data tables for points graphed in the thesis are contained in Appendix E.

Chapter 2

ELVIS, a Telephone-Based Email Agent

This chapter describes the spoken dialogue system, ELVIS (**E**mai**L** **V**oice **I**nteractive **S**ystem), a testbed for learning optimal discourse strategies in spoken dialogue systems [Walker *et al.*, 1998a, Walker *et al.*, 1998b, Walker *et al.*, 1997a]. The components of ELVIS, ELVIS' functionality, the different discourse strategies implemented in ELVIS, and ELVIS' state space representation are presented.

The sample conversation with ELVIS shown in Figure 2-1 is used throughout this chapter to illustrate the components of ELVIS' architecture.

2.1 ELVIS Components and Design

ELVIS is implemented as a distributed system using a general-purpose framework for spoken dialogue agents [Kamm *et al.*, 1997]. This section describes the components of that framework and the application-specific components of ELVIS. ELVIS' architecture is depicted in Figure 2-2.

- (1) ELVIS Hi, ELVIS here! Who is that?
- (2) User Susan
- (3) ELVIS Hi Susan! Please wait, I'll get your mail
- (4) ELVIS: You've got 5 new or unread messages and 1 old message Say "Read" or "Summarize," or say "Help" for more options.
- (5) User Summarize
- (6) ELVIS: In your top-level inbox, you are reading both old and new messages. There's 1 message from Michael, 2 messages from Kim, 1 message from Lee, and 2 messages from Dana Say "Read" or .
- (7) User (*interrupting*) Read
- (8) ELVIS: Select by sender, subject, or in order of arrival?
- (9) User. Sender
- (10) ELVIS Which sender?
- (11) User (*says nothing*)
- (12) ELVIS. The senders are Michael, Kim, Lee, and Dana Which sender?
- (13) User Kim
- (14) ELVIS: 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."
- (15) User: (*says nothing*)
- (16) ELVIS Say "Repeat" to repeat the last message, or say "Next" to read the next message in this folder Say "I'm done here" to exit this folder
- (17) User. Next.
- (18) ELVIS I didn't understand you Which message do you want to read?
- (19) User. Next
- (20) ELVIS. The next message is from Kim about "Meeting Today." "The meeting this morning is at 10.00 in 2D-516. See you there, Kim."
- (21) User. I'm done here.
- (22) ELVIS: You are back to your top-level inbox Say "Read" or "Summarize" or say "Help" for more options
- (23) User Help.
- (24) ELVIS: To access your messages, you can say "Read" or "Summarize." To switch the active messages, say "New Messages" or "All Messages" If you want me to repeat a message or a summary, say "Repeat" You can always say "Cancel" or "Help" Say "Goodbye ELVIS" to hang up
- (25) User New messages.
- (26) ELVIS Switching to new messages. Say "Read" or "Summarize" or say "Help" for more options
- (27) User Summarize
- (28) ELVIS In your top-level inbox, you are reading new messages There's 1 message from Michael, 1 message from Lee, and 1 messages from Dana. Say "Read" or "Summarize" or say "Help" for more options.
- (29) User Goodbye ELVIS
- (30) ELVIS Bye and don't be lonely!

Figure 2-1: A Conversation with ELVIS

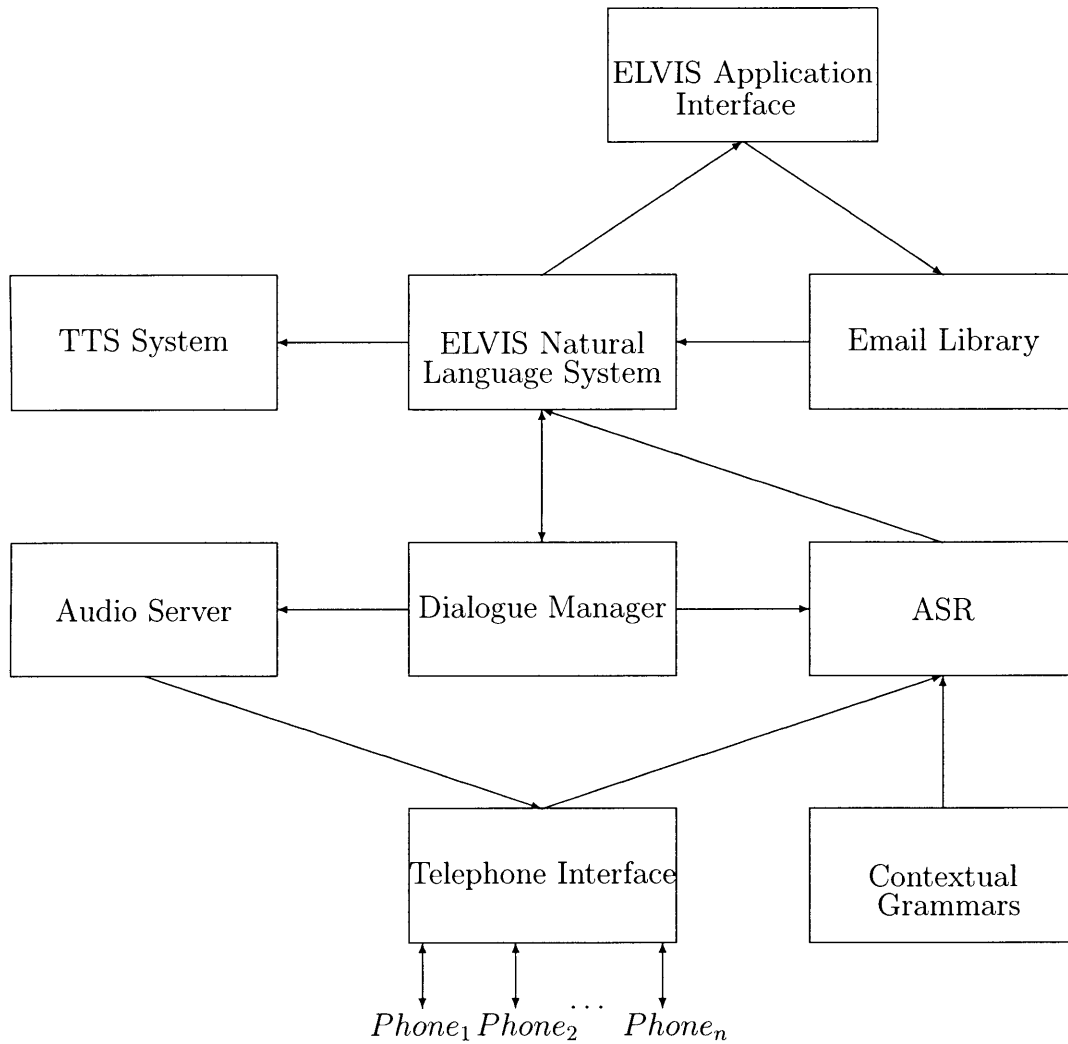


Figure 2-2: ELVIS Architecture

2.1.1 General-Purpose Framework

The general-purpose framework includes a speaker-independent automatic speech recognizer (ASR) that supports barge-in so users can interrupt an agent while it is speaking (as in utterances 6-7 of Figure 2-1). The framework also supplies an audio server for playing voice recordings and text-to-speech (TTS), a TTS system, an interface between the computer running an agent and a telephone network, a module for creating specific applications to run on the platform, a grammar tool, and a module for managing an application's dialogues.

The dialogue manager is based on a state machine, where a state consists of a descriptive name, a context-specific grammar with which to interpret a user's utterances, a transition function mapping conditions (such as what the user said or a particular discourse strategy) to states, an optional associated system function, and prompt templates. Each time a state is entered, the prompts are instantiated using the templates in conjunction with context-specific variables. There are four types of prompts:

- The initial prompt played when the user enters a state (such as utterance 14 in Figure 2-1)
- A help prompt played when the user requests help in that state (such as utterance 24 in Figure 2-1)
- Multiple timeout prompts that are played when the user does not say anything in that state (such as utterance 16 in Figure 2-1)
- Multiple reject prompts that are played when the ASR confidence is too low to continue without additional user input in that state (such as utterance 18 in Figure 2-1)

ELVIS' state space representation is described in greater detail in Section 2.4.

2.1.2 Application-Specific Component

The application-specific component consists of the ELVIS natural language system and the ELVIS application interface. The natural language system performs natural language interpretation and generation. The application interface handles some system bookkeeping, records prompts, and calls application-specific functions. This interface interacts with a library of general email functions, which include the processing of email folders, the creation of new email folders, the extraction of different fields from email messages, message searching, and message deletion.

2.2 ELVIS Functionality

This section details the functionality available in ELVIS. ELVIS' functionality was determined through a Wizard of Oz simulation of ELVIS.

| Category | Subcategory | Occurrences |
|---------------------------------|--------------------------|-------------|
| Clarification Request by Wizard | | 13 |
| Clarification Request by User | | 24 |
| Folder Action | Activation | 7 |
| | Update | 3 |
| Help | | 3 |
| Message Action | Activate | 2 |
| | Compose | 1 |
| | Delete | 13 |
| | Field Information | 5 |
| | Length Information | 4 |
| | Read | 67 |
| | Repeat Field Information | 2 |
| | Repeat Read | 2 |
| | Reply | 6 |
| | Save | 7 |
| | Search | 1 |
| | Skip | 23 |
| | Summarize | 2 |
| Message Reference | By Context | 70 |
| | By Folder | 2 |
| | By Number | 1 |
| | By Sender | 8 |
| | By Time Received | 1 |
| Search for Message | By Date | 1 |
| | By Folder | 3 |
| | By Sender | 2 |
| | By Subject | 2 |
| Summarization | By Folder | 10 |
| | By Person | 7 |
| | By Subject | 2 |
| | Order Received | 1 |
| System Status | | 4 |
| User Preferences | | 5 |

Table 2.1: Functional Analysis of Wizard of Oz Dialogues

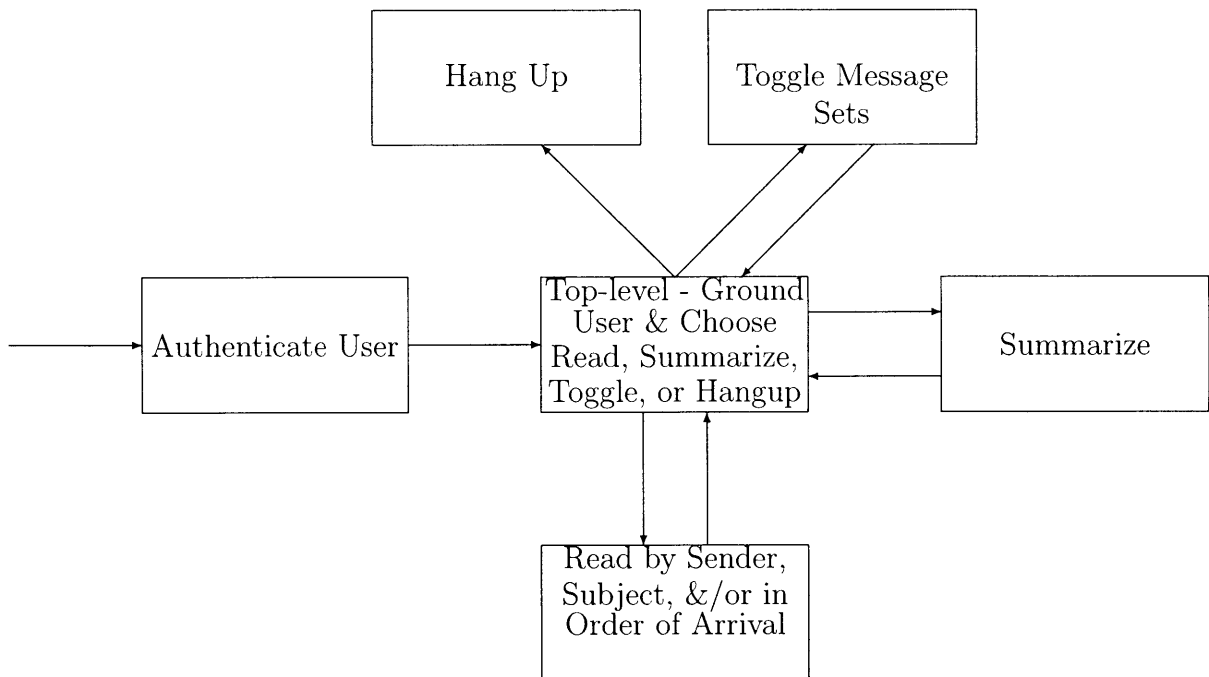


Figure 2-3: ELVIS Functionality

2.2.1 Wizard of Oz

For the Wizard of Oz (WOZ) study, six email users called a person acting as the Wizard whenever they wanted to check their email over a three-hour period. The 15 conversations that transpired during that period were recorded, transcribed, and analyzed for the sorts of functions the users had asked the Wizard to perform. The results of this analysis are depicted in Table 2.1.

2.2.2 System Functionality

Based on the WOZ results, we concluded that it was imperative to provide multiple ways to refer to messages, the ability to read and repeat messages, context-specific help, folder summarization capabilities, and system feedback.

The functionality provided to a user of ELVIS is depicted in Figure 2-3. The user calls ELVIS by telephone, whereupon ELVIS authenticates the user (utterances 1-2 in

Figure 2-1). ELVIS then retrieves the user's email (the inbox folder) and informs the user of the number of new, unread, and old messages the user currently has (utterance 4 in Figure 2-1). In ELVIS' top-level state, the user can request a summary of the email messages in the inbox, select messages to read by sender, subject, or in order of arrival, toggle between hearing about all the messages in the current folder or only the new and unread messages in that folder, or hang up. In Figure 2-1, summarization occurs in utterances 5-6 and 27-28, message selection for reading occurs in utterances 7-21, toggling occurs in utterances 25-26, and hanging up occurs in utterances 29-30. When the user requests that ELVIS read a particular set of messages, ELVIS creates a *virtual folder* (vfolder) to contain those messages and makes that vfolder the current folder. The user can choose to hear some subset of that vfolder's messages and can then reset the current folder to be the top-level inbox (utterances 14-21 in Figure 2-1). After the user hears a summary, reads messages and exits a vfolder, or toggles between different sets of messages, the user returns to ELVIS' top-level state.

2.3 Different Discourse Strategies

Dialogue agents face the problem of deciding *what* information to communicate to a user and *how* and *when* to communicate it. There exist many potential strategies for distributing initiative, summarizing messages in an email folder, and reading messages. This section describes two different initiative strategies and six different presentation strategies (three different summarization strategies and three different read strategies) that were implemented in ELVIS. Each discourse strategy variation is implemented as a combination of the prompts that are played to the user and the state transitions that ELVIS makes.

All of the claims enumerated in this section are tested and the results are reported and compared in Chapters 3 and 4.

(D1) ELVIS: You have 5 new or unread messages and 1 old message in your inbox. Say “Read,” or “Summarize,” or say “Help” for more options.
User: Read.
ELVIS: Select by Sender, Subject, or in Order of Arrival?
User: Sender.
ELVIS: Which Sender?
User: Kim.

Figure 2-4: Dialogue 1: System-Initiative Initiative Strategy

2.3.1 Initiative Strategies

In a conversation involving two participants, both interlocutors assume varying degrees of *initiative*; that is, they each control the conversation by directing questions, issuing commands, restricting potential responses, and controlling the discussion topic in differing proportions. The degree of initiative that the participants assume can change multiple times over the course of their conversation.

Two initiative strategies were implemented in ELVIS, the **System-Initiative Initiative Strategy** and the **Mixed-Initiative Initiative Strategy**. For illustrative purposes, we assume that the user’s goal is to read his messages from Kim.

System-Initiative Strategy

The System-Initiative (SI) strategy explicitly prompts users at each stage of the dialogue with directive prompts of what to say [Kamm, 1994]. Using this initiative strategy, ELVIS acquires information in small increments and constrains the user to very short utterances. Dialogue 1 (D1) in Figure 2-4 illustrates the use of the SI strategy in ELVIS.

Prior work has claimed that the performance of an agent using the SI strategy will be good because users’ utterances are explicitly constrained to short phrases, which current ASR technology can handle effectively [Marcus *et al.*, 1996, Potjer *et al.*, 1996, Danieli and Gerbino, 1995].

(D2) ELVIS: I've got your mail.
User: Read me my messages from Kim.

Figure 2-5: Dialogue 2: Mixed-Initiative Initiative Strategy

Mixed-Initiative Strategy

When ELVIS uses the Mixed-Initiative (MI) strategy, ELVIS does not volunteer information and assumes that users knows what to say. The MI strategy is designed so that users can choose to communicate a complete ELVIS function and its arguments in a single utterance. Dialogue 2 (D2) in Figure 2-5 illustrates the use of the MI strategy in ELVIS. The MI strategy is more flexible than the SI strategy, but users need more knowledge about what they can say to use it successfully. Prior work has claimed that MI is the better initiative strategy to use in advice-giving and diagnostic dialogue domains [Smith and Hipp, 1994, Walker and Whittaker, 1990].

The MI strategy may cause higher ASR error rates than the SI strategy because the MI strategy requires more complex grammars than the SI strategy does. This can lead to longer repair subdialogues for the MI strategy than for the SI strategy.

2.3.2 Summarization Strategies

When a user requests a summary of their messages from ELVIS, there are many ways that ELVIS can present a valid summary. Three of these possibilities are implemented as the **Summarize-System** Summarization Strategy, the **Summarize-Choice** Summarization Strategy, and the **Summarize-Both** Summarization Strategy.

Summarize-System Strategy

The Summarize-System strategy, illustrated in Dialogue 3 (D3) in Figure 2-6, has ELVIS summarize by subject or by sender based on context. For instance, when users are in their inbox folder, ELVIS will summarize by sender, but when a user is

(D3) ELVIS: In your top-level inbox, there's 1 message from Kim, 2 messages from Noah, 1 message from Michael, and 1 message from Owen.

Figure 2-6: Dialogue 3: Summarize-System Summarization Strategy

(D4) ELVIS: Summarize by subject, by sender, or both?

User: Subject.

ELVIS: 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."

Figure 2-7: Dialogue 4: Summarize-Choice Summarization Strategy

situated in a folder containing messages from Kim, for example, ELVIS will summarize by subject, as a summary by sender would provide no new information.

Summarize-Choice Strategy

The Summarize-Choice strategy, illustrated in Dialogue 4 (D4) in Figure 2-7, has ELVIS ask the user to specify which attributes to summarize by: subject, sender, or both subject and sender.

Summarize-Both Strategy

When using the Summarize-Both strategy, illustrated in Dialogue 5 (D5) in Figure 2-8, ELVIS uses the sender and subject attributes together in generating a summary.

2.3.3 Read Strategies

When a user requests that ELVIS read a particular set of messages and that set contains more than one message, ELVIS can validly present the set and read from it in multiple ways. Three of these possibilities are implemented as the **Read-First** Read Strategy, the **Read-Summarize-Only** Read Strategy, and the **Read-Choice-Prompt** Read strategy. For illustrative purposes, we assume that a user has just requested to read his messages from Kim.

(D5) ELVIS: 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."

Figure 2-8: Dialogue 5: Summarize-Both Summarization Strategy

(D6) ELVIS: 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 to lunch. Kim."

Figure 2-9: Dialogue 6: Read-First Read Strategy

Read-First Strategy

When ELVIS chooses the Read-First strategy, as portrayed in Dialogue 6 (D6) in Figure 2-9, ELVIS summarizes all of the messages from Kim and then takes the initiative to read the first one.

Read-Summarize-Only Strategy

The Read-Summarize-Only strategy, illustrated in Dialogue 7 (D7) in Figure 2-10, has ELVIS provide information that allows users to take the initiative to refine their selection criteria for reading.

Read-Choice-Prompt Strategy

When using the Read-Choice-Prompt strategy, illustrated in Dialogue 8 (D8) in Figure 2-11, ELVIS explicitly informs users of what to say in order to refine their selection criteria for reading.

2.4 ELVIS State Space Representation

This section describes the state space representation that ELVIS uses. ELVIS' state space and state transitions are provided in greater detail in Appendix A. As de-

(D7) ELVIS: In the messages from Kim, there's 1 message about "Interviewing Antonio" and 1 message about "Meeting Today."

Figure 2-10: Dialogue 7: Read-Summarize-Only Read Strategy

(D8) ELVIS: 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."

Figure 2-11: Dialogue 8: Read-Choice-Prompt Read Strategy

scribed earlier in Section 2.1.1, a state consists of a descriptive name, a set of dialogue prompts, a grammar with which to interpret a user's utterance, an optional associated system function, and a state transition function. After ELVIS speaks in each state, ELVIS senses the environment (by performing ASR) and invokes the state's associated system function, if one is specified. The associated system functions perform system bookkeeping, record prompts for future utterances, and use the email library. Figure 2-12 shows an annotated version of utterances 14-21 in Figure 2-1, where a transition path through states is shown.

There is no state differentiation between users first entering a state S_i , timing out in S_i , receiving help in S_i , or having their utterances rejected in S_i . ELVIS perceives each of these encapsulated states only as the single state S_i , as utterances 14-18 of Figure 2-12 illustrate.

Other distinct states are collapsed into single states to achieve efficient system performance. The state space representation provided by the dialogue manager is entirely dissociated from task progress representation since a general-purpose, task-independent platform (which includes the dialogue manager) is used to implement ELVIS. Multiple task states are collapsed into single dialogue states. For example, ELVIS can summarize messages, toggle between sets of messages, and ground the user in the single top-level dialogue state, and ELVIS' utterances 4, 6, 22, 24, 26, and 28 in Figure 2-1 all occur in dialogue state **TopLevelUserOptionsSI**.

Competing discourse strategies are implemented in ELVIS as distinct state transi-

(ELVIS enters function `BuildVfolderForSenderSI` to build a vfolder for messages from Kim)

(Since vfolder contains more than one message and the read strategy is `Read-First`, ELVIS enters state `ReadCurrentVfolderSI-first` and utters the entrance prompt)

(14) ELVIS: 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."

(15) User: *(says nothing)*

(ELVIS plays timeout prompt for state `ReadCurrentVfolderSI-first` because ELVIS has not heard any user response)

(16) ELVIS: Say "Repeat" to repeat the last message, or say "Next" to read the next message in this folder. Say "I'm done here" to exit this folder.

(17) User: Next.

(ELVIS plays reject prompt for state `ReadCurrentVfolderSI-first` because ELVIS has not been able to interpret the user's response with enough confidence to continue)

(18) ELVIS: I didn't understand you. Which message do you want to read?

(19) User: Next.

(ELVIS understands that the user wants to read the next message and enters function `ReadMessageInLoopSI` to record the next message into a prompt and update the current message pointer)

(ELVIS enters state `ReadCurrentVfolderSI` and utters the entrance prompt)

(20) ELVIS: The next message is from Kim about "Meeting Today." "The meeting this morning is at 10:00 in 2D-516. See you there, Kim."

(21) User: I'm done here.

Figure 2-12: Annotation of Utterances 14-21 in Figure 2-1

tions from their activation points. The state transition paths that competing strategies follow always converge at some later dialogue state.

The nature of the state space representation for ELVIS becomes important in learning and Section 4.3 returns to this topic. The next chapter discusses the experiments conducted using discourse strategy-based variations of ELVIS.

Chapter 3

Experimenting with ELVIS

This chapter recounts the experiments with the different discourse strategy-based versions of ELVIS. The data collection methods and the experimental scenarios are detailed, two sets of conducted experiments and the experiments' results are described, and a performance function is derived for ELVIS from the experimental data.

3.1 Data Collection Methods

Experimental dialogues were collected via two different experiments in which subjects interacted with ELVIS to complete three typical tasks that required them to access email messages in three different email inbox folders. The email folders are provided in Appendix B. In the second experiment, before performing the three tasks, subjects participated in a tutorial task dialogue to acquaint the subjects with ELVIS by guiding the subjects step-by-step through a typical task. Subjects were given instructions on a set of web pages, with one page for each experimental task.

36 subjects, AT&T summer interns, participated in the first experiment, which varied initiative strategies. 18 subjects used the SI version of ELVIS while the other 18 subjects used the MI version of ELVIS. The first experiment resulted in a corpus of 108 dialogues testing the initiative strategies.

31 subjects, MIT students, participated in the second experiment, which varied summarization and read strategies in an SI version of ELVIS. To have adequate data

(**Dialogue Quality**) Barge-Ins, Helps, Rejects, Timeouts, Mean Recog
 (**Dialogue Efficiency**) Elapsed Time, System Turns, User Turns
 (**Task Success**) Kappa, Completed
 (**User Satisfaction**) TTS Performance, ASR Performance, Task Ease, Interaction
 Pace, User Expertise, System Response, Expected Behavior, Future Use

Figure 3-1: Collected Experimental Measures

(**Subtask 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 where and when the meeting is. Find out the **Meeting Time** and the **Meeting Place**.

Figure 3-2: Typical Subtask for the ELVIS Experiments

for learning, ELVIS had to explore the space of strategy combinations and collect enough samples of each combination. Each subject interacted with 3 different versions of a parameterized ELVIS for each of the three tasks. This second experiment yielded a corpus of 124 dialogues testing the summarization and read strategies.

This section describes the task scenarios and the three different methods of data collected: dialogue recordings, dialogue manager logs, and user survey data. The objective and subjective measures shown in Figure 3-1 are collected in the ELVIS experiments and are defined in this section.

3.1.1 Task Scenarios

Each of the subjects performed all of their tasks in sequence, with each task consisting of two subtasks. In each subtask, ELVIS and the subject exchange information about criteria for message selection and for information extraction within the message body. Figure 3-2 displays a typical subtask. This subtask can be represented in terms of the attribute-value matrix (AVM) in Table 3.1. All attribute-values must be exchanged to successfully complete a subtask [Walker *et al.*, 1997b]. The AVM representation

| Attribute | Actual Value |
|--------------------|---------------|
| Selection Criteria | Kim ∨ Meeting |
| Meeting Time | 10:30 |
| Meeting Place | 2D516 |

Table 3.1: Attribute-Value Matrix Instantiation for Subtask in Figure 3-2

for each subtask is independent of the particular discourse strategies that ELVIS uses. The complete set of subtasks and their corresponding AVMs can be found in Appendix C.

3.1.2 Dialogue Recordings

All of the experimental dialogues were recorded. These recordings were used to match the subject’s utterances to ELVIS’ perceptions of those utterances (the ASR results), deriving performance measures for speech recognition. The interaction durations (**Elapsed Time**) and how often subjects interrupted, or *barged in* on, ELVIS’ utterances (**Barge-Ins**) were also calculated from the recordings.

3.1.3 Dialogue Manager Logs

ELVIS logged the state transitions that the subjects’ dialogues followed, the discourse strategies that ELVIS chose, the string perceived by the ASR for each subject utterance (**Recognized String**), the number of timeout prompts (**Timeouts**), the number of ASR rejections (**Rejects**), and when the subject requested help (**Helps**). The number of dialogue turns that the subject takes (**User Turns**) and that ELVIS takes (**System Turns**) can be computed from these logs. **Recognized String** is used in conjunction with the dialogue recordings to calculate a concept accuracy score for each utterance, ranging from 0 (entirely inaccurate) to 1 (entirely accurate).¹ Mean concept accuracy can then be calculated over each dialogue and can be used as a

¹For 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 subject said, “Read,” concept accuracy would be .5.

mean recognition score (**Mean Recog**) for each dialogue.

3.1.4 User Surveys

Subjects filled out web page forms to measure task success and user satisfaction. To assess task success, subjects indicated whether they believed they had completed the specified task (**Completed**) and filled in a task-specific AVM with the information that they acquired from interacting with ELVIS (such as **Meeting Time** and **Meeting Place** in Table 3.1). Task success can be computed from this AVM by using the **Kappa** statistic to compare the information in the subject-specified AVM with an AVM key as in Table 3.1 [Walker *et al.*, 1997b]. **Kappa** is defined as:

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)} \quad (3.1)$$

In Equation 3.1, $P(A)$ is the proportion of times that the AVMs for the set of actual dialogues agrees with the AVMs for the subtask keys and $P(E)$ is the proportion of times that the AVMs for the dialogues and the keys are expected to agree by chance. If $P(E)$ is unknown, it can be estimated from the distribution of the values in the keys as described in [Walker *et al.*, 1997b]. When all task information items are successfully exchanged, then agreement is perfect and $\kappa=1$. When agreement is only at chance, $\kappa=0$. κ accounts for inherent task complexity by correcting for agreement expected by chance and is superior to other measures of success such as transaction success [Danieli and Gerbino, 1995] and percent agreement [Gale *et al.*, 1992]. κ also provides a measure for comparing different agents performing different tasks.

User satisfaction is calculated from the subjects' responses to survey questions about ELVIS' performance [Walker *et al.*, 1998a]. The multiple-choice questions asked about **TTS Performance**, **ASR Performance**, **Task Ease**, **Interaction Pace**, **User Expertise**, **System Response**, **Expected Behavior**, **Comparable Interface**, and **Future Use**. The possible responses to most questions ranged over *{almost never, rarely, sometimes, often, almost always}* or an equivalent range. These responses were each mapped to integers between 1 and 5. Responses to *{yes, maybe,*

no} questions were mapped to integers between 1 and 3 and responses to *{yes, no}* questions were mapped to integers in *{1,2}*. Each question emphasized the subject’s experience with ELVIS in the current conversation, with the hope that satisfaction measures would indicate perceptions specific to each conversation, rather than reflecting an overall evaluation of ELVIS over the series of tasks. For each dialogue, a cumulative satisfaction score (**User Satisfaction**) was computed by summing the scores for each survey question, resulting in a value between 0 and 43. The complete user survey can be found in Appendix D.

3.2 Experimental Results

This section summarizes the results of the initiative strategy and presentation strategies experiments.

3.2.1 Initiative Strategy Experimental Results

| Measure | System (SI) | Mixed (MI) |
|-------------------|-------------|------------|
| Kappa | .9 | .91 |
| Completed | .83 | .78 |
| User Turns | 25.94 | 17.59 |
| System Turns | 28.18 | 21.74 |
| Elapsed Time | 328.59 s | 289.43 s |
| Mean Recog | .88 | .72 |
| Timeouts | 2.24 | 4.15 |
| Helps | .7 | .94 |
| Barge-Ins | 5.2 | .35 |
| Rejects | .98 | 1.67 |
| User Satisfaction | 26.6 | 23.7 |

Table 3.2: Performance Measure Means per Dialogue for Initiative Strategies

The initiative strategy experiment shows that subjects could complete the tasks with both versions of ELVIS. A comparison of the different initiative strategy measures appears in Table 3.2. More detailed discussion of the results of this experiment

can be found in [Walker *et al.*, 1998a].

3.2.2 Presentation Strategies Experimental Results

| Measure | Read-First Summarize-System | Read-Summarize-Only Summarize-Choice | Read-Choice-Prompt Summarize-Both |
|-------------------|-----------------------------|--------------------------------------|-----------------------------------|
| Kappa | .81 | .74 | .84 |
| Completed | .89 | .77 | .84 |
| User Turns | 18.8 | 19.51 | 20.5 |
| System Turns | 22.27 | 22.05 | 23.64 |
| Elapsed Time | 344.53 s | 311.9 s | 370.03 s |
| Mean Recog | .88 | .83 | .88 |
| Timeouts | 3.47 | 2.54 | 3.14 |
| Helps | .63 | .62 | .83 |
| Barge-Ins | 2.69 | 2.69 | 2.11 |
| Rejects | .84 | .97 | .78 |
| User Satisfaction | 28.86 | 27.31 | 28.42 |

Table 3.3: Performance Measure Means per Dialogue for Summarization and Read Strategies

Compared with the initiative strategy experiment, κ values show that the presentation strategies experiment was more difficult for subjects to complete. A comparison of the collected measures for the different presentation strategies appears in Table 3.3.

3.3 Deriving a Performance Function

The PARADISE evaluation framework [Walker *et al.*, 1997b] can be used to empirically identify the important performance factors in a spoken dialogue agent and can additively and linearly combine those factors to provide a performance function for the agent. The structure of objectives in Figure 3-3 provides a basis for the factors measured in estimating a performance function. The complete list of surveyed measures for ELVIS is found in Figure 3-1. Performance for any dialogue D is defined by the following equation:

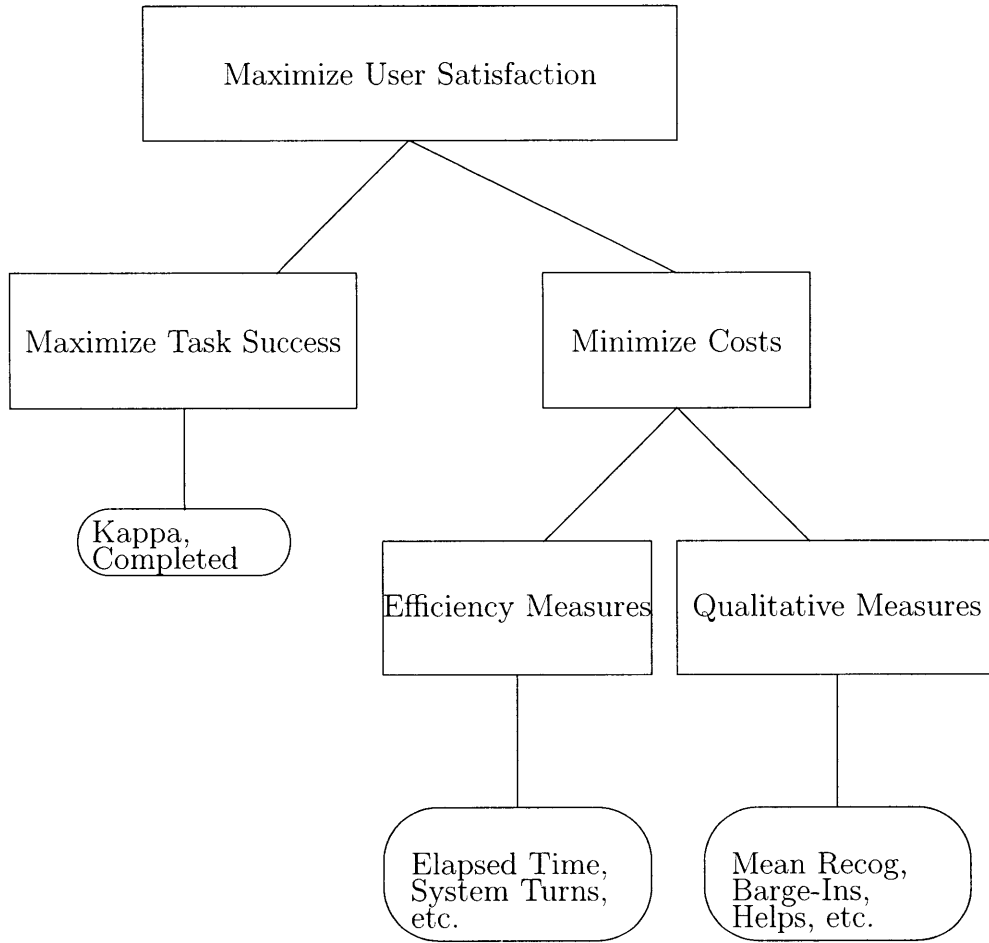


Figure 3-3: PARADISE’s Structure of Objectives for Spoken Dialogue Performance

$$Performance = (\alpha * \mathcal{N}(\kappa)) - \sum_{i=1}^n w_i * \mathcal{N}(c_i) \quad (3.2)$$

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.*, 1997b, Cohen, 1995]. The Z score normalization function ensures that the magnitude of each weight reflects the magnitude of the contribution of its associated factor to the agent’s performance. The performance function is derived through multivariate linear regression with **User Satisfaction** as the dependent variable and all of the other experimental measures as independent variables [Walker *et al.*, 1997b].

Various performance functions over many different subsets of the experimental

data can be derived. The performance function that is derived from the largest and most varied subset of experimental data is most desirable because it is the most general performance function derivable from the experimental data. Therefore, the performance function is derived from all of the data in both the initiative strategy and the presentation strategies experiments.

Using all of the data from the first and second experiments, an initial regression over the measures in Tables 3.2 and 3.3 suggests that **Mean Recog**, **Elapsed Time**, **System Turns**, **Rejects**, **Barge-Ins**, and **Completed** are all significant contributors to **User Satisfaction**. A second regression over these measures, normalized, results in the following equation:

$$\begin{aligned}
 \textit{Performance} = & .54 * \mathcal{N}(\textit{MeanRecog}) - .34 * \mathcal{N}(\textit{ElapsedTime}) + \\
 & .26 * \mathcal{N}(\textit{SystemTurns}) + .2 * \mathcal{N}(\textit{Rejects}) - \\
 & .1 * \mathcal{N}(\textit{Barge - Ins}) + .25 * \mathcal{N}(\textit{Completed}) \quad (3.3)
 \end{aligned}$$

Mean Recog (t=7.22, p=.0001), **Elapsed Time** (t=-3, p=.003), **System Turns** (t=2.04, p=.04), **Rejects** (t=2.66, p=.008), **Barge-Ins** (t=-1.69, p=.09), and **Completed** (t=4.3, p=.0001) are significant predictors, accounting for 37% of the variance in R-Squared (F(6,225)=22.32, p<.0001). The magnitude of the coefficients in Equation 3.3 demonstrates that the performance of the speech recognizer (**Mean Recog**) is the most important predictor, followed by efficiency (**Elapsed Time**), the number of system turns (**System Turns**), the subjects' perceptions of task success (**Completed**), the number of ASR rejections (**Rejects**), and the number of times that the subjects interrupted ELVIS (**Barge-Ins**).

The performance function includes the nonintuitive coefficients of .26 for **System Turns** and .2 for **Rejects**. One would expect that dialogue efficiency suffers with increasing numbers of system turns and that dialogue quality decreases with the number of ASR rejections. There may be several reasons for these coefficients, including possible tradeoffs with other factors in the model or possibly that rejections function more like help messages in this particular application. Exploring why these factors

show up as they do is a topic for future work.

The next chapter shows how to use the derived performance function to compute the utility of the final state of each dialogue and to learn the utilities of competing discourse strategies.

Chapter 4

Learning Optimal Discourse Strategies

This chapter introduces reinforcement learning, discusses how to assign local reinforcement, emphasizes the importance of the state space representation in learning, discusses the learned state transition model, and details the results of learning.

4.1 Reinforcement Learning

There are many different forms of machine learning, including inductive learning, learning in neural networks (such as back-propagation and Bayesian learning), and reinforcement learning [Russell and Norvig, 1995]. Inductive learning requires “good” examples to learn successfully, so it is not applicable to learning optimal discourse strategies; prior to actual interactions with ELVIS, which discourse strategies are optimal is unknown. ELVIS’ representation is not directly suitable to learning in a neural network. Reinforcement learning is chosen to learn optimal discourse strategies because a spoken dialogue system can learn a predictive model of its environment by trying random examples and receiving unsupervised feedback.

In *reinforcement learning*, an agent obtains percepts in its environment, maps some subset of them to positive or negative rewards, and then decides which action to take [Russell and Norvig, 1995]. The agent is never told which actions are the

“right” ones but these rewards can be used to learn a successful *policy*, a complete mapping from environmental states to actions, for the agent. This section describes reinforcement learning, some of its algorithms, and how to learn when the environment is represented as a partially observable Markov decision problem.

4.1.1 Overview

One can derive a plan to choose among different strategies by associating a utility U with each strategy (action) choice and by adhering to the **Maximum Expected Utility Principle** [Russell and Norvig, 1995, Keeney and Raiffa, 1976]:

An optimal action is one that maximizes the expected utility of outcome states.

To act optimally when in a dialogue state S_i , an agent chooses the action $a \in A$, the set of actions available in S_i , that maximizes $U(S_i)$. Reinforcement learning provides methods for deriving the utility values $U(S_i)$ for each S_i from the utilities of S_i 's successor states [Barto *et al.*, 1995, Sutton, 1991, Watkins, 1989, Bellman, 1957]. If an agent can determine the utility value for its final state, the utilities of all of the preceding states can be calculated. Section 3.3 showed how to derive a performance function for ELVIS and use it to calculate dialogue performance. The dialogue performance value for a dialogue D is used as the utility value of the final state of D . The performance function shown in Equation 3.3 that was derived from the data of both experiments is used.

4.1.2 Algorithms

This section describes four reinforcement learning algorithms: adaptive dynamic programming, Q-learning, temporal difference learning, and temporal difference Q-learning [Russell and Norvig, 1995]. These algorithms can calculate optimal discourse policies for *Markov decision problems* (MDPs), accessible, stochastic environments with a known transition model. This section concludes with a discussion of which of

the presented algorithms are most appropriate for learning optimal discourse strategies in a spoken dialogue system that converses with actual users.

Adaptive Dynamic Programming

Adaptive dynamic programming (ADP) is a learning algorithm that solves for a state space's utility values with a dynamic programming algorithm. The utilities can be computed by solving the constraint equation

$$U(S_i) = R(S_i) + \max_a \sum_j M_{ij}^a U(S_j) \quad (4.1)$$

$R(S_i)$ is a reward associated with being in state S_i , a is an action from a finite set A that is available in state S_i , and M_{ij}^a is the probability of immediately progressing to state S_j if action a is chosen in state S_i . M_{ij}^a can be approximated by calculating probabilities from observed state transitions and is discussed in more detail in Section 4.4. The utility values can be estimated to within a desired threshold through value iteration, which updates $U(S_i)$ from the values of its successor states, and Equation 4.1 becomes:

$$U_{n+1}(S_i) = R(S_i) + \max_a \sum_j M_{ij}^a U_n(S_j) \quad (4.2)$$

Q-Learning

Q-learning is a variation of ADP where utilities are learned for state-action pairs instead of learning utilities solely for states. The constraint equation for Q-learning is

$$U(a, S_i) = R(S_i) + \sum_j M_{ij}^a \max_{a'} U(a', S_j) \quad (4.3)$$

Again, the utility values can be estimated to within a chosen threshold through value iteration, yielding the following equation from Equation 4.3:

$$U_{n+1}(a, S_i) = R(S_i) + \sum_j M_{ij}^a \max_{a'} U_n(a', S_j) \quad (4.4)$$

Q-learning seems especially suited for learning the most successful actions in a particular state for a system and the goal of this thesis is to learn the optimal discourse strategies in ELVIS dialogue states where there is choice among differing strategies.

Temporal Difference Learning

It is possible to approximate the utility constraint equations without the use of a transition model and without solving the equations for all possible states by using observed transitions to adjust the utilities of the observed states so that they agree with the constraint equations. This method is called *temporal difference learning* (TD learning) and uses the following updating rule whenever a transition from state S_i to state S_j is observed:

$$U(S_i) = U(S_i) + \alpha(R(S_i) + U(S_j) - U(S_i)) \quad (4.5)$$

α is the learning rate parameter and if α is a function that decreases as the number of times a state has been visited increases, $U(S_i)$ is guaranteed convergence to the correct value [Dayan, 1992].

Temporal Difference Q-Learning

Through minor variations to Equation 4.5, analogous to the transformation of ADP to Q-learning, temporal difference can be used to learn the utilities of state-action pairs. Whenever a transition from state S_i to state S_j via action a is observed, the following update equation can be used:

$$U(a, S_i) = U(a, S_i) + \alpha(R(S_i) + \max_{a'} U(a', S_j) - U(a, S_i)) \quad (4.6)$$

This thesis learns optimal discourse strategies using data from human interactions, unlike previous work which has used simulations [Levin and Pieraccini, 1997]. The

model-based reinforcement learning algorithms of ADP and Q-learning propagate utility values throughout the state space at each learning iteration. The observance-based reinforcement learning algorithms of TD learning and TD Q-learning only update utility values for state transitions as they are observed; there is no propagation throughout the state space at each update.

Since the observance-based learning algorithms update state utilities much less often than the model-based learning algorithms, the observance-based learning algorithms need more data than the model-based learning algorithms to converge. The convergence speed differences are especially important when learning over human data instead of over simulations, since it is not straightforward to collect thousands of human-agent conversations. Because of the convergence speed differences and the limited amount of collected human data, this thesis uses the model-based learning algorithms to learn optimal discourse strategies.

4.1.3 Partially Observable Markov Decision Problems

Partially observable Markov decision problems (POMDPs) occur when an environment is inaccessible [Russell and Norvig, 1995], meaning that the perceptions of an agent do not provide accurate information or enough information to determine the state or the associated transition probabilities. ELVIS is inherently a POMDP because its percepts are not entirely reliable, as ELVIS can and does sometimes mishear utterances. Some nonessential state space characteristics that make ELVIS a POMDP are detailed in Section 4.3 and eliminated.

Methods used for MDPs are not directly applicable to POMDPs, so the learning algorithms presented in Section 4.1.2 do not seem to be useful for learning optimal discourse strategies in ELVIS. Probability distributions over the possible states given all previous percepts are calculated for POMDPs and are used for policy decisions. The standard method for solving a POMDP is to construct a new MDP in which the probability distributions play the role of state variables. This new state space has real-valued probabilities, rendering it infinite. Most practical solutions to POMDPs tend to be approximate [Russell and Norvig, 1995].

The partially observable nature of ELVIS is incorporated into the standard learning algorithms for MDPs by deriving a performance function that, among other things, accounts for ELVIS' degree of success in accurately perceiving the environment. The performance function is a linear composition of the measures that affect user satisfaction, including ELVIS' ASR performance (**Mean Recog**) and the number of user utterances that are rejected (**Rejects**). By assimilating this information into learning, the standard MDP reinforcement learning algorithms can be used to learn optimal discourse strategies in ELVIS. The incorporation of the accuracy of the system's perceptions into learning also minimizes the differences between ELVIS' internal representations and the real world. This inclusion avoids the confounding of learning results that arises from the difference between the actual state of the real world and a system's internal representation of the world [Whitehead and Ballard, 1991].

4.2 Local Reinforcement

Agents can learn more accurately and with less experience with local reinforcement signals [Kaelbling *et al.*, 1996, Mataric, 1994]. Local rewards cause learning results to be more accurate by reflecting local performance maxima and minima that affect overall dialogue performance. Using a local reward function accelerates the learning process because utility values converge more quickly by rewarding agents for taking steps toward the goal instead of just rewarding agents for achieving the final goal. This section describes different approaches to allocating rewards at the granularity of the dialogue state.

4.2.1 Qualitative Approach

A local reward function can be qualitatively defined by assigning numerical values to different features deemed important to the performance of an agent. For example, in ELVIS, completing a set of desired tasks is integral to user satisfaction and, therefore, to the performance of ELVIS. ELVIS could be assigned increasing positive degrees of reward for having no subtasks completed, having a single subtask completed, and

having both subtasks completed. In another example, states where a user’s utterances are rejected by ELVIS can receive smaller rewards than states where user’s utterances are not rejected.

4.2.2 Using the Performance Function

According to discourse theory, dialogue has structure above the level of the turn and below the level of the entire dialogue [Mann and Thompson, 1987, Grosz and Sidner, 1986, Power, 1979, Power, 1974]. In a task-oriented dialogue, there is structure at the level of the subtasks [Isard and Carletta, 1995]. This provides a basis for using the dialogue performance function below the level of the whole dialogue in order to assign different local rewards to each dialogue state.

The performance function in Equation 3.3 can be used at the state level to assign a reward to each state S_i , which can then be used as the value of $R(S_i)$ in computing Equations 4.1, 4.3, 4.5, and 4.6 for the learning algorithms discussed in Section 4.1.2.

To easily compute $R(S_i)$ for each S_i , a different model is fit to the experimental data that utilizes metrics which are easily extracted from the state-by-state information in the system logs. For example, it is not possible to non-manually calculate **Elapsed Time** for each state occurrence in the distributed system that comprises ELVIS. All of the dialogue quality measures found in Figure 3-1, the efficiency measures of **System Turns** and **User Turns**, and the task success measure for the degree of task completion can be measured on a state-by-state basis. An initial regression over these state performance measures reveals that **Mean Recog**, **System Turns**, **Rejects**, and **Completed** are significant. A second regression results in the following equation:

$$\begin{aligned}
 Performance(S_i) = & .55 * \mathcal{N}(MeanRecog_i) - .09 * \mathcal{N}(SystemTurns_i) + \\
 & .23 * \mathcal{N}(Rejects_i) + .25 * \mathcal{N}(Completed_i) \quad (4.7)
 \end{aligned}$$

Mean Recog (t=7.32, p=.0001), **System Turns** (t=-1.52, p=.1), **Rejects** (t=3.03, p=.003), and **Completed** (t=4.24, p=.0001) are significant predictors, accounting

for 35% of the variance in R-Squared ($F(4,227)=30.11$, $p<.0001$). $MeanRecog_i$ is the recognition score for the occurrence of state S_i . $SystemTurns_i$ is the number of system turns taken in S_i , including helps, rejections, and timeouts, and $Rejects_i$ is the number of rejections that occur in S_i . $Completed_i$ is $\frac{1}{3}$ if no subtasks have been completed, $\frac{2}{3}$ if 1 subtask has been completed, and 1 if both subtasks have been completed by the occurrence of S_i .

$Performance(S_i)$ is computed for each occurrence of each S_i using Equation 4.7 and $R(S_i)$ is then calculated with the following equation that takes the mean over all the values of performance for occurrences of S_i :

$$R(S_i) = \overline{Performance(S_i)} \quad (4.8)$$

4.3 State Space Representation

The state space representation used in reinforcement learning affects the outcomes of the learning [Koenig and Simmons, 1996, Lin, 1992, Whitehead and Ballard, 1991]. [Lin, 1992] points out that reinforcement learning is representation-dependent and history-insensitive. Each bit of history relevant to learning must be represented as a state variable. [Koenig and Simmons, 1996] report that the complexity of running learning algorithms depends on the representation chosen. By representation, they refer only to the initial utility values and the local reward function. This section addresses other aspects of the state space representation that affect learning results. For accurate learning, ELVIS' state space representation as described in Section 2.4 needs to be transformed to avoid losing relevant information and to reflect the tasks performed by ELVIS and its users. This section describes how to convert the ELVIS state space to learn reliably.

4.3.1 Separating States by Prompt Type

As described in Section 2.4, there is no system distinction by prompt type: initially entering a state S_i , requesting help in S_i , timing out in S_i , and facing an ASR rejection

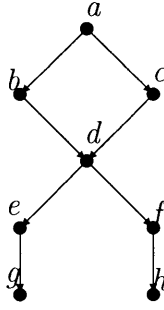


Figure 4-1: A Graph with a Reentrant Node

in S_i each count as an instance of S_i . For example, **ReadCurrentVfolderSI-first** is the state in which utterances 14, 16, and 18 in Figure 2-12 all occur even though utterance 14 is the initial utterance, utterance 16 is a timeout utterance, and utterance 18 is a rejection utterance. These circumstances are very different and to make the state space more Markovian, the state space is transformed to differentiate between the four situations by using a state variable in each S_i to represent the prompt type. The state associated with utterance 14 remains **ReadCurrentVfolderSI-first**, the state associated with utterance 16 becomes **ReadCurrentVfolderSI-first-timeout1**, and the state associated with utterance 18 becomes **ReadCurrentVfolderSI-first-reject1**.

4.3.2 Strategy Branching

ELVIS' state space can be viewed as a graph, where nodes are states and arcs are state transitions. A graph that contains reentrant nodes, as depicted in Figure 4-1, is not ideal for reinforcement learning using the model-based algorithms presented in Section 4.1.2. Such representations lose information at the reentrant nodes (and their predecessor nodes). In Figure 4-1, information loss occurs at reentrant node d and predecessor nodes a , b , and c . Suppose node e has a utility of .9, node f has a utility of -.3, and d progresses to e $\frac{1}{2}$ of the time and to f the other $\frac{1}{2}$ of the time. Using Equation 4.2 for learning, d is assigned a utility of .3. Both b and c then inherit a utility of .3 from d . Any way in which b differs from c is lost.

Information loss occurs because the learning algorithms only use a state's imme-

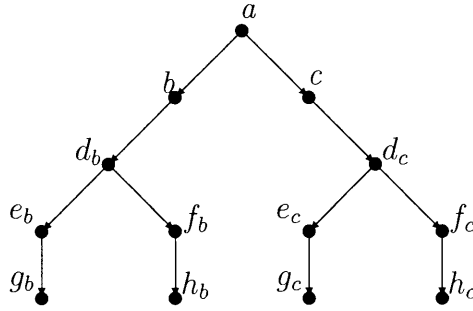


Figure 4-2: Figure 4-1 Transformed to Remove Reentrancy

mediate successor states in computing utilities. The presented learning algorithms can be adjusted to use multiple successor states in computing utilities, as described in [Jaakkola *et al.*, 1994, Singh *et al.*, 1994], but this does not solve the information loss for variable-length paths prior to reentrance.

The ELVIS state space as described in Section 2.4 and detailed in Appendix A has reentrant nodes where the different initiative, read, and summarization strategies eventually converge. Any information learned about differences between the MI and SI strategies, between the Read-Choice-Prompt, Read-First, and Read-Summarize-Only strategies, and between the Summarize-Both, Summarize-Choice, and Summarize-System strategies is lost, which runs counter to the goal of learning the comparative utilities of the implemented discourse strategies.

The state space can be transformed to remove the reentrant nodes where competing discourse strategy transition paths converge and recover the information lost by the presence of these reentrant nodes through *strategy branching*. The transformed state space branches for each strategy when the strategy is first used for the duration of an interaction with ELVIS. The state space depicted in Figure 4-1 becomes the state space in Figure 4-2; the choice of b or c is remembered as a state variable associated with all states that follow the choice (here, states d , e , f , g , and h). The strategy branching transformation associates strategy variables with each ELVIS state, effectively remembering which initiative, read, and summarization strategies

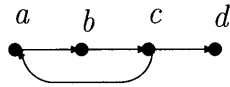


Figure 4-3: Simplified Representation of Successfully Completed ELVIS Interaction

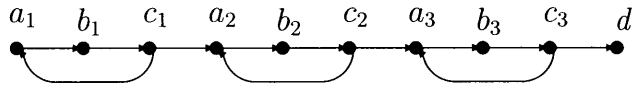


Figure 4-4: Figure 4-3 Transformed to Show Task Progress

are in use.

4.3.3 Representing Task Progress

ELVIS' state space, even with strategy branching in place, does not reflect task progress, as described in Section 2.4. In the experimental scenarios, there are three levels of task progress, from least to most: no subtasks have been completed (1), just one of the two subtasks has been completed (2), and both subtasks have been completed (3). Representing task progress is important because different discourse strategies may be most appropriate and effective at different stages of task completion. For example, summarizing messages before any subtasks have been completed is essentially a different state than summarizing messages after both subtasks have been completed, so discourse generation may have to be handled differently at each of these stages. The state space can be transformed to reflect this by adding a state variable that marks how much of the task has been completed.

Figure 4-3 is a greatly simplified representation of a successfully completed ELVIS interaction. *a* represents the top-level state, *b* represents email folder summarization,

c represents reading messages, and d represents hanging up. The arcs represent state transitions. The effects of the transformation that represents task progress is shown in Figure 4-4, where the subscripts show the highest level of task progress that has been attained.

4.4 The State Transition Model

The first thing that the agent learns from interactions with users is the state transition model M_{ij}^a . M_{ij}^a must be estimated from the conversations between ELVIS and actual users because what users choose to say to ELVIS and how they say it affects state transition probabilities and is nondeterministic. For example, an experienced user of the system may interrupt ELVIS' top-level utterance and say, "Summarize," causing ELVIS to progress to ELVIS' summarization state, while an inexperienced user might say, "Uh, I'd like, um, to get a summary of my messages," which is likely to lead to the top-level rejection state. ELVIS' utterances can guide users in utterance generation by telling them what they can say or can confuse them by offering inaccurate, vague, or no information about what can be said. At each iteration of learning, M_{ij}^a is recalculated from the cumulative conversational data. This section reports some qualitative results of the learned state transition model.

When users converse with ELVIS, they must adjust to more constrained conversations than most human-human dialogues because ELVIS only understands a small subset of the utterances that most humans can process. Users who know what they can say to ELVIS tend to avoid system timeouts, helps, and ASR rejections.

In the initiative strategies experiment, subjects accidentally requested to hang up 1.2 times more when using the SI version of ELVIS than when using the MI version of ELVIS. Subjects were often confused because they could say, "I'm done here" in one state to exit a virtual folder and return to their top-level inbox. In a different state, saying, "I'm done here" would cause ELVIS to ask the user, "Do you want to hang up now?"

In the presentation strategies experiment, when situated in a vfolder that con-

tained more than one message, subjects' utterances were rejected three times more often when using the Read-Choice-Prompt and Read-First strategies than when using the Read-Summarize-Only strategy. Additionally, timeouts occurred 2.25 times as often when subjects used the Read-First strategy than when they used the Read-Summarize-Only strategy and eight times more often when subjects used the Read-Summarize-Only strategy than when they used the Read-Choice-Prompt strategy. It is not evident why the rejections would occur disproportionately while using the read strategies but the distribution of timeouts among the read strategies is apparent. In the Read-Choice-Prompt strategy, ELVIS tells the user what to say to read particular messages, while the Read-First strategy, after reading the first message, and the Read-Summarize-Only strategy do not offer any information about how to select messages. When subjects did not know what to say, they would often timeout, waiting for ELVIS to provide more helpful information.

4.5 Learning Results

This section presents the results of running ADP and Q-learning on both sets of experimental data. This section describes results where local reward is the quantitatively-based function in Equation 4.8 and where local reward is set to zero. The iteration difference threshold for ADP and Q-learning is set to be 5% of the dialogue performance range, which is .1. The data tables that form the bases for the graphs in this section are provided in Appendix E. The absolute values of the presented utilities do not have any meaning; in choosing between possible states, only the relative utility values of those states matter.

4.5.1 Initiative Strategies

The utility of the SI initiative strategy ($U(SI)$) learned through ADP without local reward is .247, while the utility of the MI initiative strategy ($U(MI)$) learned through ADP without local reward is -.252. The learning trials for ADP without local reward are presented in Figure 4-5. For ADP with local reinforcement, $U(SI)=-1.291$ and

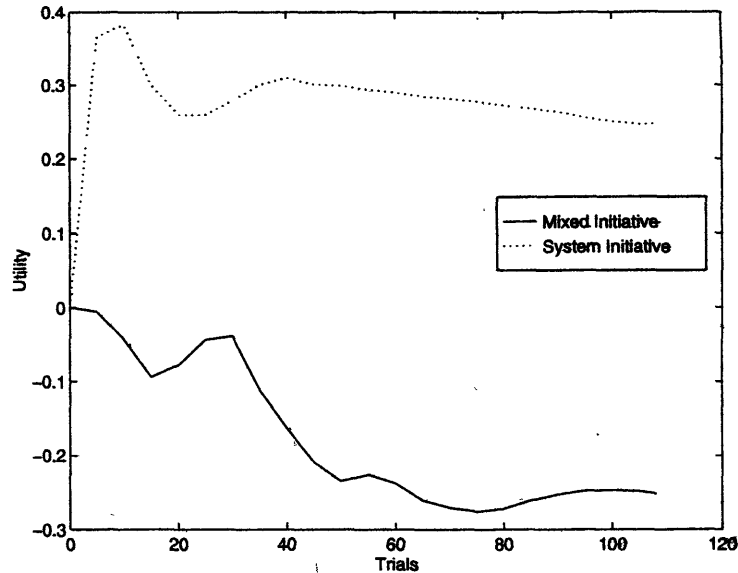


Figure 4-5: Results of ADP and Q-Learning without Local Reward for Initiative Strategies

$U(MI)=-2.338$. The learning trials for ADP with local reinforcement are presented in Figure 4-6.

Q-learning without the use of local rewards produces virtually identical results to ADP learning without the use of local rewards (Figure 4-5). Q-learning with the use of local rewards yields a final utility of -1.291 for SI and -2.325 for MI, as shown in Figure 4-7.

The utility values for the different algorithms can be normalized to compare the results of the different algorithms using a Z score normalization function [Cohen, 1995].

The results of utility normalization for SI are shown in Figure 4-8 and the results of utility normalization for MI are shown in Figure 4-9. In these figures, ADP and Q-learning are represented by an almost identical line because, when normalized, both algorithms achieve nearly identical results. The graphs support the convergence of the utility values for both SI and MI when learned with or without local reinforcement.

The convergence of the learning results for the initiative strategies is strengthened further by the ratio test for convergence [Edwards and Penney, 1990]. The ratio

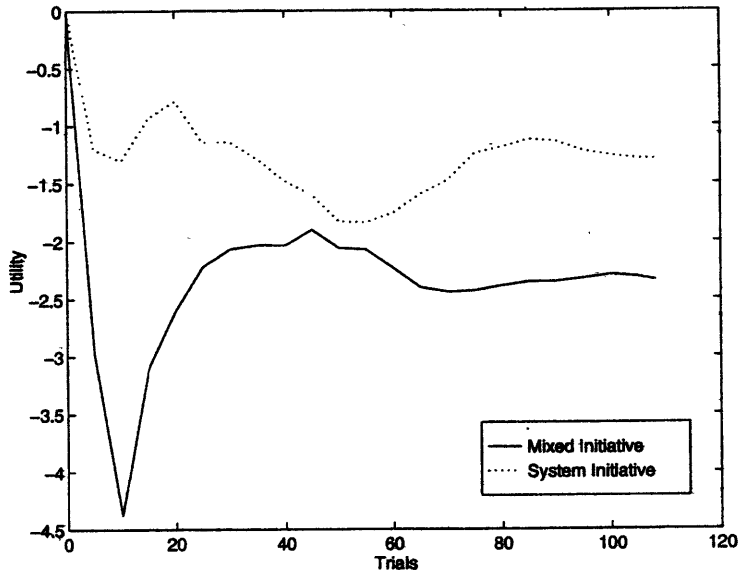


Figure 4-6: Results of ADP Learning with Local Reward for Initiative Strategies

test examines the ratio of difference between utility values in consecutive learning iterations. If the ratio converges to less than 1, learning can be deemed to have converged on the correct utility values. If the ratio converges to any other value, learning has not yet converged on accurate utility values. Since learning occurs over real-world data, it is unlikely that the series of the differences between utility values of consecutive iterations will increase or decrease purely monotonically; instead, convergence is granted when at least 50% of the last third of the learning iterations pass the ratio test. The utility values for SI and MI learned with and without local reinforcement have converged according to the ratio test.

The results of learning for the initiative strategies confirm the hypothesis that in ELVIS, SI is a better strategy than MI seemingly because of the better ASR performance rates and user inexperience. Given more technologically advanced spoken dialogue agents or more experienced users, users may be more satisfied with MI than SI because the users have more dialogue control with MI.

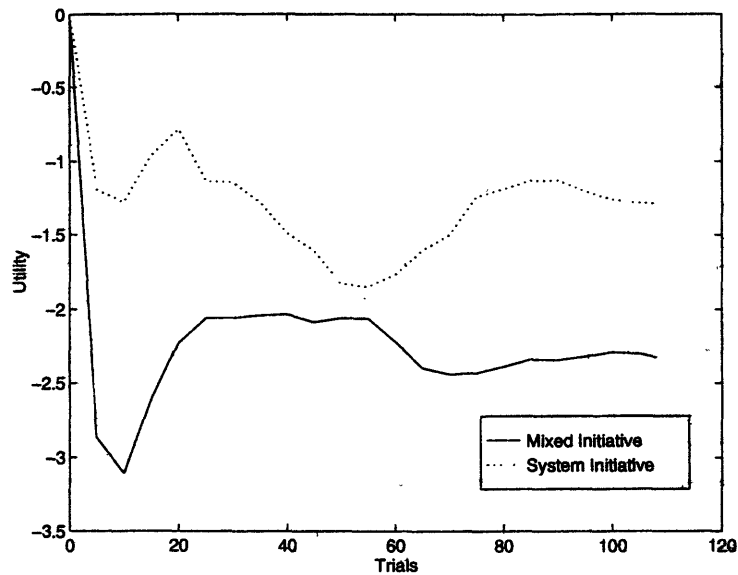


Figure 4-7: Results of Q-Learning with Local Reward for Initiative Strategies

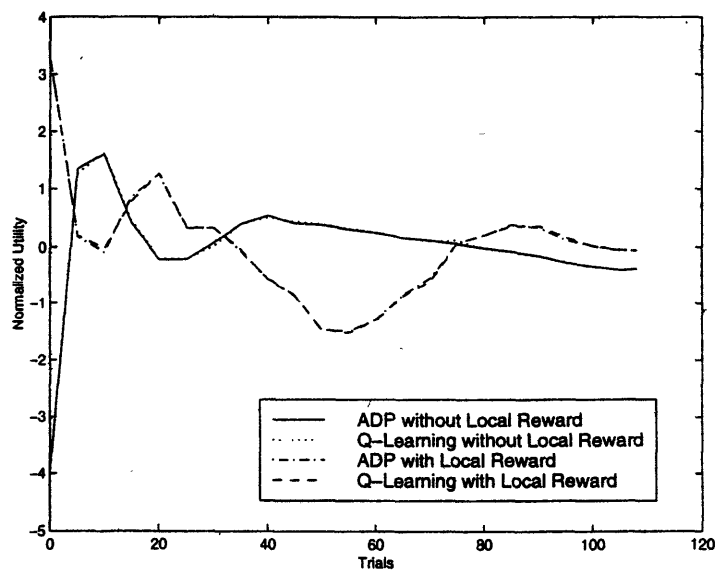


Figure 4-8: Comparison of Learning Algorithms and Local Reinforcement Use for System-Initiative

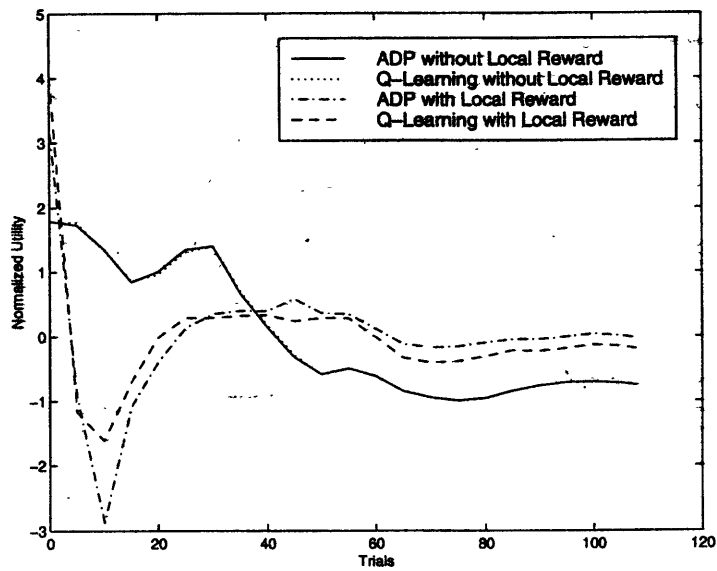


Figure 4-9: Comparison of Learning Algorithms and Local Reinforcement Use for Mixed-Initiative

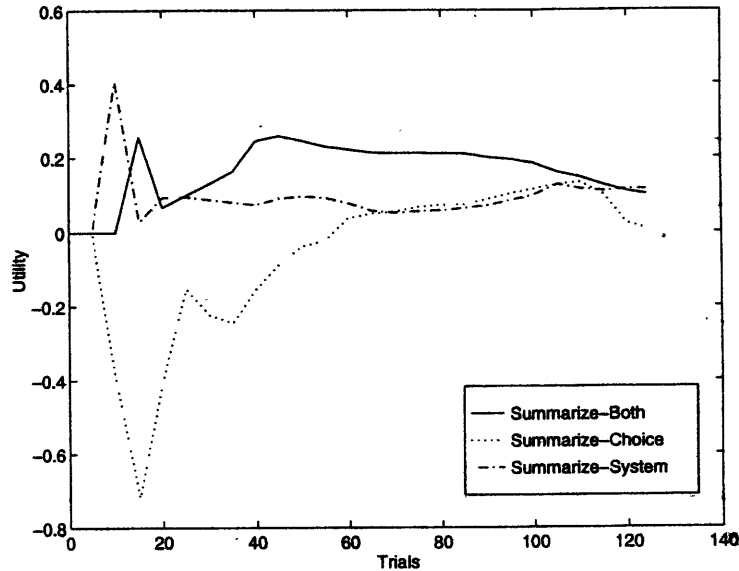


Figure 4-10: Results of ADP and Q-Learning without Local Reward for Summarization Strategies at Task Progress Level 1

4.5.2 Summarization Strategies

Unlike the initiative strategies, which are invoked once toward the start of a conversation with ELVIS (even if they are used throughout an interaction with ELVIS), the summarization strategies can be invoked multiple times in a conversation with ELVIS. The summarization strategy learning results are presented at the three different task progress levels that are described in Section 4.3.3.

The results of ADP are virtually identical to the results of Q-learning and they are presented in Tables 4.1 and 4.2. The learning process is shown in Figures 4-10 - 4-15.

The utility values for the different algorithms and use of local reinforcement can be normalized to compare the results of the different algorithms using a Z score normalization function. These comparisons are shown in Figures 4-16 - 4-24. ADP and Q-learning with the same type of local reinforcement are represented almost identically because both algorithms achieve nearly identical results. The graphs support the convergence of all the summarization strategy utility values learned with local re-

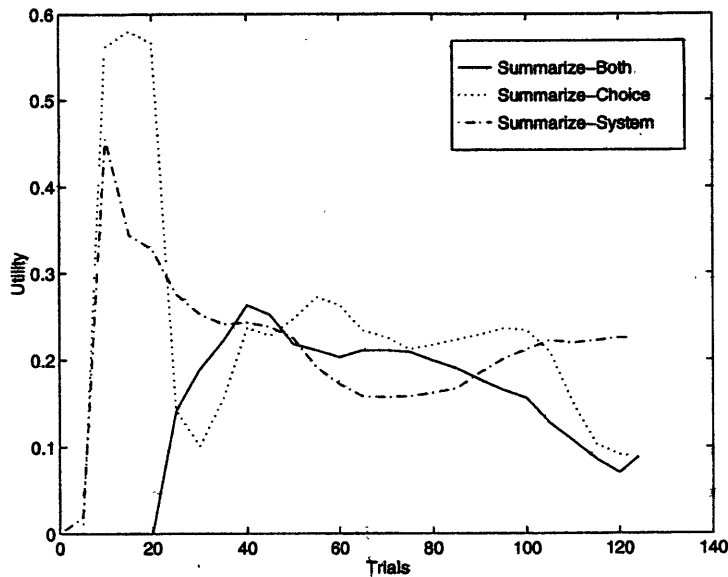


Figure 4-11: Results of ADP and Q-Learning without Local Reward for Summarization Strategies at Task Progress Level 2

inforcement, while only the utility values for Summarize-System at all task progress levels appear to have converged when learned without local reinforcement.

The ratio test shows that all of the summarization strategy learning results that use local reinforcement excluding the utility value for Summarize-Choice at task progress level 2 have converged. Without local reinforcement, the ratio test demonstrates that only the utility values for Summarize-System at all task progress levels and Summarize-Choice at task progress levels 1 and 3 have converged. More human data would be necessary for the convergence of the summarization strategies' utility values that have not yet converged. Though some conflicting results for the comparative utilities of the three implemented summarization strategies have been obtained, the ratio test judges the results learned through local reinforcement to be more accurate than those learned without reinforcement. Local reinforcement is necessary for incorporating knowledge of local performance minima and maxima into the reinforcement learning process and the results of ADP and Q-learning with local reward are deemed the most reliable.

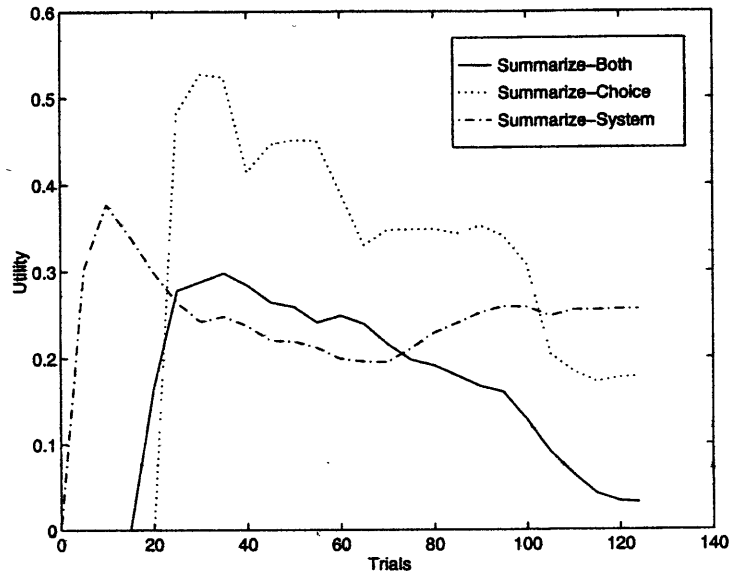


Figure 4-12: Results of ADP and Q-Learning without Local Reward for Summarization Strategies at Task Progress Level 3

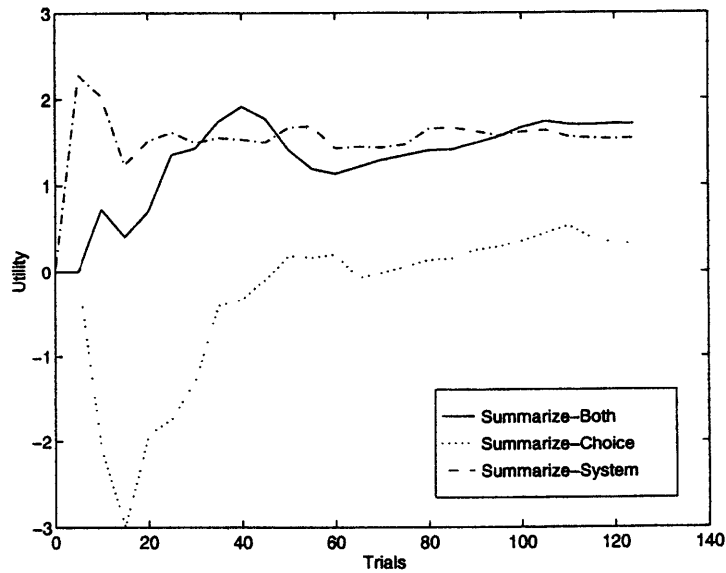


Figure 4-13: Results of ADP and Q-Learning with Local Reward for Summarization Strategies at Task Progress Level 1

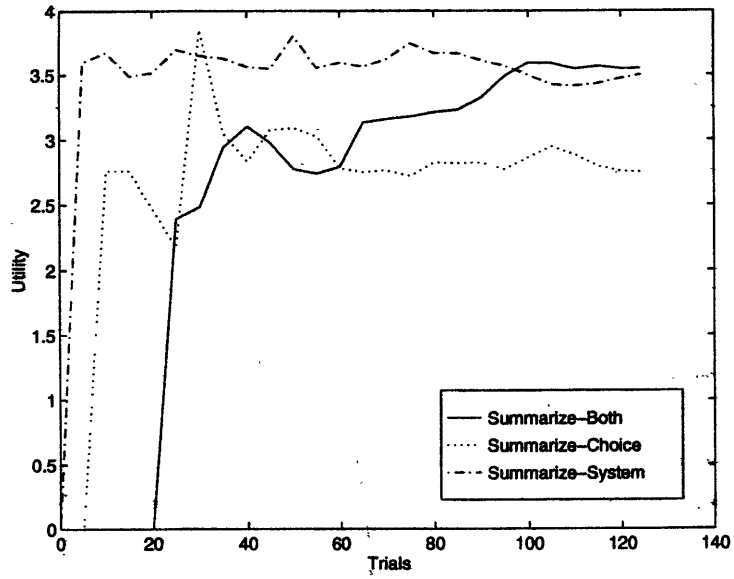


Figure 4-14: Results of ADP and Q-Learning with Local Reward for Summarization Strategies at Task Progress Level 2

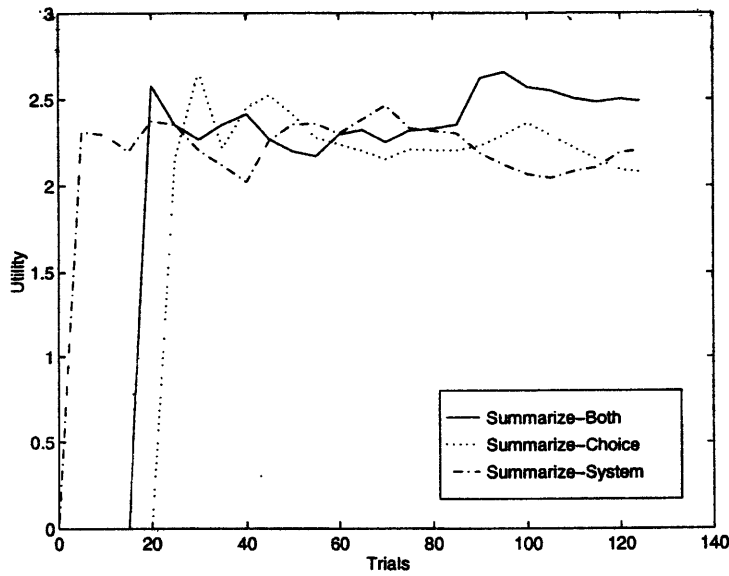


Figure 4-15: Results of ADP and Q-Learning with Local Reward for Summarization Strategies at Task Progress Level 3

| Local Reward? | Task Progress Level | Summarize-System | Summarize-Choice | Summarize-Both |
|---------------|---------------------|------------------|------------------|----------------|
| No | 1 | .115 | .01 | .101 |
| No | 2 | .225 | .088 | .088 |
| No | 3 | .256 | .178 | .032 |
| Yes | 1 | 1.542 | .321 | 1.709 |
| Yes | 2 | 3.497 | 2.751 | 3.547 |
| Yes | 3 | 2.205 | 2.08 | 2.491 |

Table 4.1: Final Utilities from ADP and Q-Learning for Summarization Strategies

| Local Reward? | Task Progress Level | Summarization Strategies |
|---------------|---------------------|--|
| No | 1 | Summarize-System > Summarize-Both > Summarize-Choice |
| No | 2 | Summarize-System > Summarize-Both = Summarize-Choice |
| No | 3 | Summarize-System > Summarize-Choice > Summarize-Both |
| Yes | 1 | Summarize-Both > Summarize-System > Summarize-Choice |
| Yes | 2 | Summarize-Both > Summarize-System > Summarize-Choice |
| Yes | 3 | Summarize-Both > Summarize-System > Summarize-Choice |

Table 4.2: Summarization Strategy Preferences Learned through ADP and Q-Learning

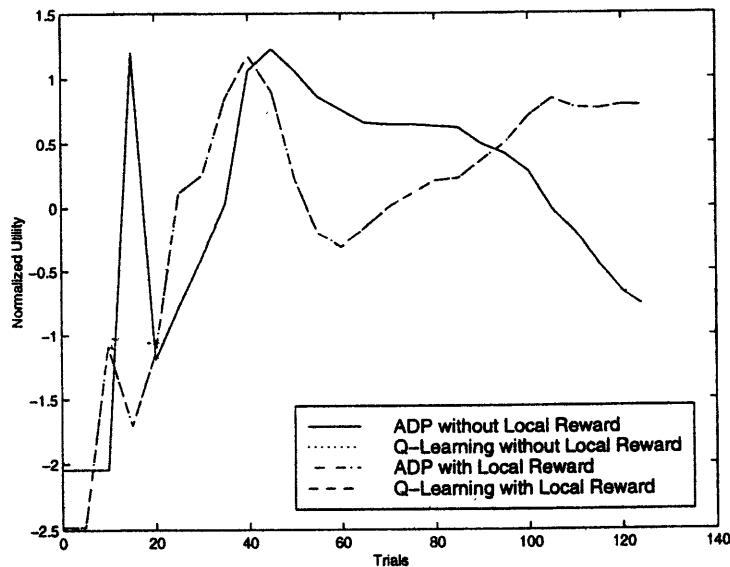


Figure 4-16: Comparison of Learning Algorithms and Local Reinforcement Use for Summarize-Both at Task Progress Level 1

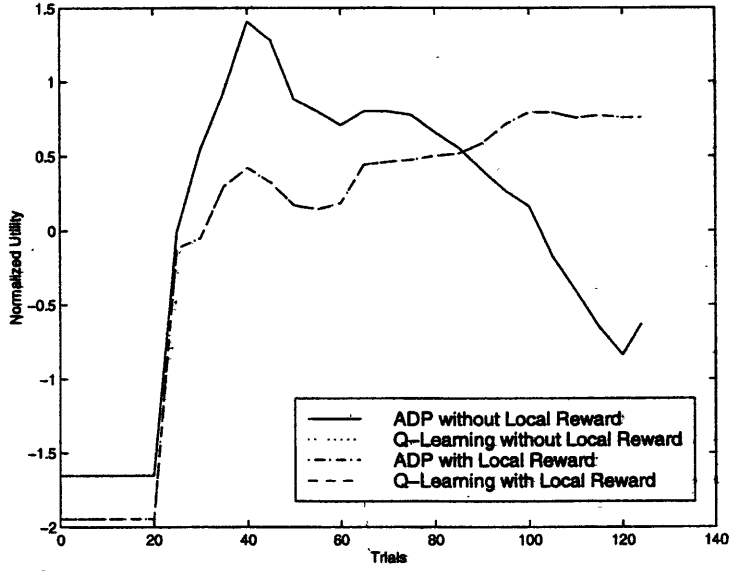


Figure 4-17: Comparison of Learning Algorithms and Local Reinforcement Use for Summarize-Both at Task Progress Level 2

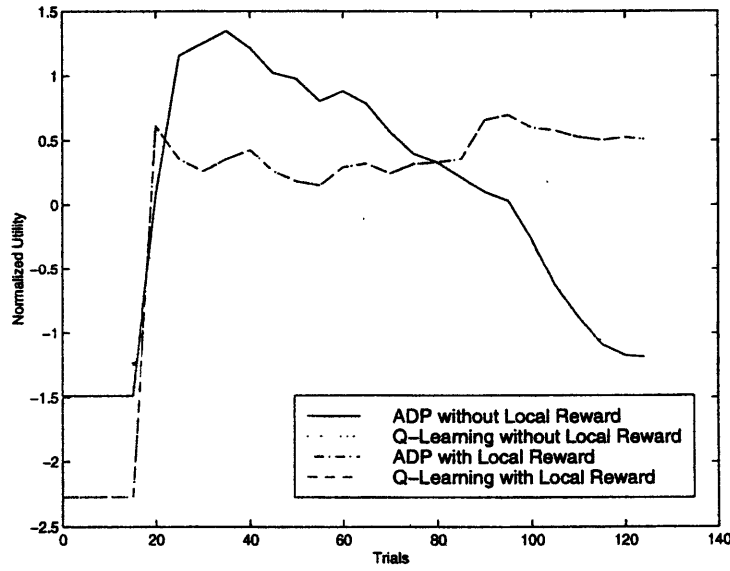


Figure 4-18: Comparison of Learning Algorithms and Local Reinforcement Use for Summarize-Both at Task Progress Level 3

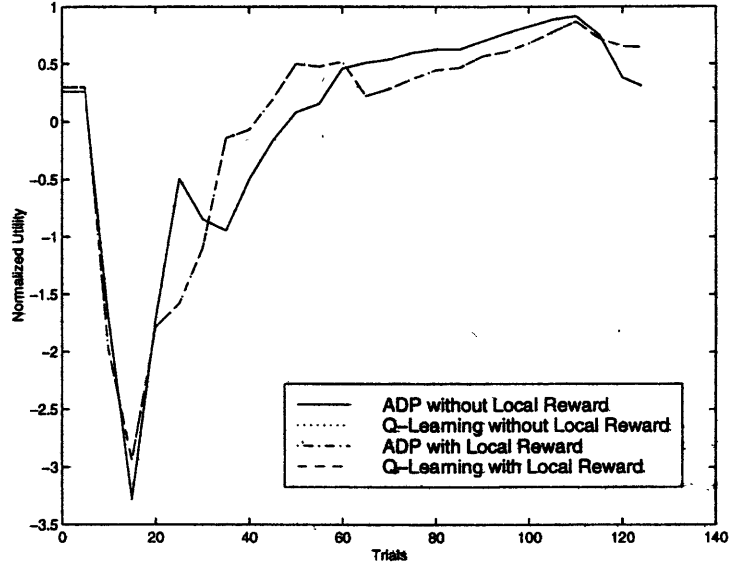


Figure 4-19: Comparison of Learning Algorithms and Local Reinforcement Use for Summarize-Choice at Task Progress Level 1

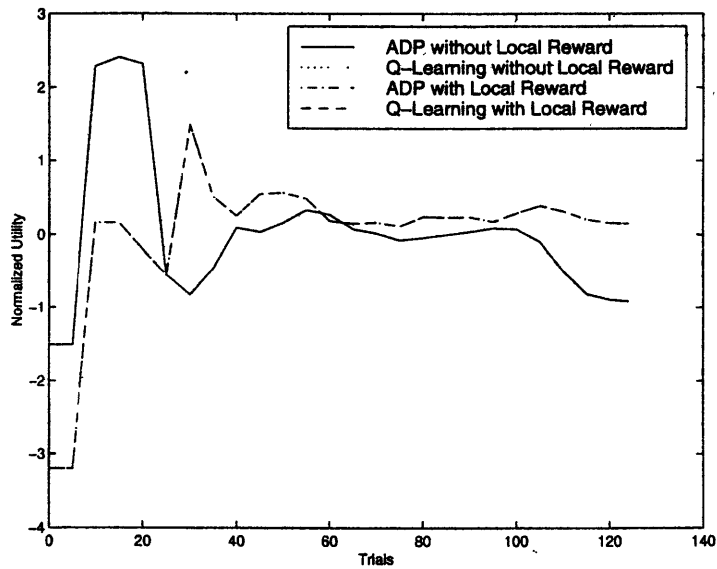


Figure 4-20: Comparison of Learning Algorithms and Local Reinforcement Use for Summarize-Choice at Task Progress Level 2

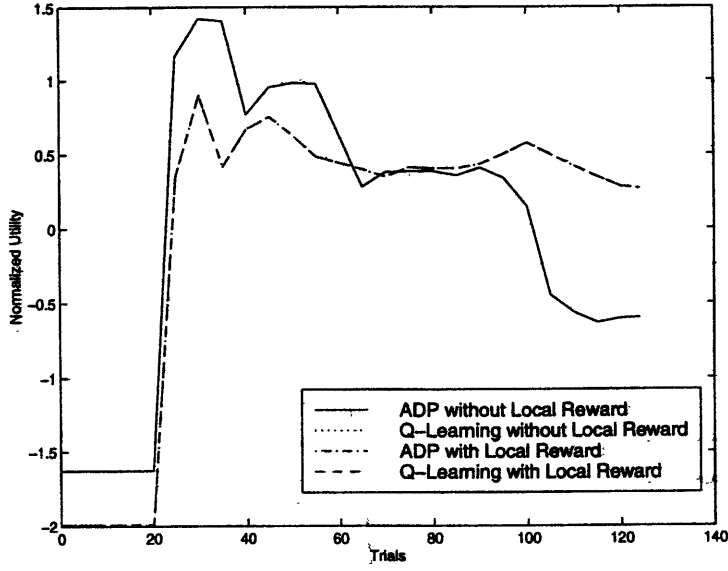


Figure 4-21: Comparison of Learning Algorithms and Local Reinforcement Use for Summarize-Choice at Task Progress Level 3

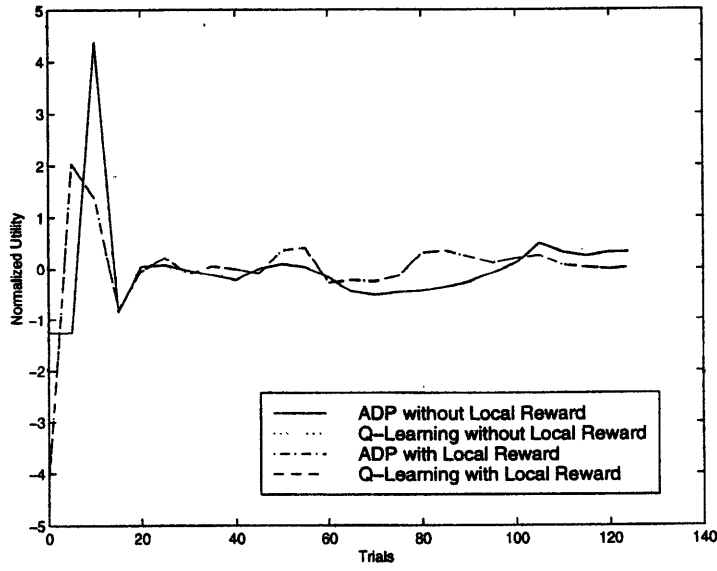


Figure 4-22: Comparison of Learning Algorithms and Local Reinforcement Use for Summarize-System at Task Progress Level 1

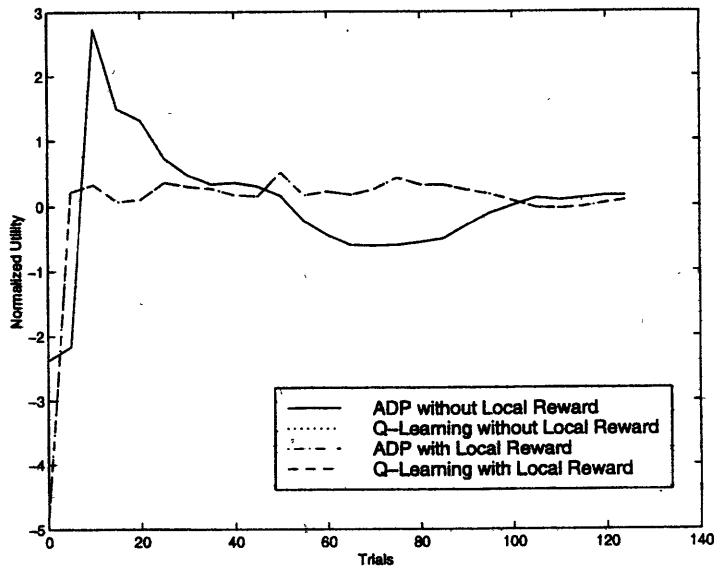


Figure 4-23: Comparison of Learning Algorithms and Local Reinforcement Use for Summarize-System at Task Progress Level 2

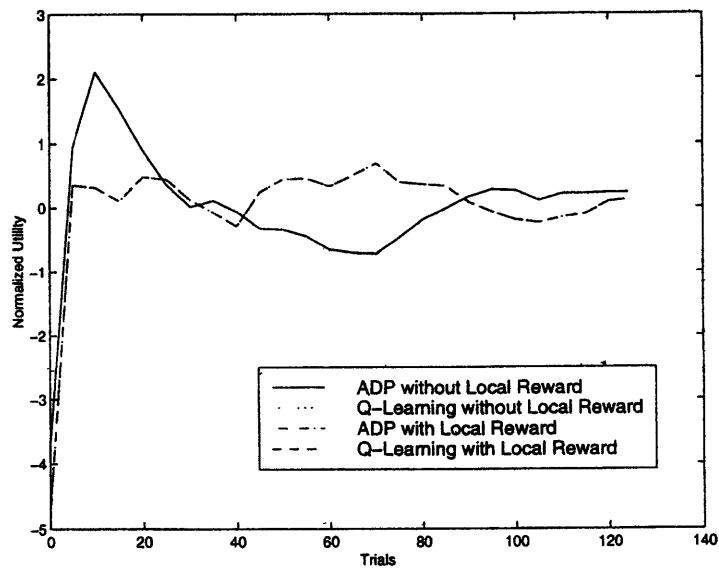


Figure 4-24: Comparison of Learning Algorithms and Local Reinforcement Use for Summarize-System at Task Progress Level 3

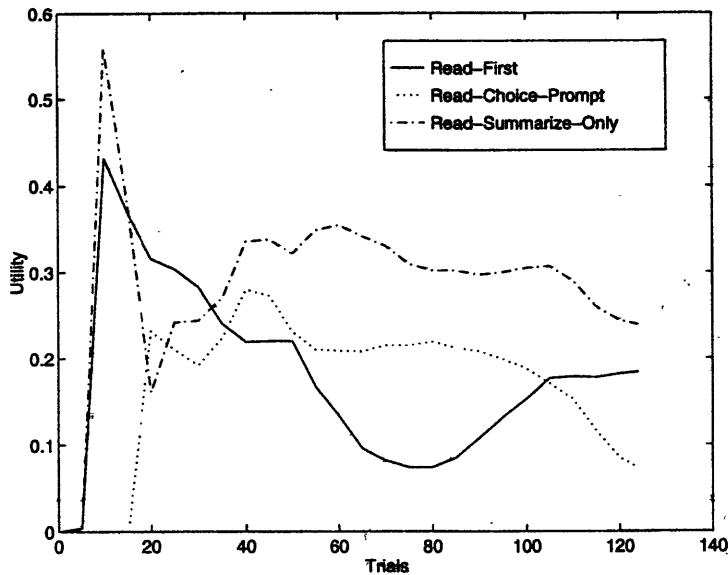


Figure 4-25: Results of ADP and Q-Learning without Local Reward for Read Strategies at Task Progress Level 1

4.5.3 Read Strategies

Like the summarization strategies implemented in ELVIS, the read strategies can be invoked multiple times in a conversation with ELVIS. The read strategy learning results are presented at the three different task progress levels as described in Section 4.3.3.

The results of ADP are virtually identical to the results of Q-learning and they are presented in Tables 4.3 and 4.4. The learning process is shown in Figures 4-25 - 4-30.

The utility values for the different algorithms and use of local reinforcement can be normalized to compare the results of the different algorithms using a Z score normalization function. These comparisons are shown in Figures 4-31 - 4-39. ADP and Q-learning with the same type of local reinforcement are represented almost identically because both algorithms achieve nearly identical results. The graphs indicate that all of the read strategy utility values learned with local reinforcement have converged, while only the utility values for Read-First at task progress levels 1 and

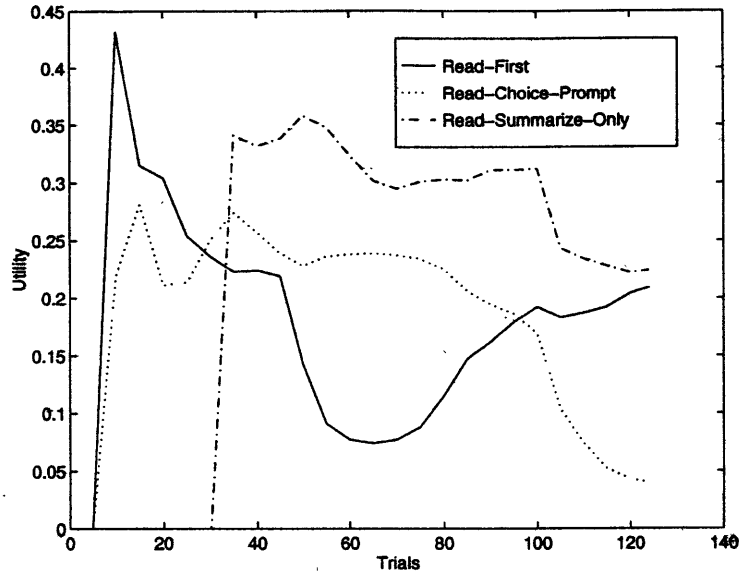


Figure 4-26: Results of ADP and Q-Learning without Local Reward for Read Strategies at Task Progress Level 2

| Local Reward? | Task Progress Level | Read-First | Read-Summarize-Only | Read-Choice-Prompt |
|---------------|---------------------|------------|---------------------|--------------------|
| No | 1 | .184 | .239 | .073 |
| No | 2 | .209 | .224 | .04 |
| No | 3 | .217 | .22 | .037 |
| Yes | 1 | 2.767 | 2.478 | 1.919 |
| Yes | 2 | 2.2 | 1.851 | 2.394 |
| Yes | 3 | 2.759 | 3.383 | 4.285 |

Table 4.3: Final Utilities from ADP and Q-Learning for Read Strategies

Read-Summarize-Only at task progress levels 2 and 3 appear to have converged when learned without local reinforcement.

The ratio test shows that all of the read strategy learning results that use local reinforcement have converged. Without local reinforcement, the ratio test demonstrates that only the utility values for Read-Summarize-Only at task progress levels 1 and 2 have converged. More human data would be necessary for the convergence of the read strategies' utility values that have not yet converged. Though some conflicting results for the comparative utilities of the three implemented read strategies have

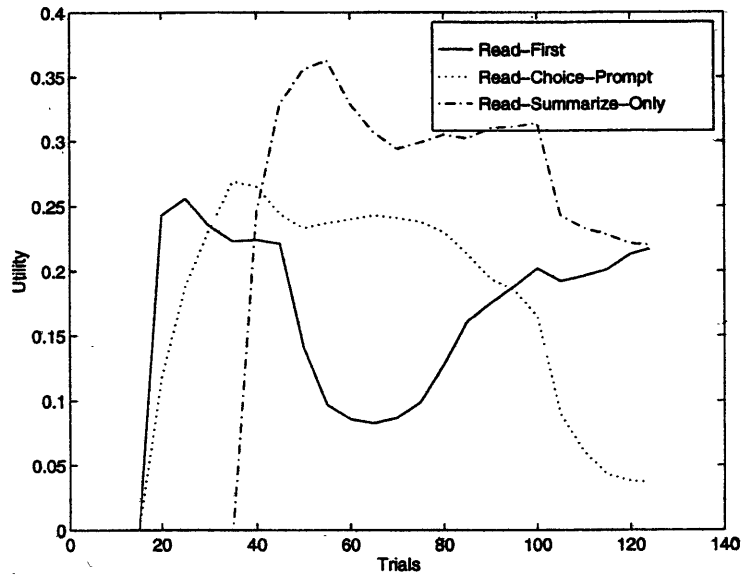


Figure 4-27: Results of ADP and Q-Learning without Local Reward for Read Strategies at Task Progress Level 3

been obtained, the ratio test judges the results learned through local reinforcement to be more accurate than those learned without reinforcement. Local reinforcement is necessary for incorporating knowledge of local performance minima and maxima into the reinforcement learning process and the results of ADP and Q-learning with local reward are deemed the most reliable.

The next chapter describes work related to ELVIS, the spoken dialogue agent and the learner of discourse strategies.

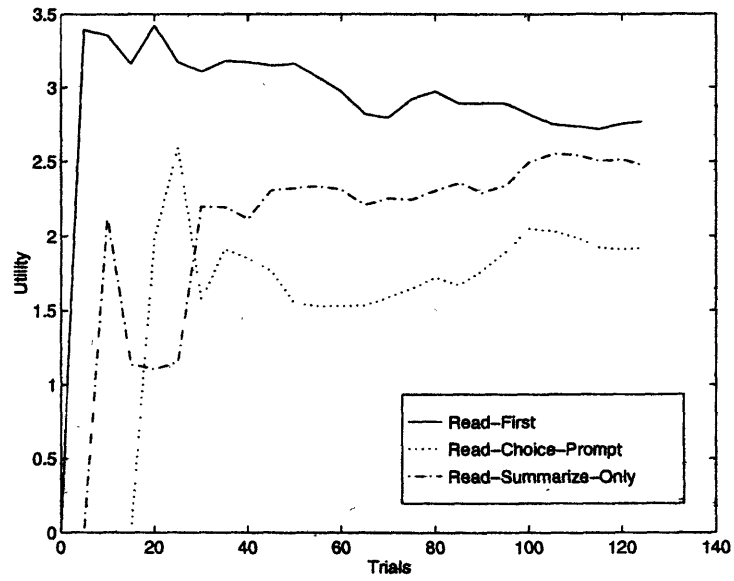


Figure 4-28: Results of ADP and Q-Learning with Local Reward for Read Strategies at Task Progress Level 1

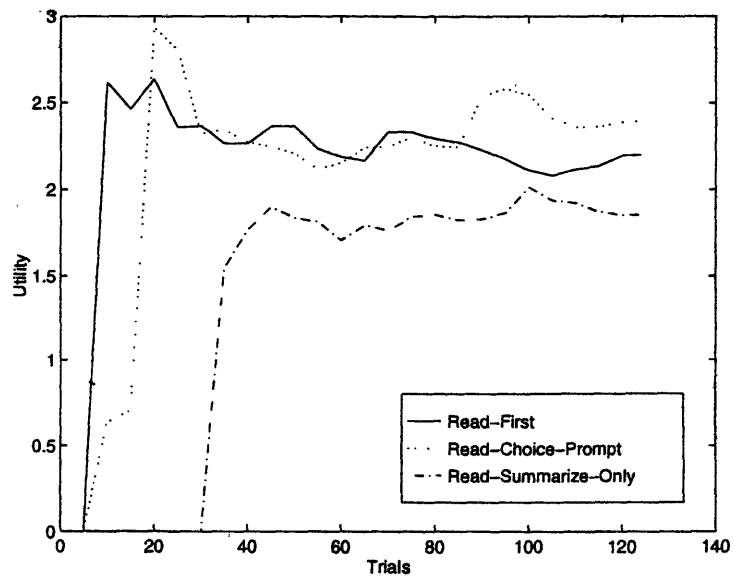


Figure 4-29: Results of ADP and Q-Learning with Local Reward for Read Strategies at Task Progress Level 2

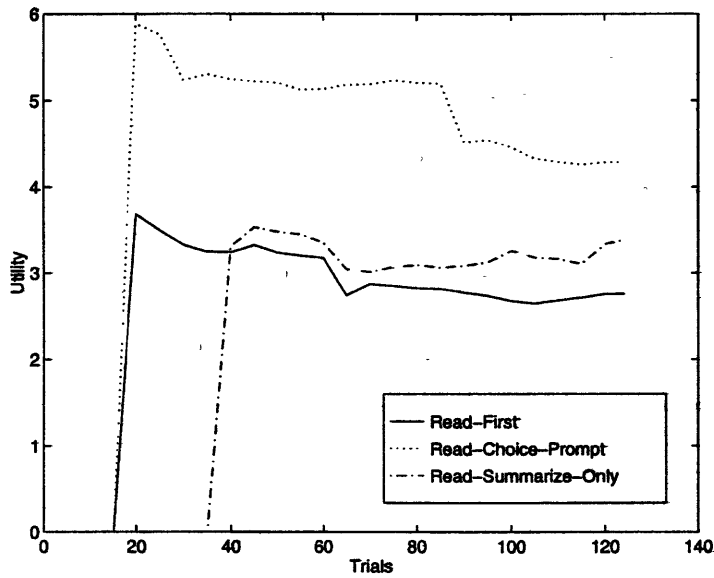


Figure 4-30: Results of ADP and Q-Learning with Local Reward for Read Strategies at Task Progress Level 3

| Local Reward? | Task Progress Level | Read Strategies |
|---------------|---------------------|---|
| No | 1 | Read-Summarize-Only > Read-First > Read-Choice-Prompt |
| No | 2 | Read-Summarize-Only > Read-First > Read-Choice-Prompt |
| No | 3 | Read-Summarize-Only > Read-First > Read-Choice-Prompt |
| Yes | 1 | Read-First > Read-Choice-Prompt > Read-Summarize-Only |
| Yes | 2 | Read-Choice-Prompt > Read-First > Read-Summarize-Only |
| Yes | 3 | Read-Choice-Prompt > Read-Summarize-Only > Read-First |

Table 4.4: Read Strategy Preferences Learned through ADP and Q-Learning

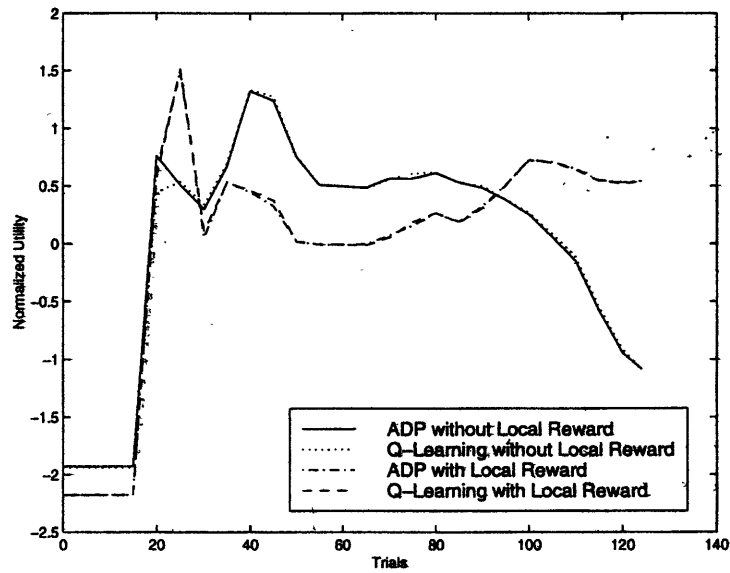


Figure 4-31: Comparison of Learning Algorithms and Local Reinforcement Use for Read-Choice-Prompt at Task Progress Level 1

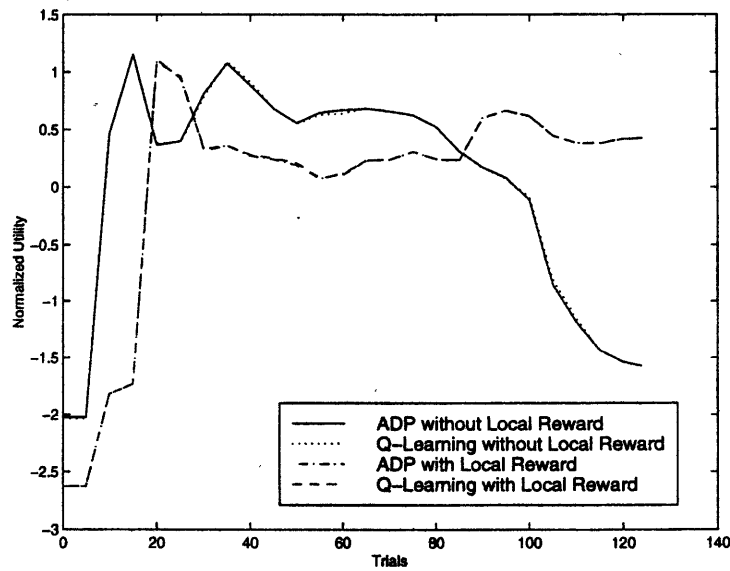


Figure 4-32: Comparison of Learning Algorithms and Local Reinforcement Use for Read-Choice-Prompt at Task Progress Level 2

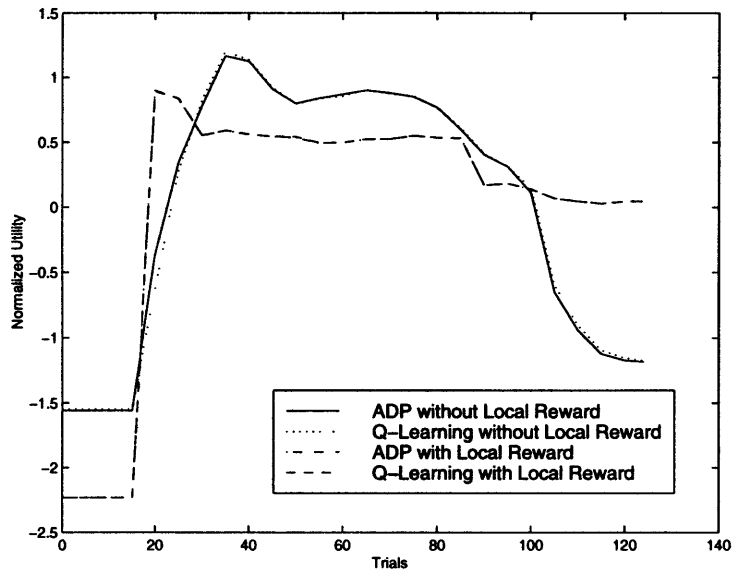


Figure 4-33: Comparison of Learning Algorithms and Local Reinforcement Use for Read-Choice-Prompt at Task Progress Level 3

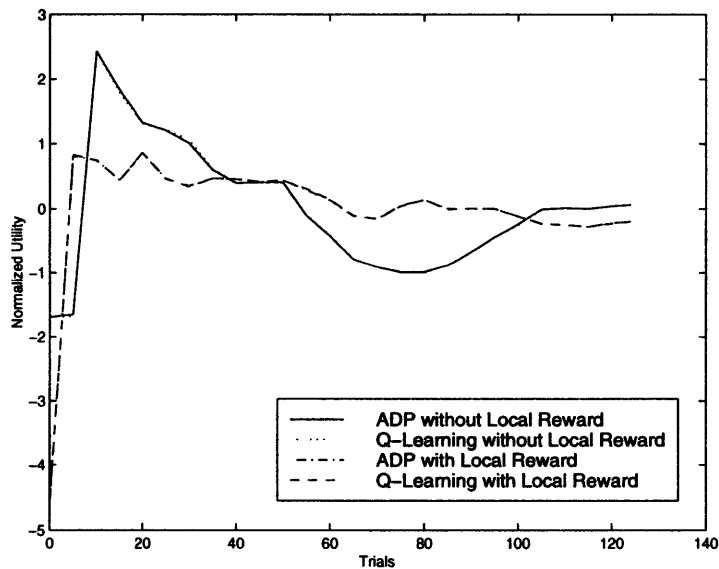


Figure 4-34: Comparison of Learning Algorithms and Local Reinforcement Use for Read-First at Task Progress Level 1

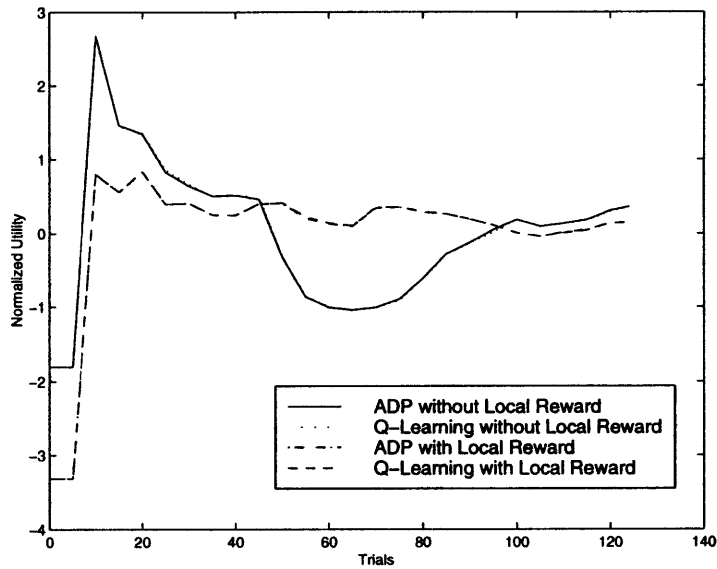


Figure 4-35: Comparison of Learning Algorithms and Local Reinforcement Use for Read-First at Task Progress Level 2

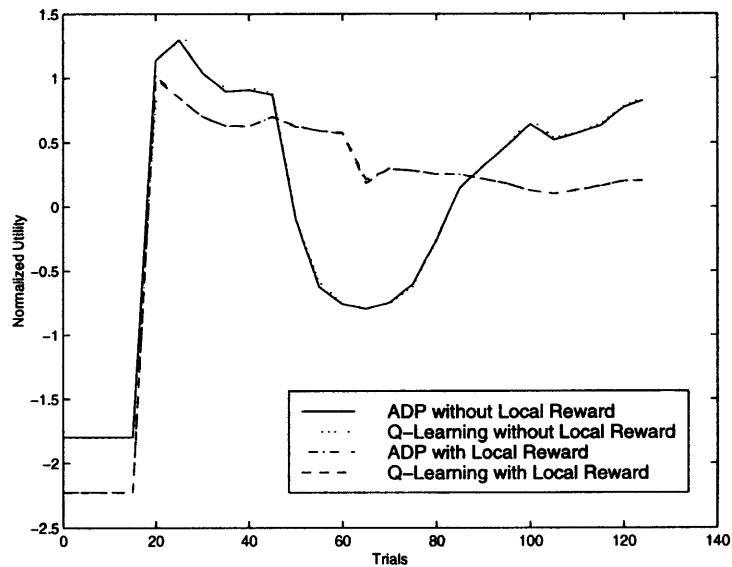


Figure 4-36: Comparison of Learning Algorithms and Local Reinforcement Use for Read-First at Task Progress Level 3

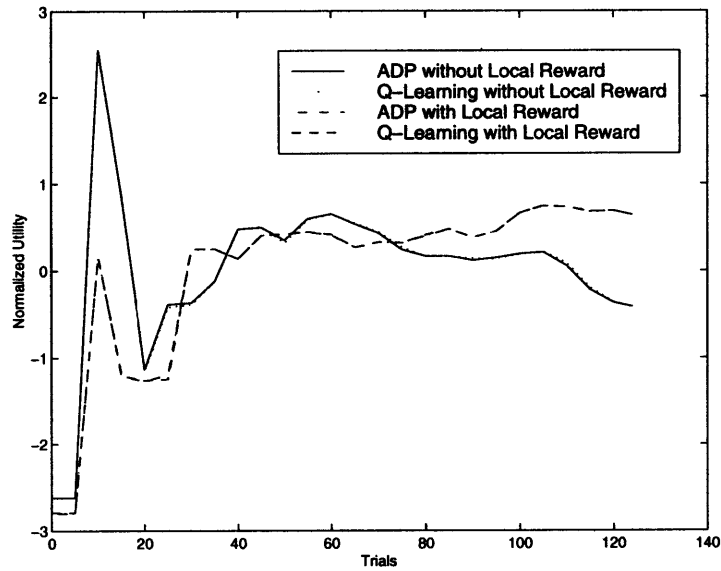


Figure 4-37: Comparison of Learning Algorithms and Local Reinforcement Use for Read-Summarize-Only at Task Progress Level 1

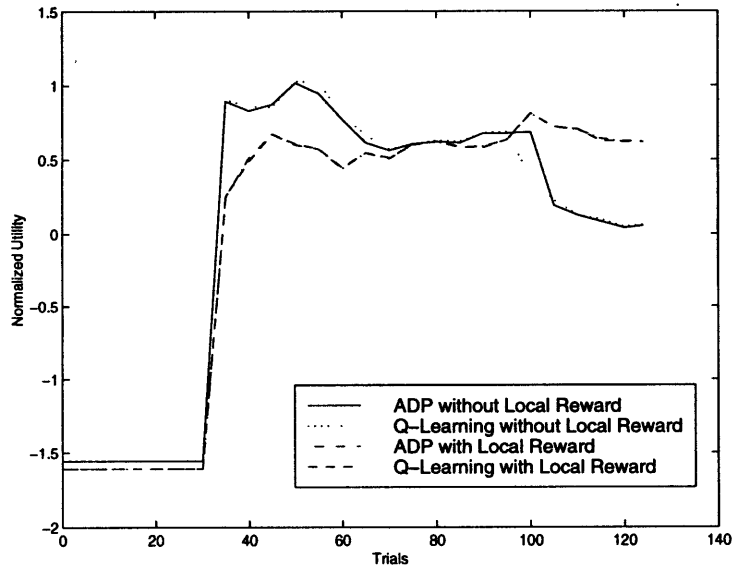


Figure 4-38: Comparison of Learning Algorithms and Local Reinforcement Use for Read-Summarize-Only at Task Progress Level 2

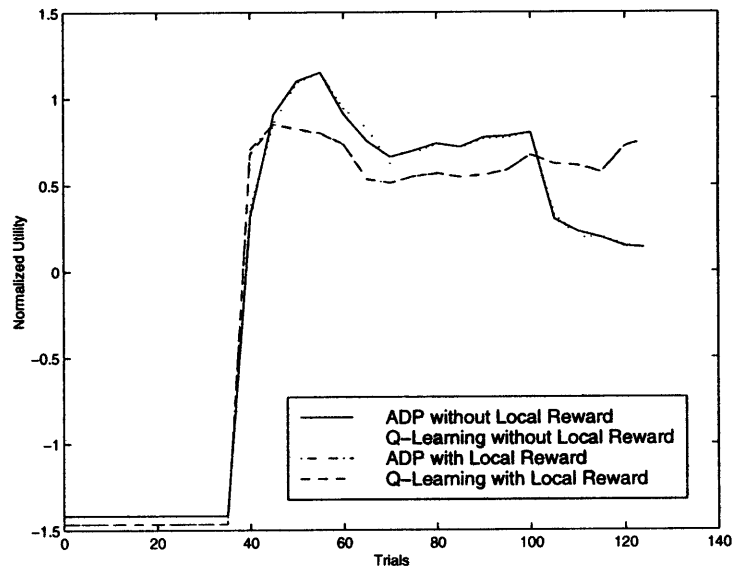


Figure 4-39: Comparison of Learning Algorithms and Local Reinforcement Use for Read-Summarize-Only at Task Progress Level 3

Chapter 5

Related Work

This chapter describes work related to ELVIS and learning optimal discourse strategies in a spoken dialogue system. This chapter reports on spoken dialogue systems, natural language content planning, adaptive agents, the effect of representation on learning, and some related learning applications and discusses how the work reported in this thesis relates to previous research.

Natural language systems are computational systems that provide components for the comprehension and generation of natural language. Usually, such components include syntactic analyzers, semantic interpreters, discourse understanders, and pragmatic handlers. (For an overview of natural language systems, see [Allen, 1995, Grosz *et al.*, 1986].) The focus of this thesis is on the discourse-related issues in dialogue systems. (More information on computational discourse issues can be found in [Cohen *et al.*, 1990, Grosz and Sidner, 1986, Brady and Berwick, 1982].)

5.1 Spoken Dialogue Systems

This section describes other spoken dialogue systems related to email access and claims about initiative distribution within spoken dialogue systems.

5.1.1 Previous Systems for Accessing Email

[Crabbe *et al.*, 1995, Martin and Kehler, 1994, Yankelovich, 1994] describe SpeechActs, a telephone-based spoken dialogue system framework. Spoken dialogue systems have been created using SpeechActs to support access to voice mail and email. MailCall is another telephone-based messaging system, where incoming voice mail and email messages are categorized and prioritized based on the user's current interests as inferred from the user's calendar [Marx, 1995]. MailCall supports inferential learning rather than the reinforcement learning that ELVIS handles.

5.1.2 Initiative

In a spoken dialogue system, users generally interact with the system through speech input (ASR systems) and output (TTS systems). High recognition accuracy is not easily achievable for large, low-context directories given the current state of ASR technology [Kamm *et al.*, 1995]. For example, [Kamm *et al.*, 1995] report that the recognition accuracy rate for automating telephone directory assistance calls decreases from 82.5% for a 200-name directory to 16.5% for a 1.5 million name directory. [Kamm, 1994] recommends that designers of spoken dialogue systems consider the system's task requirements, the capabilities and limitations of the system's technology, and the characteristics of the user population. Given the constraints of current ASR technology, [Kamm, 1994] advises that a system's grammars be kept small.

[Kamm, 1994] additionally advises that a spoken dialogue system's prompts be directive, allowing for as little vocabulary variability and user initiative as possible. On the other end of the initiative spectrum, [Walker and Whittaker, 1990] explore mixed initiative dialogue and hypothesize that users feel in control in a mixed-initiative dialogue system but need to accrue experience with the system to learn what to say. This thesis tested the claims of [Kamm *et al.*, 1995, Kamm, 1994, Walker and Whittaker, 1990] in the initiative strategies experiment and the results are reported in Chapters 3 and 4.

5.2 Natural Language Content Planning

In [McKeown, 1985], a natural language generation system accomplishes communicative goals by producing natural language. The system decides what content to generate and organizes this content so a user can easily understand the generated natural language. Traditional natural language generation systems contain a strategic component to determine what to say and the ordering of what will be said and a tactical component which determines how to say the content that has been selected [Robin, 1995, Zukerman and McConachy, 1995, Moore and Paris, 1993, Hovy, 1993].

This thesis is concerned with the strategic component of generation. Domain-independent language planning rules that rely on rhetorical strategies [Mann and Thompson, 1987] and heuristics are frequently provided for use within the strategic component [Harvey and Carberry, 1998, Callaway and Lester, 1997, Appelt, 1985]. In this thesis, strategic templates are provided to ELVIS instead of planning rules; these templates are instantiated whenever an implemented discourse strategy is invoked. These templates are for distribution of initiative, summarizing email folders, and reading messages. Instead of using planning rules, ELVIS learns planning rules (discourse strategy policies) through interactions with actual users followed by evaluation using the PARADISE framework and reinforcement learning.

5.3 Adaptive Agents

An *agent* is a system that perceives and acts [Russell and Norvig, 1995]; artificial intelligence can be viewed as the enterprise of understanding and building rational agents. ELVIS is an agent that interacts with users using natural language, providing them with remote access to their email. ELVIS learns optimal discourse strategies through interactions with actual users and can be seen as an adaptive interface agent.

In prior work on architectures for adaptable agents, [Morignot and Hayes-Roth, 1995, Hayes-Roth *et al.*, 1993] present an architecture for adaptable motivational profiles for autonomous agents. This architecture is psychologically inspired and allows

motivations to be configured so an agent can adapt in different environments. This agent architecture type is a generalized form of a natural language agent architecture that provides language planning rules, as described in Section 5.2. ELVIS differs from the motivated agent because different behaviors (strategies) are specified without any explicit motivation configuration and motivation can effectively be learned through ELVIS' real-world experiences.

5.4 Reinforcement Learning

Reinforcement learning, as detailed in Chapter 4, is the learning of utilities for different percepts and actions in the environment. (For an overview of reinforcement learning, see [Kaelbling, 1996, Kaelbling *et al.*, 1996, Sutton, 1992, Watkins, 1989, Keeney and Raiffa, 1976, Bellman, 1957].)

[Sutton, 1991, Sutton, 1990] describe integrated architectures for learning, planning, and reacting using dynamic programming algorithms for reinforcement learning. They describe the Dyna architecture, which integrates reinforcement learning with execution-time planning in a single process. Currently, ELVIS acts and later learns off-line; Section 6.2.1 describes how to integrate acting and learning.

The following sections describe work related to the effects of state space representation on reinforcement learning results and some applications of reinforcement learning.

5.4.1 State Space Representation

Reinforcement learning has some limitations related to the state space representation of a system: learning is representation-dependent and is history-insensitive [Lin, 1992]. In ELVIS, these problems are avoided by tailoring ELVIS' representation to the tasks performed in ELVIS and by remembering relevant history information as state variables, as described in Section 4.3.

[Whitehead and Ballard, 1991] discuss the differences between the internal state representation that an agent has and the world's representation. A problem with

active perception is that the agent's internal representation confounds external world states. *Perceptual aliasing* is the overlap between the world and the agent's internal representation. Most of the information that distinguishes states internally is irrelevant to the immediate task faced by the agent and interferes with decision making due to irrelevant details. [Agre and Chapman, 1987] have recognized this problem and suggest using *deictic representations*, which register information only about objects that are relevant to the task at hand. It is not straightforward to integrate deictic representations and reinforcement learning because in order for an agent to learn to solve a task, it must accurately represent the world with respect to that task, but in order to learn an accurate representation, the agent must know how to solve the task.

When perceptual aliasing occurs, according to [Whitehead and Ballard, 1991], learning cannot work properly because the system confounds world states that it must be able to distinguish in order to solve its task. This makes the state space non-Markovian because an action's effects are not independent of the past but depend on hidden, unperceived states of the actual, underlying state space. When one internal state refers to many world states, that state takes on a utility somewhere between the values for the world states it represents and aberrational maxima occur. The optimal policy for this system is unstable and the system will oscillate among policies. An aberrational maximum attracts nearby states and causes them to change their policies from optimal. These maxima prevent the learning of optimal policies by causing the system to average different values from different world states. An internal state should have the same utility as its corresponding world state and should have one internal action when executed that maps to the optimal action in the world. This thesis partially addresses the differences between the real world and a system's internal model of the world because ELVIS learns from interactions in the real world and any inaccuracies in its sensing capabilities are reflected in the final reward and local reinforcements that ELVIS receives in learning.

[Koenig and Simmons, 1996] discuss how the complexity of running learning algorithms depends on the representation chosen. By representation, they refer to the initial utility values and the local reward function. There is a great deal more to

state space representation than the initial utility values and local reward function and Section 4.3 describes the representation issues that are relevant to reinforcement learning in ELVIS.

5.4.2 Different Applications

This section discusses some of the applications that use reinforcement learning.

[Tesauro, 1992] describes the training of connectionist networks to play backgammon using a TD reinforcement learning algorithm. Other reinforcement learning applications include the optimal allocation of market assets [Neuneier, 1995], graphical user interface adaptation [Lenzmann, 1996], and network traffic control adaptation [Choi and Yeung, 1995].

ELVIS is unlike a game since optimally, it has no losers: ELVIS and the ELVIS user cooperate in information exchange, where the goal is to maximize user satisfaction. More similar to the cooperative nature of ELVIS is work where mobile robots engage in a cooperative task and must learn to communicate [Yanco and Stein, 1993]. In this work, vocabulary elements are learned with different meanings in different experimental simulations. In contrast, this thesis uses real interactions with real users as data for learning.

Instead of using real conversations with actual users, dialogue interactions can be stochastically generated and reinforcement learning can be administered to these sets of dialogues [Levin and Pieraccini, 1997]. This approach takes much longer than ELVIS to learn optimal discourse strategies because it is stochastic and does not use what tend to be typical interactions with real users. This approach, unlike ELVIS, does not use TTS and ASR systems in the learning process and cannot account for these performance-related issues in learning. ELVIS measures performance in terms of user satisfaction, an external validation criterion, which is impossible with stochastic dialogue generation.

The next chapter concludes this thesis by reviewing the results, listing some future work, and discussing how to extend the presented methodology to automate the online

learning of optimal discourse strategies in spoken dialogue systems.

Chapter 6

Conclusions and Future Work

This chapter reviews the results of this thesis, lists some future work, and discusses how to extend the methodology presented in this thesis to automate the online learning of optimal discourse strategies in spoken dialogue systems.

6.1 Results

This thesis presented a spoken dialogue system that can learn optimal discourse strategies through interactions with users. This thesis described the spoken dialogue system, ELVIS, which supports access to email by telephone. Several discourse strategies for initiative distribution, summarizing email folders, and reading messages were implemented. Users interacted with discourse strategy-based variations of ELVIS in an experimental setting. These experimental results were presented and used to derive a general dialogue performance function.

The performance function was used in conjunction with reinforcement learning techniques to learn optimal discourse strategies in ELVIS. Optimal discourse strategies are learned using reinforcement learning techniques instead of other machine learning techniques because learning must be unsupervised; there are no “right” examples to provide as training data but reinforcement can be assigned to conversations and conversation states. ELVIS learned using the model-based learning algorithms of ADP and Q-learning because those learning algorithms converge faster than the

observance-based learning algorithms of TD learning and TD Q-learning; convergence speed is important when using human data instead of simulations because of the bounded amount of available data.

This thesis shows how ELVIS' state space representation affects the outcome of learning and can be transformed to yield a more appropriate state representation for learning optimal discourse strategies. The results of applying different reinforcement learning algorithms to the ELVIS data are presented and compared, showing that learning with the use of local reinforcement provides results that converge more quickly than learning without it.

6.2 Future Work

This section discusses some extensions to the system component and the learning component of ELVIS and then describes how to automate online learning of discourse strategies in a spoken dialogue system.

There are many ways to improve ELVIS as a spoken language-based email agent. Natural language is currently generated through the use of templates. ELVIS' utterances could be generated more flexibly by using a general-purpose natural language generation system, such as FUF [Elhadad, 1992]. The set of email functionality provided in ELVIS could be extended to allow, among other things, the deletion and composition of email messages.

The learning component of ELVIS can be enhanced and extended. More data could be collected to explore the effectiveness of the observance-based learning of the TD algorithms in comparison with the model-based reinforcement learning algorithms. Currently, ELVIS only learns for the general user in the context of different task progress levels; ELVIS can be extended to learn over various user models (such as novice versus expert) and different contexts (such as large email folders versus small email folders). Additionally, ELVIS could learn across applications if ELVIS were coupled with other agents.

The work described in this thesis can be broadened by automating online learning

in ELVIS in an act-learn cycle.

6.2.1 Automating Online Learning of Optimal Discourse Strategies

As currently implemented, ELVIS learns by interacting with users, deriving a performance function, applying the performance function to each of ELVIS' dialogues for final state utilities, applying the performance function to each of ELVIS' states for local reward assignments, and running the reinforcement learning algorithms. Currently, ELVIS cannot learn online as users interact with ELVIS.

To extend ELVIS to learn online, the performance function must consist of factors that can be measured automatically by ELVIS. An alternative performance function can be derived by substituting highly correlated and automatically measurable factors for those measures that cannot be automatically measured and then performing multivariate linear regression within the PARADISE framework on the new set of measures. For example, the recognition confidence score assigned by the ASR system to a recognized string correlates strongly with **Mean Recog**, so the confidence score can be substituted for **Mean Recog**.

A new local reward function with factors that could be measured directly by ELVIS at the dialogue state level could be derived. A state's local reward function could be updated each time that state is entered by storing state frequency statistics along with the state transition model and current local reward values.

The necessary transformations to ELVIS' state space for learning can automatically be produced. States could automatically be separated by prompt type. Strategy branching can be accomplished by inserting a new marked branch each time that a discourse strategy is first used. The representation of task progress can only be achieved through an estimation of task progress, as tasks will differ from user to user and from session to session. Every time a user reads a message, ELVIS can assume that one of the user's subtasks was completed, either because the user wanted to hear that message and retrieved useful information or because the user eliminated irrelevant

information from further attention.

In this way, ELVIS could be extended to learn optimal discourse strategies online while interacting with actual users in an act-learn cycle.

Appendix A

ELVIS States and State Transitions

States are represented as:

State Name: Optional System Function

*Condition*₁: Go to *State*₁

*Condition*₂: Go to *State*₂

⋮

*Condition*_{*n*}: Go to *State*_{*n*}

ELVIS has an associated set of allowable conditions for state transitions (such as *success*, *yes*, *no*, and *cancel*). Other named conditions for state transitions can also be used; some are domain-specific (such as *no-messages* and *messages*), while some are related to strategy parameters (such as *read-first*, *read-summarize-only*, and *read-choice-prompt*).

AnnounceElvis: SetUser

success: AnnounceWillGetMail

cancel: AnnounceElvisAgain

AnnounceElvisAgain: SetUser

success: AnnounceWillGetMail

cancel: AnnounceElvisAgain

AnnounceWillGetMail: GetMail

no-messages: AnnounceNoMsgs
messages: TopLevelUserOptions
AnnounceNoMessages: *No Function*
success: GoodBye
TopLevelUserOptions: *No Function*
chooseSI: TopLevelUserOptionsSI
chooseMI: TopLevelUserOptionsMI
TopLevelUserOptionsSI: TakeTopLevelActionSI
summarize-sender: SummCurrVfolderSenderSI
summarize-choice: GetWhichSummarySI
summarize-both: SummCurrVfolderSenderSubjectSI
read: ReadMessageSI
done: HangUpQuery
repeat-or-toggle: TopLevelUserOptionsSI
cancel: AnnounceElvisAgain
TopLevelUserOptionsMI: TakeTopLevelActionMI
read-one-message: ReadCurrentVfolderMI
pop: PopVfolderMI
readthem: TopLevelUserOptionsMI
summarize-choice: GetWhichSummaryMI
summarize-else: TopLevelUserOptionsMI
read-first: ReadCurrentVfolderMI-first
read-summarize-only: ReadCurrentVfolderMI-SO
read-choice-prompt: ReadCurrentVfolderMI-CP
toggle: TopLevelUserOptionsMI
cancel: AnnounceElvisAgain
GetWhichSummarySI: DetermineSummaryTypeAndSummarizeSI
sender: TopLevelUserOptionsSI
subject: TopLevelUserOptionsSI
both: TopLevelUserOptionsSI
cancel: TopLevelUserOptionsSI
GetWhichSummaryMI: DetermineSummaryTypeAndSummarizeMI
success: TopLevelUserOptionsMI

cancel: TopLevelUserOptionsMI

ReadMessageSI: TakeRefActionSI

sender: TakeSenderRefActionSI

subject: TakeSubjectRefActionSI

position: GetWhichRelativeSI

cancel: TopLevelUserOptionsSI

TakeSenderRefActionSI: BuildVfolderForSenderSI

no-messages: TopLevelUserOptionsSI

one-message: ReadCurrentVfolderSI

success-first: ReadCurrentVfolderSI-first

success-summarize-only: ReadCurrentVfolderSI-SO

success-choice-prompt: ReadCurrentVfolderSI-CP

cancel: ReadMessageSI

TakeSubjectRefActionSI: BuildVfolderForSubjectSI

no-messages: TopLevelUserOptionsSI

one-message: ReadCurrentVfolderSI

success-first: ReadCurrentVfolderSI-first

success-summarize-only: ReadCurrentVfolderSI-SO

success-choice-prompt: ReadCurrentVfolderSI-CP

cancel: ReadMessageSI

ReadCurrentVfolderSI: ReadMessageInLoopSI

success: ReadCurrentVfolderSI

cancel: PopVfolderSI

ReadCurrentVfolderSI-first: ReadMessageInLoopSI

success: ReadCurrentVfolderSI

cancel: PopVfolderSI

ReadCurrentVfolderSI-CP: ReadMessageInLoopSI

success: ReadCurrentVfolderSI

cancel: PopVfolderSI

ReadCurrentVfolderSI-SO: ReadMessageInLoopSI

success: ReadCurrentVfolderSI

cancel: PopVfolderSI

ReadCurrentVfolderMI: ReadMessageInLoopMI

success: ReadCurrentVfolderMI

cancel: PopVfolderMI

ReadCurrentVfolderMI-first: ReadMessageInLoopMI

success: ReadCurrentVfolderMI

cancel: PopVfolderMI

ReadCurrentVfolderMI-CP: ReadMessageInLoopMI

success: ReadCurrentVfolderMI

cancel: PopVfolderMI

ReadCurrentVfolderMI-SO: ReadMessageInLoopMI

success: ReadCurrentVfolderMI

cancel: PopVfolderMI

GetWhichRelativeSI: TakeRelativeActionSI

success: GetWhichRelativeSI

pop: TopLevelUserOptionsSI

cancel: ReadMessageSI

PopVfolderSI: *No Function*

top-level: HangUpQuery

non-top-level: TopLevelUserOptionsSI

PopVfolderMI: *No Function*

top-level: HangUpQuery

non-top-level: TopLevelUserOptionsMI

HangUpQuery: *No Function*

cancel: TopLevelUserOptions

yes: GoodBye

no: TopLevelUserOptions

Appendix B

Experimental Inbox Folders

B.1 Inbox Folder for Tutorial

From dana@research.att.com Mon Mar 17 17:23:37 1998

Return-Path: <michael@research.att.com>

Date: Mon, 17 Mar 1998 17:23:35 -0500 (EST)

Message-Id: <tut1>

From: Dana <dana@research.att.com>

To: Michael@research.att.com

Subject: technical memo

Status: 0

Hi Michael,

Jay has requested some changes to your technical memo before it can be released. Come by and I'll explain what has to be done.

Thanks,

Dana

From kim@research.att.com Tue Mar 18 09:43:34 1998

Return-Path: <kim@research.att.com>
Sender: kim@research.att.com
Message-Id: <tut2>
Date: Tue, 18 Mar 1998 09:43:31 -0500
From: Kim <kim@research.att.com>
Organization: AT&T Labs
To: Michael <Michael@research.att.com>
Subject: discourse book
Status: 0

The book you were looking for is called "Discourse Processing" and the author is Mary Jones.

Kim

From kim@walrus.com Wed Mar 12 15:53:25 1998
Return-Path: <kim@walrus.com>
Date: Wed, 12 Mar 1998 15:51:49 -0500 (EST)
From: Kim <kim@walrus.com>
To: Michael@research.att.com
Subject: call me tomorrow
Message-Id: <tut3>
Status: 0

I will be working at home tomorrow. Give me a call when you get a chance. The number is 201-555-8956.

Kim

From noah@research.att.com Mon Apr 7 07:05:18 1998

Return-Path: <noah@research.att.com>
Date: Mon, 7 Apr 1998 07:05:15 -0400 (EDT)
Message-Id: <tut4>
From: Noah <noah@research.att.com>
To: "Michael" <Michael@research.att.com>
Subject: interviewing
Status: 0

Hi Mike, sorry I won't be around to talk to the interview candidate. I'm going to be at a conference all week.

Noah

From dana@research.att.com Mon Mar 17 17:59:54 1998
Return-Path: <dana@research.att.com>
From: Dana <dana@research.att.com>
Message-Id: <tut5>
Date: Mon, 17 Mar 1998 17:57:37 -0500
To: "Michael" <Michael@research.att.com>
Subject: fax
Status: OR

Michael, I got a fax for you from President Clinton and dropped it off at your office.

Dana

B.2 Inbox Folder for Task 1

From michael@research.att.com Mon Mar 17 17:23:37 1998

Return-Path: <michael@research.att.com>
Date: Mon, 17 Mar 1998 17:23:35 -0500 (EST)
Message-Id: <t1-1>
From: Michael <michael@research.att.com>
To: Susan@research.att.com
Subject: evaluation technical memo
Status: 0

Hi Susan, can you look over the Evaluation TM today? I think it is just about ready to release.

Thanks,

Michael

From kim@research.att.com Tue Mar 18 09:43:34 1998
Return-Path: <kim@research.att.com>
Sender: kim@research.att.com
Message-Id: <t1-2>
Date: Tue, 18 Mar 1998 09:43:31 -0500
From: Kim <kim@research.att.com>
To: Susan <Susan@research.att.com>
Subject: interviewing antonio
Status: 0

I'd like to interview him. I could also go along to lunch.

Kim

From kim@walrus.com Wed Mar 12 15:53:25 1998

Return-Path: <kim@walrus.com>
Date: Wed, 12 Mar 1998 15:51:49 -0500 (EST)
From: Kim <kim@walrus.com>
To: Susan@research.att.com
Subject: meeting today
Message-Id: <t1-3>
Status: 0

The meeting this morning is at 10:00 in 2D516.

See you there,
Kim

From lee@research.att.com Mon Apr 7 07:05:18 1998
Return-Path: <lee@research.att.com>
Date: Mon, 7 Apr 1998 07:05:15 -0400 (EDT)
Message-Id: <t1-4>
From: Lee <lee@research.att.com>
To: "Susan" <Susan@research.att.com>
Subject: where i'll be
Status: 0

Hi,

In case you need to reach me I'm going to be working at home today.
The number there is 201-555-8888.

Lee

From dana@research.att.com Mon Mar 17 17:59:54 1998

Return-Path: <dana@research.att.com>
From: Dana <dana@research.att.com>
Message-Id: <t1-5>
Date: Mon, 17 Mar 1998 17:57:37 -0500
To: "Susan" <Susan@research.att.com>
Subject: fax
Status: OR

Susan, there's a fax for you from Jay.

Dana

From dana@research.att.com Mon Mar 17 17:59:54 1998
Return-Path: <dana@research.att.com>
From: Dana <dana@research.att.com>
Date: Mon, 17 Mar 1998 17:57:37 -0500
To: "Susan" <Susan@research.att.com>
Message-Id: <t1-6>
Subject: meeting today
Status: 0

Jay asked me to let everyone know that he won't be able to make the meeting today.

Dana

B.3 Inbox Folder for Task 2

From kim@walrus.com Wed Mar 12 15:53:25 1998
Return-Path: <kim@walrus.com>

Date: Wed, 12 Mar 1998 15:51:49 -0500 (EST)
From: Kim <kim@walrus.com>
To: jeannie@research.att.com
Subject: lunch
Message-Id: <t2-1>
Status: 0

Hi Jeannie, do you want to have lunch today? I could go at about 12:30.

Kim

From noah@research.att.com Tue Mar 18 08:25:29 1998
Return-Path: <noah@research.att.com>
Date: Tue, 18 Mar 1998 08:25:27 -0500 (EST)
Message-Id: <t2-2>
From: Noah <noah@research.att.com>
To: jeannie@research.att.com
Subject: interviewing antonio
Status: 0

I would be interested in interviewing him.

I'm fairly free that day. I think anytime between 10 AM and 4:30 PM would be okay. I could do lunch but not dinner.

Noah

From noah@research.att.com Tue Mar 18 08:41:51 1998
Return-Path: <noah@research.att.com>

From: Noah <noah@research.att.com>
Message-Id: <t2-3>
Subject: call tomorrow.
To: jeannie@research.att.com (Jeannie)
Date: Tue, 18 Mar 1998 08:41:35 -0500 (EST)
Status: 0

Hi,

When you get in tomorrow, would you give me a call? I'll be working at extension 6752.

Noah

From michael@research.att.com Mon Mar 17 17:23:37 1998
Return-Path: <michael@research.att.com>
Date: Mon, 17 Mar 1998 17:23:35 -0500 (EST)
Message-Id: <t2-4>
From: Michael <michael@research.att.com>
To: jeannie@research.att.com
Subject: evaluation group meeting
Status: 0

Hi - I have scheduled a room so that people interested in evaluation can get together and discuss work in progress. We'll get together on Friday, at 3:00, in 2B432.

Michael

From owen@research.att.com Mon Mar 17 17:22:26 1998
Return-Path: <owen@research.att.com>

From: Owen <owen@research.att.com>
Message-Id: <t2-5>
Date: Mon, 17 Mar 1998 17:21:20 -0500
To: "Jeannie" <jeannie@research.att.com>
Subject: agent personality
Status: OR

Can you let me know how to get a copy of your paper on agent personality?

Owen

From owen@research.att.com Mon Mar 17 17:23:37 1998
Return-Path: <michael@research.att.com>
Date: Mon, 17 Mar 1998 17:23:35 -0500 (EST)
Message-Id: <t2-6>
From: Owen <owen@research.att.com>
To: jeannie@research.att.com
Subject: call tomorrow
Status: 0

Hi Jeannie! I want to remind you that I will be setting up a conference call tomorrow morning between Candy, Jay, and myself around 10. Do you want me to relay anything to them from you?

Owen

B.4 Inbox Folder for Task 3

From lee@research.att.com Mon Apr 7 07:05:18 1998

Return-Path: <lee@research.att.com>
Date: Mon, 7 Apr 1998 07:05:15 -0400 (EDT)
Message-Id: <t3-1>
From: Lee <lee@research.att.com>
To: "Lawrence" <Lawrence@research.att.com>
Subject: I fixed the bug.
Status: OR

Hi Lawrence,

Thought you would be interested in knowing that I finally figured out what was causing the problem with sending arguments to the remote shell. Come by and I'll show you how I fixed it.

Lee

From noah@research.att.com Tue Mar 18 08:41:51 1998
Return-Path: <noah@research.att.com>
From: Noah <noah@research.att.com>
Message-Id: <t3-2>
Subject: I've got your book.
To: Lawrence@research.att.com (Lawrence)
Date: Tue, 18 Mar 1998 08:41:35 -0500 (EST)
Status: 0

Hi, I borrowed your nonparametric statistics book this morning. I'll bring it back later. Let me know if you need it.

Noah

From owen@research.att.com Mon Mar 17 17:22:26 1998

Return-Path: <owen@research.att.com>

From: Owen <owen@research.att.com>

Message-Id: <t3-3>

Date: Mon, 17 Mar 1998 17:21:20 -0500

To: "Lawrence" <Lawrence@research.att.com>

Subject: Discourse Discussion Group

Status: 0

I won't be able to make it all morning. How about meeting at 4 in my office?

Owen

From dana@research.att.com Mon Mar 17 18:20:39 1998

Return-Path: <dana@research.att.com>

From: Dana <dana@research.att.com>

Date: Mon, 17 Mar 1998 18:20:37 -0500 (EST)

Message-Id: <t3-4>

To: Lawrence@research.att.com

Subject: your performance objectives

Status: 0

Hi,

Remember that your performance objectives are due to me today. You can get the form from me if you've misplaced it.

Dana

From dana@research.att.com Mon Mar 17 17:59:54 1998

Return-Path: <dana@research.att.com>

From: Dana <dana@research.att.com>

Message-Id: <t3-5>

Date: Mon, 17 Mar 1998 17:57:37 -0500

To: "Lawrence" <Lawrence@research.att.com>

Subject: message

Status: 0

Hi,

You got a message while you were offsite earlier. Caroline wants you to call her at 614-653-7900.

Dana

From noah@research.att.com Tue Mar 18 08:41:51 1998

Return-Path: <noah@research.att.com>

From: Noah <noah@research.att.com>

Message-Id: <t3-6>

Subject: message

To: Lawrence@research.att.com (Lawrence)

Date: Tue, 18 Mar 1998 08:41:35 -0500 (EST)

Status: 0

Can you leave me a message about whether you are coming to the department picnic?

Noah

Appendix C

Experimental Task Scenarios

C.1 Task Scenario for Tutorial

In the current task, we ask you to pretend that you are a user named Michael, and that you are accessing your email inbox by phone.

The scenario for the Tutorial has two subtasks that you must do in the same conversation with Elvis. You should do both tasks together. Do not hang up the phone between tasks.

(Subtask 0.1) Find out the **telephone number** in the message about *call me tomorrow*.

(Subtask 0.2) You have received a fax. Find out **who the fax is from**.

C.2 Task Scenario for Task 1

In the current task, we ask you to pretend that you are a user named Susan, and that you are accessing your email inbox by phone.

The scenario for Task 1 has two subtasks that you must do in the same conversation with Elvis. You should do both tasks together. Do not hang up the phone between tasks.

| Attribute | Actual Value |
|--------------------|-------------------------|
| Selection Criteria | <i>Call me tomorrow</i> |
| Telephone Number | 201-555-8956 |

Table C.1: Attribute-Value Matrix Key for Subtask 0.1

| Attribute | Actual Value |
|--------------------|-------------------|
| Selection Criteria | Fax |
| Fax Sender | President Clinton |

Table C.2: Attribute-Value Matrix Key for Subtask 0.2

(Subtask 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**.

(Subtask 1.2) The second task involves finding information in a different message. Yesterday evening, you had told Lee you might want to call him this morning. Lee said he would send you a message telling you where to reach him. Find out **Lee's Phone Number**.

C.3 Task Scenario for Task 2

In the current task, we ask you to pretend that you are a user named Jeannie, and that you are accessing your email inbox by phone.

The scenario for Task 2 has two subtasks that you must do in the same conversation with Elvis. You should do both tasks together. Do not hang up the phone between tasks.

(Subtask 2.1) When you got into work, you went directly to a meeting. Since some people were late, you've decided to call Elvis to check your mail to see what

| Attribute | Actual Value |
|--------------------|---------------|
| Selection Criteria | Kim ∨ Meeting |
| Meeting Time | 10:30 |
| Meeting Place | 2D516 |

Table C.3: Attribute-Value Matrix Key for Subtask 1.1

| Attribute | Actual Value |
|------------------------|--------------|
| Selection Criteria | Lee |
| Lee's Telephone Number | 201-555-8888 |

Table C.4: Attribute-Value Matrix Key for Subtask 1.2

other meetings may have been scheduled. Find out the **day**, **place**, and **time** of any scheduled meetings.

(Subtask 2.2) The second task involves finding information in a different message. Find out if you need to call anyone. If so, find out the **number to call**.

C.4 Task Scenario for Task 3

In the current task, we ask you to pretend that you are a user named Lawrence, and that you are accessing your email inbox by phone.

The scenario for Task 3 has two subtasks that you must do in the same conversation with Elvis. You should do both tasks together. Do not hang up the phone between tasks.

(Subtask 3.1) You are expecting a message telling you when the Discourse Discussion Group can meet. Find out the **place** and **time of the meeting**.

(Subtask 3.2) The second task involves finding information in a different message. Your secretary has taken a phone call for you and left you a message. Find out **who called** and the **phone number** where you can reach them.

| Attribute | Actual Value |
|--------------------|--------------|
| Selection Criteria | Meeting |
| Meeting Day | Friday |
| Meeting Time | 3:00 |
| Meeting Place | 2B432 |

Table C.5: Attribute-Value Matrix Key for Subtask 2.1

| Attribute | Actual Value |
|--------------------|--------------|
| Selection Criteria | Call |
| Telephone Number | x6752 |

Table C.6: Attribute-Value Matrix Key for Subtask 2.2

| Attribute | Actual Value |
|--------------------|----------------------------|
| Selection Criteria | Discourse Discussion Group |
| Meeting Place | Owen's office |
| Meeting Time | 4:00 |

Table C.7: Attribute-Value Matrix Key for Subtask 3.1

| Attribute | Actual Value |
|--------------------|--------------|
| Selection Criteria | Message |
| Who Called | Caroline |
| Telephone Number | 614-653-7900 |

Table C.8: Attribute-Value Matrix Key for Subtask 3.2

Appendix D

Experimental User Survey

1. Login Name (e.g. fromer): _____
2. Did you complete the task and get the information you needed?
 - (a) Yes
 - (b) No
3. Was Elvis easy to understand in this conversation?
 - (a) Very Easy
 - (b) Somewhat Easy
 - (c) Neither Easy nor Difficult
 - (d) Somewhat Difficult
 - (e) Very Difficult
4. In this conversation, did Elvis understand what you said?
 - (a) Almost Never
 - (b) Rarely
 - (c) Sometimes
 - (d) Often

(e) Almost Always

5. Was it easy to find the message you wanted in this conversation?

(a) Very Easy

(b) Somewhat Easy

(c) Neither Easy nor Difficult

(d) Somewhat Difficult

(e) Very Difficult

6. Was the pace of interaction with Elvis appropriate in this conversation?

(a) No, the pace was almost always too fast for me

(b) No, sometimes the pace was too fast for me

(c) The pace was just right

(d) No, sometimes the pace was too slow

(e) No, the pace was almost always too slow

7. In this conversation, did you know what you could say at each point of the dialogue?

(a) Almost Never

(b) Rarely

(c) Sometimes

(d) Often

(e) Almost Always

8. How often was Elvis sluggish and slow to reply to you in this conversation?

(a) Almost Never

(b) Rarely

- (c) Sometimes
- (d) Often
- (e) Almost Always

9. In this conversation, did Elvis work the way you expected him to?

- (a) Yes
- (b) No

10. In this conversation, how did Elvis's voice interface compare to touch-tone interfaces to voice mail?

- (a) Much Better
- (b) Better
- (c) The Same
- (d) Worse
- (e) Much Worse

11. From your current experience with using Elvis to get your email, do you think you use Elvis regularly to access your mail when you are away from your desk?

- (a) Yes
- (b) Maybe
- (c) No

12. Please provide any additional comments you have about Elvis:

Appendix E

Data Tables

| Trial | System-Initiative No Local Reward | Mixed-Initiative No Local Reward | System-Initiative With Local Reward | Mixed-Initiative With Local Reward |
|-------|--------------------------------------|-------------------------------------|--|---------------------------------------|
| 0 | 0 | 0 | 0 | 0 |
| 5 | .365 | -.006 | -1.202 | -2.974 |
| 10 | .383 | -.043 | -1.301 | -4.378 |
| 15 | .301 | -.094 | -.939 | -3.106 |
| 20 | .259 | -.078 | -.789 | -2.603 |
| 25 | .259 | -.044 | -1.144 | -2.221 |
| 30 | .278 | -.039 | -1.143 | -2.069 |
| 35 | .3 | -.112 | -1.294 | -2.034 |
| 40 | .31 | -.163 | -1.482 | -2.037 |
| 45 | .301 | -.209 | -1.6 | -1.902 |
| 50 | .299 | -.235 | -1.83 | -2.06 |
| 55 | .294 | -.226 | -1.84 | -2.072 |
| 60 | .29 | -.238 | -1.759 | -2.232 |
| 65 | .284 | -.261 | -1.595 | -2.402 |
| 70 | .281 | -.271 | -1.478 | -2.442 |
| 75 | .277 | -.276 | -1.242 | -2.43 |
| 80 | .272 | -.272 | -1.189 | -2.389 |
| 85 | .268 | -.261 | -1.123 | -2.355 |
| 90 | .263 | -.253 | -1.14 | -2.353 |
| 95 | .255 | -.248 | -1.223 | -2.325 |
| 100 | .25 | -.247 | -1.262 | -2.293 |
| 105 | .246 | -.249 | -1.288 | -2.312 |
| 108 | .247 | -.252 | -1.291 | -2.338 |

Table E.1: Utilities for Initiative Strategies Learned via ADP

| Trial | System-Initiative No Local Reward | Mixed-Initiative No Local Reward | System-Initiative With Local Reward | Mixed-Initiative With Local Reward |
|-------|--------------------------------------|-------------------------------------|--|---------------------------------------|
| 0 | 0 | 0 | 0 | 0 |
| 5 | .359 | 0 | -1.193 | -2.863 |
| 10 | .384 | -.043 | -1.279 | -3.108 |
| 15 | .304 | -.091 | -.964 | -2.607 |
| 20 | .26 | -.081 | -.783 | -2.227 |
| 25 | .26 | -.046 | -1.136 | -2.059 |
| 30 | .274 | -.038 | -1.145 | -2.059 |
| 35 | .3 | -.107 | -1.281 | -2.042 |
| 40 | .308 | -.159 | -1.49 | -2.036 |
| 45 | .303 | -.205 | -1.606 | -2.088 |
| 50 | .3 | -.235 | -1.83 | -2.061 |
| 55 | .295 | -.226 | -1.852 | -2.063 |
| 60 | .291 | -.237 | -1.767 | -2.222 |
| 65 | .284 | -.26 | -1.604 | -2.398 |
| 70 | .28 | -.271 | -1.501 | -2.44 |
| 75 | .281 | -.276 | -1.24 | -2.43 |
| 80 | .272 | -.273 | -1.189 | -2.388 |
| 85 | .267 | -.26 | -1.13 | -2.339 |
| 90 | .263 | -.253 | -1.13 | -2.344 |
| 95 | .256 | -.248 | -1.206 | -2.32 |
| 100 | .25 | -.247 | -1.262 | -2.29 |
| 105 | .246 | -.248 | -1.285 | -2.301 |
| 108 | .247 | -.251 | -1.291 | -2.325 |

Table E.2: Utilities for Initiative Strategies Learned via Q-Learning

| Trial | Task Progress=1 | Task Progress=2 | Task Progress=3 | Task Progress=1 | Task Progress=2 | Task Progress=3 |
|-------|-----------------|-----------------|-----------------|-------------------|-------------------|-------------------|
| | No Local Reward | No Local Reward | No Local Reward | With Local Reward | With Local Reward | With Local Reward |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0.726 | 0 | 0 |
| 15 | 0.256 | 0 | 0 | 0.405 | 0 | 0 |
| 20 | 0.067 | 0 | 0.165 | 0.701 | 0 | 2.58 |
| 25 | 0.1 | 0.141 | 0.278 | 1.357 | 2.391 | 2.351 |
| 30 | 0.131 | 0.189 | 0.288 | 1.429 | 2.484 | 2.268 |
| 35 | 0.164 | 0.222 | 0.298 | 1.739 | 2.942 | 2.353 |
| 40 | 0.245 | 0.263 | 0.284 | 1.913 | 3.103 | 2.415 |
| 45 | 0.258 | 0.252 | 0.264 | 1.768 | 2.976 | 2.27 |
| 50 | 0.245 | 0.218 | 0.259 | 1.412 | 2.774 | 2.199 |
| 55 | 0.229 | 0.211 | 0.241 | 1.191 | 2.74 | 2.171 |
| 60 | 0.221 | 0.203 | 0.249 | 1.133 | 2.792 | 2.297 |
| 65 | 0.213 | 0.211 | 0.239 | 1.215 | 3.132 | 2.322 |
| 70 | 0.212 | 0.211 | 0.216 | 1.297 | 3.157 | 2.253 |
| 75 | 0.212 | 0.209 | 0.198 | 1.352 | 3.176 | 2.319 |
| 80 | 0.211 | 0.199 | 0.191 | 1.406 | 3.211 | 2.33 |
| 85 | 0.21 | 0.19 | 0.179 | 1.415 | 3.229 | 2.352 |
| 90 | 0.2 | 0.177 | 0.167 | 1.489 | 3.322 | 2.624 |
| 95 | 0.194 | 0.165 | 0.16 | 1.558 | 3.489 | 2.659 |
| 100 | 0.183 | 0.156 | 0.128 | 1.668 | 3.594 | 2.57 |
| 105 | 0.16 | 0.127 | 0.09 | 1.738 | 3.589 | 2.55 |
| 110 | 0.146 | 0.107 | 0.064 | 1.702 | 3.544 | 2.505 |
| 115 | 0.126 | 0.086 | 0.042 | 1.697 | 3.565 | 2.485 |
| 120 | 0.109 | 0.07 | 0.033 | 1.712 | 3.545 | 2.503 |
| 124 | 0.101 | 0.088 | 0.032 | 1.709 | 3.547 | 2.491 |

Table E.3: Utilities for Summarize-Both Summarize Strategy Learned via ADP

| Trial | Task Progress=1 | Task Progress=2 | Task Progress=3 | Task Progress=1 | Task Progress=2 | Task Progress=3 |
|-------|-----------------|-----------------|-----------------|-------------------|-------------------|-------------------|
| | No Local Reward | No Local Reward | No Local Reward | With Local Reward | With Local Reward | With Local Reward |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | -0.397 | 0.562 | 0 | -2.103 | 2.76 | 0 |
| 15 | -0.719 | 0.58 | 0 | -2.994 | 2.76 | 0 |
| 20 | -0.402 | 0.567 | 0 | -1.925 | 2.465 | 0 |
| 25 | -0.153 | 0.142 | 0.482 | -1.737 | 2.174 | 2.154 |
| 30 | -0.225 | 0.101 | 0.527 | -1.284 | 3.852 | 2.661 |
| 35 | -0.245 | 0.155 | 0.524 | -0.405 | 3.048 | 2.217 |
| 40 | -0.154 | 0.237 | 0.414 | -0.338 | 2.833 | 2.448 |
| 45 | -0.086 | 0.228 | 0.446 | -0.099 | 3.076 | 2.525 |
| 50 | -0.037 | 0.247 | 0.451 | 0.183 | 3.091 | 2.412 |
| 55 | -0.022 | 0.272 | 0.45 | 0.162 | 3.025 | 2.278 |
| 60 | 0.04 | 0.263 | 0.39 | 0.202 | 2.779 | 2.24 |
| 65 | 0.05 | 0.234 | 0.33 | -0.073 | 2.75 | 2.201 |
| 70 | 0.056 | 0.226 | 0.347 | -0.015 | 2.762 | 2.153 |
| 75 | 0.068 | 0.212 | 0.348 | 0.064 | 2.719 | 2.212 |
| 80 | 0.074 | 0.217 | 0.348 | 0.135 | 2.824 | 2.206 |
| 85 | 0.074 | 0.223 | 0.343 | 0.154 | 2.819 | 2.204 |
| 90 | 0.089 | 0.229 | 0.352 | 0.249 | 2.822 | 2.231 |
| 95 | 0.103 | 0.236 | 0.34 | 0.282 | 2.77 | 2.293 |
| 100 | 0.115 | 0.234 | 0.307 | 0.352 | 2.864 | 2.363 |
| 105 | 0.127 | 0.209 | 0.204 | 0.442 | 2.946 | 2.291 |
| 110 | 0.133 | 0.15 | 0.184 | 0.529 | 2.883 | 2.221 |
| 115 | 0.1 | 0.103 | 0.172 | 0.39 | 2.795 | 2.154 |
| 120 | 0.024 | 0.091 | 0.177 | 0.324 | 2.755 | 2.093 |
| 124 | 0.01 | 0.088 | 0.178 | 0.321 | 2.751 | 2.08 |

Table E.4: Utilities for Summarize-Choice Summarize Strategy Learned via ADP

| Trial | Task Progress=1 | Task Progress=2 | Task Progress=3 | Task Progress=1 | Task Progress=2 | Task Progress=3 |
|-------|-----------------|-----------------|-----------------|-------------------|-------------------|-------------------|
| | No Local Reward | No Local Reward | No Local Reward | With Local Reward | With Local Reward | With Local Reward |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0.019 | 0.302 | 2.283 | 3.598 | 2.313 |
| 10 | 0.407 | 0.455 | 0.377 | 2.034 | 3.674 | 2.295 |
| 15 | 0.029 | 0.344 | 0.34 | 1.236 | 3.489 | 2.198 |
| 20 | 0.094 | 0.328 | 0.299 | 1.516 | 3.515 | 2.373 |
| 25 | 0.096 | 0.276 | 0.264 | 1.614 | 3.696 | 2.353 |
| 30 | 0.088 | 0.253 | 0.242 | 1.493 | 3.65 | 2.205 |
| 35 | 0.081 | 0.241 | 0.248 | 1.551 | 3.63 | 2.117 |
| 40 | 0.074 | 0.243 | 0.237 | 1.53 | 3.56 | 2.021 |
| 45 | 0.091 | 0.238 | 0.22 | 1.497 | 3.546 | 2.263 |
| 50 | 0.097 | 0.225 | 0.219 | 1.668 | 3.804 | 2.355 |
| 55 | 0.093 | 0.191 | 0.212 | 1.685 | 3.554 | 2.359 |
| 60 | 0.078 | 0.172 | 0.199 | 1.43 | 3.596 | 2.302 |
| 65 | 0.058 | 0.158 | 0.196 | 1.447 | 3.562 | 2.383 |
| 70 | 0.053 | 0.157 | 0.195 | 1.438 | 3.621 | 2.465 |
| 75 | 0.057 | 0.158 | 0.211 | 1.476 | 3.743 | 2.333 |
| 80 | 0.059 | 0.162 | 0.229 | 1.652 | 3.668 | 2.316 |
| 85 | 0.065 | 0.167 | 0.24 | 1.666 | 3.666 | 2.303 |
| 90 | 0.072 | 0.185 | 0.252 | 1.623 | 3.612 | 2.188 |
| 95 | 0.085 | 0.201 | 0.259 | 1.574 | 3.567 | 2.121 |
| 100 | 0.099 | 0.212 | 0.258 | 1.613 | 3.496 | 2.065 |
| 105 | 0.127 | 0.222 | 0.248 | 1.635 | 3.42 | 2.044 |
| 110 | 0.115 | 0.219 | 0.255 | 1.559 | 3.411 | 2.083 |
| 115 | 0.11 | 0.222 | 0.255 | 1.543 | 3.43 | 2.107 |
| 120 | 0.115 | 0.225 | 0.256 | 1.532 | 3.472 | 2.193 |
| 124 | 0.115 | 0.225 | 0.256 | 1.542 | 3.497 | 2.205 |

Table E.5: Utilities for Summarize-System Summarize Strategy Learned via ADP

| Trial | Task Progress=1 | Task Progress=2 | Task Progress=3 | Task Progress=1 | Task Progress=2 | Task Progress=3 |
|-------|-----------------|-----------------|-----------------|-------------------|-------------------|-------------------|
| | No Local Reward | No Local Reward | No Local Reward | With Local Reward | With Local Reward | With Local Reward |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0.726 | 0 | 0 |
| 15 | 0.256 | 0 | 0 | 0.405 | 0 | 0 |
| 20 | 0.067 | 0 | 0.165 | 0.701 | 0 | 2.58 |
| 25 | 0.1 | 0.141 | 0.278 | 1.357 | 2.391 | 2.351 |
| 30 | 0.131 | 0.189 | 0.288 | 1.429 | 2.484 | 2.268 |
| 35 | 0.164 | 0.222 | 0.298 | 1.739 | 2.942 | 2.353 |
| 40 | 0.245 | 0.263 | 0.284 | 1.913 | 3.103 | 2.415 |
| 45 | 0.258 | 0.252 | 0.264 | 1.768 | 2.976 | 2.27 |
| 50 | 0.245 | 0.218 | 0.259 | 1.412 | 2.774 | 2.199 |
| 55 | 0.229 | 0.211 | 0.241 | 1.191 | 2.74 | 2.171 |
| 60 | 0.221 | 0.203 | 0.249 | 1.133 | 2.792 | 2.297 |
| 65 | 0.213 | 0.211 | 0.239 | 1.215 | 3.132 | 2.322 |
| 70 | 0.212 | 0.211 | 0.216 | 1.297 | 3.157 | 2.253 |
| 75 | 0.212 | 0.209 | 0.198 | 1.352 | 3.176 | 2.319 |
| 80 | 0.211 | 0.199 | 0.191 | 1.406 | 3.211 | 2.33 |
| 85 | 0.21 | 0.19 | 0.179 | 1.415 | 3.229 | 2.352 |
| 90 | 0.2 | 0.177 | 0.167 | 1.489 | 3.322 | 2.624 |
| 95 | 0.194 | 0.165 | 0.16 | 1.558 | 3.489 | 2.659 |
| 100 | 0.183 | 0.156 | 0.128 | 1.668 | 3.594 | 2.57 |
| 105 | 0.16 | 0.127 | 0.09 | 1.738 | 3.589 | 2.55 |
| 110 | 0.146 | 0.107 | 0.064 | 1.702 | 3.544 | 2.505 |
| 115 | 0.126 | 0.086 | 0.042 | 1.697 | 3.565 | 2.485 |
| 120 | 0.109 | 0.07 | 0.033 | 1.712 | 3.545 | 2.503 |
| 124 | 0.101 | 0.088 | 0.032 | 1.709 | 3.547 | 2.491 |

Table E.6: Utilities for Summarize-Both Summarize Strategy Learned via Q-Learning

| Trial | Task Progress=1 | Task Progress=2 | Task Progress=3 | Task Progress=1 | Task Progress=2 | Task Progress=3 |
|-------|-----------------|-----------------|-----------------|-------------------|-------------------|-------------------|
| | No Local Reward | No Local Reward | No Local Reward | With Local Reward | With Local Reward | With Local Reward |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | -0.397 | 0.562 | 0 | -2.103 | 2.76 | 0 |
| 15 | -0.719 | 0.58 | 0 | -2.994 | 2.76 | 0 |
| 20 | -0.402 | 0.567 | 0 | -1.925 | 2.465 | 0 |
| 25 | -0.153 | 0.142 | 0.482 | -1.737 | 2.174 | 2.154 |
| 30 | -0.225 | 0.101 | 0.527 | -1.284 | 3.852 | 2.661 |
| 35 | -0.245 | 0.155 | 0.524 | -0.405 | 3.048 | 2.217 |
| 40 | -0.154 | 0.237 | 0.414 | -0.338 | 2.833 | 2.448 |
| 45 | -0.086 | 0.228 | 0.446 | -0.099 | 3.076 | 2.525 |
| 50 | -0.037 | 0.247 | 0.451 | 0.183 | 3.091 | 2.412 |
| 55 | -0.022 | 0.272 | 0.45 | 0.162 | 3.025 | 2.278 |
| 60 | 0.04 | 0.263 | 0.39 | 0.202 | 2.779 | 2.24 |
| 65 | 0.05 | 0.234 | 0.33 | -0.073 | 2.75 | 2.201 |
| 70 | 0.056 | 0.226 | 0.347 | -0.015 | 2.762 | 2.153 |
| 75 | 0.068 | 0.212 | 0.348 | 0.064 | 2.719 | 2.212 |
| 80 | 0.074 | 0.217 | 0.348 | 0.135 | 2.824 | 2.206 |
| 85 | 0.074 | 0.223 | 0.343 | 0.154 | 2.819 | 2.204 |
| 90 | 0.089 | 0.229 | 0.352 | 0.249 | 2.822 | 2.231 |
| 95 | 0.103 | 0.236 | 0.34 | 0.282 | 2.77 | 2.293 |
| 100 | 0.115 | 0.234 | 0.307 | 0.352 | 2.864 | 2.363 |
| 105 | 0.127 | 0.209 | 0.204 | 0.442 | 2.946 | 2.291 |
| 110 | 0.133 | 0.15 | 0.184 | 0.529 | 2.883 | 2.221 |
| 115 | 0.1 | 0.103 | 0.172 | 0.39 | 2.795 | 2.154 |
| 120 | 0.024 | 0.091 | 0.177 | 0.324 | 2.755 | 2.093 |
| 124 | 0.01 | 0.088 | 0.178 | 0.321 | 2.751 | 2.08 |

Table E.7: Utilities for Summarize-Choice Summarize Strategy Learned via Q-Learning

| Trial | Task Progress=1 | Task Progress=2 | Task Progress=3 | Task Progress=1 | Task Progress=2 | Task Progress=3 |
|-------|-----------------|-----------------|-----------------|-------------------|-------------------|-------------------|
| | No Local Reward | No Local Reward | No Local Reward | With Local Reward | With Local Reward | With Local Reward |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0.019 | 0.302 | 2.283 | 3.598 | 2.313 |
| 10 | 0.407 | 0.455 | 0.377 | 2.034 | 3.674 | 2.295 |
| 15 | 0.029 | 0.344 | 0.34 | 1.236 | 3.489 | 2.198 |
| 20 | 0.094 | 0.328 | 0.299 | 1.516 | 3.515 | 2.373 |
| 25 | 0.096 | 0.276 | 0.264 | 1.614 | 3.696 | 2.353 |
| 30 | 0.088 | 0.253 | 0.242 | 1.493 | 3.65 | 2.205 |
| 35 | 0.081 | 0.241 | 0.248 | 1.551 | 3.63 | 2.117 |
| 40 | 0.074 | 0.243 | 0.237 | 1.53 | 3.56 | 2.021 |
| 45 | 0.091 | 0.238 | 0.22 | 1.497 | 3.546 | 2.263 |
| 50 | 0.097 | 0.225 | 0.219 | 1.668 | 3.804 | 2.355 |
| 55 | 0.093 | 0.191 | 0.212 | 1.685 | 3.554 | 2.359 |
| 60 | 0.078 | 0.172 | 0.199 | 1.43 | 3.596 | 2.302 |
| 65 | 0.058 | 0.158 | 0.196 | 1.447 | 3.562 | 2.383 |
| 70 | 0.053 | 0.157 | 0.195 | 1.438 | 3.621 | 2.465 |
| 75 | 0.057 | 0.158 | 0.211 | 1.476 | 3.743 | 2.333 |
| 80 | 0.059 | 0.162 | 0.229 | 1.652 | 3.668 | 2.316 |
| 85 | 0.065 | 0.167 | 0.24 | 1.666 | 3.666 | 2.303 |
| 90 | 0.072 | 0.185 | 0.252 | 1.623 | 3.612 | 2.188 |
| 95 | 0.085 | 0.201 | 0.259 | 1.574 | 3.567 | 2.121 |
| 100 | 0.099 | 0.212 | 0.258 | 1.613 | 3.496 | 2.065 |
| 105 | 0.127 | 0.222 | 0.248 | 1.635 | 3.42 | 2.044 |
| 110 | 0.115 | 0.219 | 0.255 | 1.559 | 3.411 | 2.083 |
| 115 | 0.11 | 0.222 | 0.255 | 1.543 | 3.43 | 2.107 |
| 120 | 0.115 | 0.225 | 0.256 | 1.532 | 3.472 | 2.193 |
| 124 | 0.115 | 0.225 | 0.256 | 1.542 | 3.497 | 2.205 |

Table E.8: Utilities for Summarize-System Summarize Strategy Learned via Q-Learning

| Trial | Task Progress=1 | Task Progress=2 | Task Progress=3 | Task Progress=1 | Task Progress=2 | Task Progress=3 |
|-------|-----------------|-----------------|-----------------|-------------------|-------------------|-------------------|
| | No Local Reward | No Local Reward | No Local Reward | With Local Reward | With Local Reward | With Local Reward |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0.22 | 0 | 0 | 0.635 | 0 |
| 15 | 0 | 0.281 | 0 | 0 | 0.704 | 0 |
| 20 | 0.232 | 0.211 | 0.118 | 1.982 | 2.933 | 5.888 |
| 25 | 0.21 | 0.214 | 0.188 | 2.593 | 2.804 | 5.769 |
| 30 | 0.192 | 0.251 | 0.231 | 1.573 | 2.324 | 5.238 |
| 35 | 0.224 | 0.274 | 0.269 | 1.914 | 2.346 | 5.307 |
| 40 | 0.28 | 0.257 | 0.265 | 1.851 | 2.275 | 5.248 |
| 45 | 0.273 | 0.239 | 0.244 | 1.766 | 2.25 | 5.222 |
| 50 | 0.231 | 0.228 | 0.233 | 1.549 | 2.211 | 5.207 |
| 55 | 0.21 | 0.236 | 0.237 | 1.53 | 2.12 | 5.127 |
| 60 | 0.209 | 0.238 | 0.24 | 1.53 | 2.154 | 5.137 |
| 65 | 0.208 | 0.239 | 0.243 | 1.533 | 2.243 | 5.184 |
| 70 | 0.215 | 0.237 | 0.241 | 1.586 | 2.248 | 5.187 |
| 75 | 0.215 | 0.234 | 0.238 | 1.647 | 2.303 | 5.236 |
| 80 | 0.219 | 0.225 | 0.23 | 1.722 | 2.252 | 5.208 |
| 85 | 0.212 | 0.206 | 0.213 | 1.67 | 2.247 | 5.199 |
| 90 | 0.208 | 0.194 | 0.194 | 1.766 | 2.538 | 4.516 |
| 95 | 0.199 | 0.186 | 0.185 | 1.894 | 2.582 | 4.541 |
| 100 | 0.188 | 0.169 | 0.165 | 2.049 | 2.545 | 4.46 |
| 105 | 0.171 | 0.103 | 0.09 | 2.036 | 2.411 | 4.327 |
| 110 | 0.153 | 0.074 | 0.061 | 1.993 | 2.362 | 4.285 |
| 115 | 0.116 | 0.052 | 0.043 | 1.925 | 2.362 | 4.255 |
| 120 | 0.085 | 0.043 | 0.038 | 1.91 | 2.39 | 4.283 |
| 124 | 0.073 | 0.04 | 0.037 | 1.919 | 2.394 | 4.285 |

Table E.9: Utilities for Read-Choice-Prompt Read Strategy Learned via ADP

| Trial | Task Progress=1 | Task Progress=2 | Task Progress=3 | Task Progress=1 | Task Progress=2 | Task Progress=3 |
|-------|-----------------|-----------------|-----------------|-------------------|-------------------|-------------------|
| | No Local Reward | No Local Reward | No Local Reward | With Local Reward | With Local Reward | With Local Reward |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0.004 | 0 | 0 | 3.395 | 0 | 0 |
| 10 | 0.432 | 0.432 | 0 | 3.356 | 2.616 | 0 |
| 15 | 0.369 | 0.315 | 0 | 3.163 | 2.463 | 0 |
| 20 | 0.316 | 0.304 | 0.243 | 3.425 | 2.635 | 3.684 |
| 25 | 0.304 | 0.254 | 0.256 | 3.176 | 2.358 | 3.494 |
| 30 | 0.283 | 0.236 | 0.235 | 3.109 | 2.368 | 3.324 |
| 35 | 0.24 | 0.223 | 0.223 | 3.182 | 2.268 | 3.246 |
| 40 | 0.219 | 0.224 | 0.224 | 3.176 | 2.266 | 3.242 |
| 45 | 0.22 | 0.219 | 0.221 | 3.151 | 2.364 | 3.324 |
| 50 | 0.22 | 0.143 | 0.141 | 3.164 | 2.368 | 3.237 |
| 55 | 0.167 | 0.091 | 0.097 | 3.069 | 2.237 | 3.2 |
| 60 | 0.134 | 0.077 | 0.086 | 2.972 | 2.189 | 3.174 |
| 65 | 0.096 | 0.074 | 0.083 | 2.819 | 2.168 | 2.738 |
| 70 | 0.082 | 0.077 | 0.087 | 2.794 | 2.334 | 2.871 |
| 75 | 0.074 | 0.088 | 0.099 | 2.922 | 2.333 | 2.849 |
| 80 | 0.074 | 0.115 | 0.128 | 2.974 | 2.295 | 2.822 |
| 85 | 0.085 | 0.147 | 0.161 | 2.891 | 2.274 | 2.815 |
| 90 | 0.108 | 0.162 | 0.175 | 2.893 | 2.23 | 2.773 |
| 95 | 0.132 | 0.179 | 0.188 | 2.891 | 2.177 | 2.735 |
| 100 | 0.153 | 0.192 | 0.202 | 2.817 | 2.112 | 2.674 |
| 105 | 0.177 | 0.183 | 0.192 | 2.749 | 2.082 | 2.644 |
| 110 | 0.179 | 0.187 | 0.196 | 2.736 | 2.117 | 2.68 |
| 115 | 0.178 | 0.192 | 0.201 | 2.718 | 2.137 | 2.715 |
| 120 | 0.182 | 0.204 | 0.213 | 2.753 | 2.197 | 2.756 |
| 124 | 0.184 | 0.209 | 0.217 | 2.767 | 2.2 | 2.759 |

Table E.10: Utilities for Read-First Read Strategy Learned via ADP

| Trial | Task Progress=1 | Task Progress=2 | Task Progress=3 | Task Progress=1 | Task Progress=2 | Task Progress=3 |
|-------|-----------------|-----------------|-----------------|-------------------|-------------------|-------------------|
| | No Local Reward | No Local Reward | No Local Reward | With Local Reward | With Local Reward | With Local Reward |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0.56 | 0 | 0 | 2.135 | 0 | 0 |
| 15 | 0.376 | 0 | 0 | 1.137 | 0 | 0 |
| 20 | 0.161 | 0 | 0 | 1.105 | 0 | 0 |
| 25 | 0.242 | 0 | 0 | 1.153 | 0 | 0 |
| 30 | 0.244 | 0 | 0 | 2.198 | 0 | 0 |
| 35 | 0.271 | 0.341 | 0 | 2.195 | 1.545 | 0 |
| 40 | 0.336 | 0.332 | 0.246 | 2.116 | 1.764 | 3.308 |
| 45 | 0.338 | 0.338 | 0.329 | 2.308 | 1.896 | 3.531 |
| 50 | 0.322 | 0.358 | 0.356 | 2.323 | 1.836 | 3.48 |
| 55 | 0.349 | 0.348 | 0.363 | 2.335 | 1.813 | 3.449 |
| 60 | 0.355 | 0.324 | 0.329 | 2.311 | 1.704 | 3.351 |
| 65 | 0.342 | 0.302 | 0.307 | 2.21 | 1.791 | 3.044 |
| 70 | 0.331 | 0.295 | 0.294 | 2.252 | 1.76 | 3.012 |
| 75 | 0.31 | 0.301 | 0.299 | 2.246 | 1.842 | 3.07 |
| 80 | 0.302 | 0.303 | 0.305 | 2.305 | 1.853 | 3.095 |
| 85 | 0.302 | 0.302 | 0.302 | 2.36 | 1.823 | 3.063 |
| 90 | 0.297 | 0.311 | 0.31 | 2.292 | 1.828 | 3.087 |
| 95 | 0.3 | 0.311 | 0.311 | 2.34 | 1.864 | 3.122 |
| 100 | 0.305 | 0.312 | 0.314 | 2.494 | 2.012 | 3.259 |
| 105 | 0.307 | 0.243 | 0.243 | 2.552 | 1.936 | 3.179 |
| 110 | 0.29 | 0.234 | 0.233 | 2.544 | 1.921 | 3.164 |
| 115 | 0.26 | 0.228 | 0.228 | 2.504 | 1.871 | 3.107 |
| 120 | 0.244 | 0.222 | 0.221 | 2.513 | 1.852 | 3.336 |
| 124 | 0.239 | 0.224 | 0.22 | 2.478 | 1.851 | 3.383 |

Table E.11: Utilities for Read-Summarize-Only Read Strategy Learned via ADP

| Trial | Task Progress=1 | Task Progress=2 | Task Progress=3 | Task Progress=1 | Task Progress=2 | Task Progress=3 |
|-------|-----------------|-----------------|-----------------|-------------------|-------------------|-------------------|
| | No Local Reward | No Local Reward | No Local Reward | With Local Reward | With Local Reward | With Local Reward |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0.22 | 0 | 0 | 0.635 | 0 |
| 15 | 0 | 0.282 | 0 | 0 | 0.704 | 0 |
| 20 | 0.204 | 0.213 | 0.091 | 1.945 | 2.913 | 5.874 |
| 25 | 0.212 | 0.214 | 0.181 | 2.605 | 2.815 | 5.773 |
| 30 | 0.195 | 0.249 | 0.236 | 1.614 | 2.31 | 5.232 |
| 35 | 0.226 | 0.276 | 0.273 | 1.911 | 2.344 | 5.309 |
| 40 | 0.28 | 0.26 | 0.267 | 1.862 | 2.28 | 5.254 |
| 45 | 0.275 | 0.24 | 0.246 | 1.8 | 2.257 | 5.222 |
| 50 | 0.231 | 0.228 | 0.233 | 1.549 | 2.224 | 5.219 |
| 55 | 0.21 | 0.235 | 0.237 | 1.534 | 2.123 | 5.13 |
| 60 | 0.209 | 0.236 | 0.239 | 1.531 | 2.144 | 5.133 |
| 65 | 0.207 | 0.24 | 0.244 | 1.527 | 2.242 | 5.184 |
| 70 | 0.214 | 0.237 | 0.241 | 1.576 | 2.248 | 5.187 |
| 75 | 0.218 | 0.235 | 0.239 | 1.669 | 2.302 | 5.233 |
| 80 | 0.219 | 0.225 | 0.23 | 1.726 | 2.248 | 5.205 |
| 85 | 0.211 | 0.207 | 0.215 | 1.667 | 2.245 | 5.196 |
| 90 | 0.209 | 0.194 | 0.195 | 1.756 | 2.532 | 4.515 |
| 95 | 0.199 | 0.186 | 0.185 | 1.893 | 2.582 | 4.541 |
| 100 | 0.189 | 0.172 | 0.169 | 2.049 | 2.545 | 4.46 |
| 105 | 0.173 | 0.108 | 0.095 | 2.033 | 2.41 | 4.326 |
| 110 | 0.156 | 0.078 | 0.064 | 1.988 | 2.36 | 4.283 |
| 115 | 0.119 | 0.053 | 0.045 | 1.922 | 2.358 | 4.251 |
| 120 | 0.088 | 0.044 | 0.039 | 1.907 | 2.389 | 4.282 |
| 124 | 0.073 | 0.04 | 0.037 | 1.918 | 2.394 | 4.285 |

Table E.12: Utilities for Read-Choice-Prompt Read Strategy Learned via Q-Learning

| Trial | Task Progress=1 | Task Progress=2 | Task Progress=3 | Task Progress=1 | Task Progress=2 | Task Progress=3 |
|-------|-----------------|-----------------|-----------------|-------------------|-------------------|-------------------|
| | No Local Reward | No Local Reward | No Local Reward | With Local Reward | With Local Reward | With Local Reward |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0.001 | 0 | 0 | 3.408 | 0 | 0 |
| 10 | 0.432 | 0.432 | 0 | 3.354 | 2.614 | 0 |
| 15 | 0.365 | 0.315 | 0 | 3.167 | 2.464 | 0 |
| 20 | 0.315 | 0.306 | 0.214 | 3.428 | 2.641 | 3.656 |
| 25 | 0.305 | 0.258 | 0.26 | 3.183 | 2.359 | 3.495 |
| 30 | 0.289 | 0.239 | 0.24 | 3.099 | 2.367 | 3.325 |
| 35 | 0.242 | 0.223 | 0.223 | 3.183 | 2.269 | 3.246 |
| 40 | 0.22 | 0.224 | 0.224 | 3.176 | 2.266 | 3.242 |
| 45 | 0.22 | 0.224 | 0.221 | 3.152 | 2.36 | 3.323 |
| 50 | 0.22 | 0.143 | 0.141 | 3.163 | 2.372 | 3.245 |
| 55 | 0.168 | 0.092 | 0.1 | 3.082 | 2.251 | 3.198 |
| 60 | 0.135 | 0.078 | 0.086 | 2.979 | 2.197 | 3.183 |
| 65 | 0.097 | 0.073 | 0.083 | 2.823 | 2.174 | 2.77 |
| 70 | 0.082 | 0.077 | 0.086 | 2.792 | 2.326 | 2.864 |
| 75 | 0.074 | 0.086 | 0.097 | 2.915 | 2.334 | 2.851 |
| 80 | 0.073 | 0.115 | 0.128 | 2.975 | 2.291 | 2.818 |
| 85 | 0.086 | 0.147 | 0.16 | 2.886 | 2.276 | 2.818 |
| 90 | 0.108 | 0.161 | 0.174 | 2.894 | 2.232 | 2.775 |
| 95 | 0.132 | 0.176 | 0.187 | 2.891 | 2.177 | 2.735 |
| 100 | 0.152 | 0.193 | 0.204 | 2.817 | 2.112 | 2.674 |
| 105 | 0.176 | 0.184 | 0.192 | 2.747 | 2.082 | 2.644 |
| 110 | 0.179 | 0.187 | 0.195 | 2.735 | 2.113 | 2.675 |
| 115 | 0.178 | 0.192 | 0.201 | 2.718 | 2.136 | 2.713 |
| 120 | 0.182 | 0.203 | 0.212 | 2.75 | 2.196 | 2.755 |
| 124 | 0.184 | 0.209 | 0.217 | 2.767 | 2.2 | 2.759 |

Table E.13: Utilities for Read-First Read Strategy Learned via Q-Learning

| Trial | Task Progress=1 | Task Progress=2 | Task Progress=3 | Task Progress=1 | Task Progress=2 | Task Progress=3 |
|-------|-----------------|-----------------|-----------------|-------------------|-------------------|-------------------|
| | No Local Reward | No Local Reward | No Local Reward | With Local Reward | With Local Reward | With Local Reward |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0.56 | 0 | 0 | 2.135 | 0 | 0 |
| 15 | 0.377 | 0 | 0 | 1.145 | 0 | 0 |
| 20 | 0.159 | 0 | 0 | 1.105 | 0 | 0 |
| 25 | 0.239 | 0 | 0 | 1.115 | 0 | 0 |
| 30 | 0.242 | 0 | 0 | 2.193 | 0 | 0 |
| 35 | 0.271 | 0.341 | 0 | 2.196 | 1.548 | 0 |
| 40 | 0.336 | 0.335 | 0.253 | 2.114 | 1.744 | 3.267 |
| 45 | 0.34 | 0.334 | 0.322 | 2.308 | 1.896 | 3.531 |
| 50 | 0.319 | 0.358 | 0.355 | 2.317 | 1.84 | 3.488 |
| 55 | 0.348 | 0.357 | 0.363 | 2.337 | 1.813 | 3.45 |
| 60 | 0.355 | 0.327 | 0.334 | 2.316 | 1.708 | 3.353 |
| 65 | 0.344 | 0.308 | 0.318 | 2.207 | 1.791 | 3.046 |
| 70 | 0.333 | 0.295 | 0.289 | 2.25 | 1.763 | 3.012 |
| 75 | 0.314 | 0.299 | 0.297 | 2.245 | 1.834 | 3.066 |
| 80 | 0.302 | 0.303 | 0.305 | 2.312 | 1.855 | 3.097 |
| 85 | 0.302 | 0.302 | 0.302 | 2.362 | 1.823 | 3.063 |
| 90 | 0.3 | 0.31 | 0.309 | 2.289 | 1.821 | 3.078 |
| 95 | 0.299 | 0.311 | 0.31 | 2.34 | 1.864 | 3.122 |
| 100 | 0.305 | 0.268 | 0.314 | 2.494 | 2.012 | 3.259 |
| 105 | 0.308 | 0.247 | 0.247 | 2.551 | 1.935 | 3.178 |
| 110 | 0.294 | 0.233 | 0.228 | 2.543 | 1.922 | 3.166 |
| 115 | 0.263 | 0.229 | 0.229 | 2.506 | 1.857 | 3.109 |
| 120 | 0.245 | 0.222 | 0.222 | 2.511 | 1.856 | 3.339 |
| 124 | 0.239 | 0.224 | 0.22 | 2.478 | 1.851 | 3.383 |

Table E.14: Utilities for Read-Summarize-Only Read Strategy Learned via Q-Learning

Bibliography

- [Agre and Chapman, 1987] P.E. Agre and D. Chapman. Pengi: an implementation of a theory of activity. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, pages 268–272, Los Altos, CA, 1987. Morgan Kaufmann.
- [Allen, 1995] James F. Allen. *Natural language understanding*. Benjamin/Cummings, Redwood City, California, 1995.
- [Appelt, 1985] Douglas E. Appelt. Planning English referring expressions. *Artificial Intelligence*, 26(1):1–33, 1985.
- [Barto *et al.*, 1995] A.G. Barto, S.J. Bradtke, and S.P. Singh. Learning to act using real-time dynamic programming. *Artificial Intelligence*, 72(1-2):81–138, 1995.
- [Bellman, 1957] Richard E. Bellman. *Dynamic programming*. Princeton University Press, Princeton, NJ, 1957.
- [Brady and Berwick, 1982] M. Brady and R.C. Berwick. *Computational models of discourse*. MIT Press, Cambridge, MA, 1982.
- [Callaway and Lester, 1997] Charles B. Callaway and James C. Lester. Dynamically improving explanations: a revision-based approach to explanation generation. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence*, Nagoya, Japan, August 1997.
- [Choi and Yeung, 1995] Samuel P.M. Choi and Dit-Yan Yeung. Predictive Q-routing: a memory-based reinforcement learning approach to adaptive traffic control. In *Advances in Neural Information Processing Systems 7*, 1995.

- [Cohen *et al.*, 1990] P. Cohen, J. Morgan, and M. Pollack. *Intentions in communication*. MIT Press, Cambridge, MA, 1990.
- [Cohen, 1995] Paul R. Cohen. *Empirical methods for artificial intelligence*. MIT Press, Cambridge, MA, 1995.
- [Crabbe *et al.*, 1995] Fredrick Crabbe, Paul Martin, Stuart Adams, Eric Baatz, and Nicole Yankelovich. SpeechActs: a spoken language framework. *IEEE Computer Special Issue on Natural Language Processing*, 1995.
- [Danieli and Gerbino, 1995] M. Danieli and E. Gerbino. Metrics for evaluating dialogue strategies in a spoken language system. In Marilyn Walker and Johanna Moore, editors, *Proceedings of the 1995 AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, pages 34–39, 1995.
- [Dayan, 1992] Peter Dayan. The convergence of TD λ for general λ . *Machine Learning*, 8(3–4):341–362, 1992.
- [Edwards and Penney, 1990] C.H. Edwards, Jr. and David E. Penney. *Calculus and analytic geometry*. Prentice Hall, Englewood Cliffs, NJ, 1990.
- [Elhadad, 1992] Michael Elhadad. *Using agumentation to control lexical choice: a functional unification-based approach*. PhD thesis, Columbia University, 1992.
- [Gale *et al.*, 1992] William Gale, Ken W. Church, and David Yarowsky. Estimating upper and lower bounds on the performance of word-sense disambiguation programs. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, pages 249–256, Newark, Delaware, 1992.
- [Grosz and Sidner, 1986] Barbara J. Grosz and Candace L. Sidner. Attentions, intentions, and the structure of discourse. *Computational Linguistics*, 12:175–204, 1986.
- [Grosz *et al.*, 1986] Barbara J. Grosz, Karen Sparck Jones, and Bonnie L. Webber. *Readings in natural language processing*. Morgan Kaufmann, San Mateo, California, 1986.

- [Harvey and Carberry, 1998] Terrence Harvey and Sandra Carberry. Integrating text plans for conciseness and coherence. In *Proceedings of COLING-ACL '98*, 1998.
- [Hayes-Roth *et al.*, 1993] Barbara Hayes-Roth, Karl Pfelger, Philippe Morignot, and Philippe Lalanda. Plans and behavior in intelligent agents. Technical Report KSL-93-43, Stanford University Knowledge Systems Laboratory, 1993.
- [Hovy, 1993] Eduard H. Hovy. Automated discourse generation using discourse structure relations. *Artificial Intelligence*, 63:341–385, 1993.
- [Isard and Carletta, 1995] Amy Isard and Jean C. Carletta. Replicability of transaction and action coding in the map task corpus. In Marilyn Walker and Johanna Moore, editors, *Proceedings of the 1995 AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, pages 60–67, 1995.
- [Jaakkola *et al.*, 1994] Thommi Jaakkola, Satinder P. Singh, and Michael I. Jordan. Reinforcement learning algorithm for partially observable Markov decision problems. In *Advances in Neural Information Processing Systems 7*, 1994.
- [Kaelbling *et al.*, 1996] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: a survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.
- [Kaelbling, 1996] Leslie Pack Kaelbling, editor. *Recent advances in reinforcement learning*. Kluwer Academic Publishers, Boston, 1996.
- [Kamm *et al.*, 1995] Candace A. Kamm, C.R. Shamieh, and S. Singhal. Speech recognition issues for directory assistance applications. *Speech Communication*, 17:303–311, 1995.
- [Kamm *et al.*, 1997] Candace Kamm, Shrikanth Narayanan, Dawn Dutton, and Russell Ritenour. Evaluating spoken dialog systems for telecommunication services. In *Proceedings of the European Conference on Speech Communication and Technology, EUROSPEECH97*, 1997.

- [Kamm, 1994] Candace Kamm. User interfaces for voice applications. In David Roe and Jay Wilpon, editors, *Voice communication between humans and machines*. National Academy of Sciences Press, Washington, D.C., 1994.
- [Keeney and Raiffa, 1976] Ralph Keeney and Howard Raiffa. *Decisions with multiple objectives: preferences and value tradeoffs*. John Wiley and Sons, 1976.
- [Koenig and Simmons, 1996] Sven Koenig and Reid G. Simmons. The effect of representation and knowledge on goal-directed exploration with reinforcement-learning algorithms. *Machine Learning*, 22:227–250, 1996.
- [Lenzmann, 1996] Britta Lenzmann. Interface agents for interfacing with virtual environments. In *CHI '96 Doctorial Consortium*, 1996.
- [Levin and Pieraccini, 1997] Esther Levin and Roberto Pieraccini. A stochastic model for learning dialogue strategies. In *Proceedings of the European Conference on Speech Communication and Technology, EUROSPEECH97*, 1997.
- [Lin, 1992] Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, 8:293–321, 1992.
- [Mann and Thompson, 1987] W.C. Mann and S.A. Thompson. Rhetorical structure theory: description and construction of text structures. In Gerard Kempen, editor, *Natural Language Generation*, pages 83–96. Martinus Nijhoff, 1987.
- [Marcus *et al.*, 1996] S.M. Marcus, D.W. Brown, R.G. Goldberg, M.S. Schoeffler, W.R. Wetzell, and R.R. Rosinski. Prompt constrained natural language - evolving the next generation of telephony services. In *Proceedings of the International Conference on Spoken Language Processing ICSLP*, pages 857–860, 1996.
- [Martin and Kehler, 1994] Paul Martin and Andrew Kehler. SpeechActs: a testbed for continuous speech applications. In *AAAI '94 Workshop on the Integration of Natural Language and Speech Processing*, Seattle, 1994.

- [Marx, 1995] M. Marx. Toward effective conversational messaging. Master's thesis, MIT Media Laboratory, 1995.
- [Mataric, 1994] Maja J. Mataric. Reward functions for accelerated learning. In W. W. Cohen and H. Hirsh, editors, *Proceedings of the Eleventh International Conference on Machine Learning*. Morgan Kaufmann, 1994.
- [McKeown, 1985] Kathleen R. McKeown. Discourse strategies for generating natural-language text. *Artificial Intelligence*, 27(1):1–42, 1985.
- [Moore and Paris, 1993] Johanna Moore and Cécile Paris. Planning text for advisory dialogues: capturing intentional and rhetorical information. *Computational Linguistics*, 19(4):651–695, 1993.
- [Morignot and Hayes-Roth, 1995] Philippe Morignot and Barbara Hayes-Roth. Adaptable motivational profiles for autonomous agents. Report KSL-95-01, Stanford University Department of Computer Science Knowledge Systems Lab, 1995.
- [Neuneier, 1995] Ralph Neuneier. Optimal asset allocation using adaptive dynamic programming. In *Advances in Neural Information Processing Systems 7*, 1995.
- [Potjer *et al.*, 1996] J. Potjer, A. Russel, L. Boves, and E. den Os. Subjective and objective evaluation of two types of dialogues in a call assistance service. In *1996 IEEE 3rd Workshop: Interactive Voice Technology for Telecommunications Applications, IVTTA*, pages 89–92. IEEE, 1996.
- [Power, 1974] Richard Power. *A computer model of conversation*. PhD thesis, University of Edinburgh, 1974.
- [Power, 1979] Richard Power. The organisation of purposeful dialogues. *Linguistics*, 17:107–152, 1979.
- [Robin, 1995] Jacques Robin. *Revision-based generation of natural language summaries providing historical background: corpus-based analysis, design, implementation, and evaluation*. PhD thesis, Columbia University, 1995.

- [Russell and Norvig, 1995] Stuart Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Prentice Hall, Englewood Cliffs, NJ, 1995.
- [Singh *et al.*, 1994] Satinder P. Singh, Tommi Jaakkola, and Michael I. Jordan. Learning without state-estimation in partially observable Markovian decision processes. In *Proceedings of the 11th International Conference on Machine Learning*, pages 284–292, 1994.
- [Smith and Hipp, 1994] R.W. Smith and D.R. Hipp. *Spoken natural language dialog systems: a practical approach*. Oxford University Press, 1994.
- [Sutton, 1990] Richard S. Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Proceedings of the 7th International Conference on Machine Learning*, pages 216–224. Morgan Kaufmann, 1990.
- [Sutton, 1991] Richard S. Sutton. Planning by incremental dynamic programming. In *Proceedings of the 9th Conference on Machine Learning*, pages 353–357. Morgan Kaufmann, 1991.
- [Sutton, 1992] Richard S. Sutton, editor. *Reinforcement learning*. Kluwer Academic Publishers, Boston, 1992.
- [Tesauro, 1992] Gerald Tesauro. Practical issues in temporal difference learning. *Machine Learning*, 8:257–277, 1992.
- [Walker and Whittaker, 1990] Marilyn Walker and S. Whittaker. Mixed initiative in dialogue: an investigation into discourse segmentation. In *Proceedings of the 28th Annual Meeting of the Association of Computational Linguistics*, pages 70–79, 1990.
- [Walker *et al.*, 1997a] Marilyn Walker, Don Hindle, Jeanne Fromer, Giuseppe Di Fabrizio, and Craig Mestel. Evaluating competing agent strategies for a voice email agent. In *Proceedings of the European Conference on Speech Communication and Technology, EUROSPEECH97*, September 1997.

- [Walker *et al.*, 1997b] Marilyn Walker, Diane Litman, Candace Kamm, and Alicia Abella. PARADISE: a framework for evaluating spoken dialogue systems. In *Proceedings of the 35th Annual Meeting of the Association of Computational Linguistics, ACL/EACL 97*, Madrid, Spain, July 1997. MIT Press.
- [Walker *et al.*, 1998a] Marilyn Walker, Jeanne Fromer, Giuseppe Di Fabrizio, Craig Mestel, and Don Hindle. What can I say?: evaluating a spoken language interface to email. In *Proceedings of CHI '98*, April 1998.
- [Walker *et al.*, 1998b] Marilyn Walker, Jeanne Fromer, and Shrikanth Narayanan. Learning optimal dialogue strategies: a case study of a spoken dialogue agent for email. In *Proceedings of COLING-ACL '98*, August 1998.
- [Watkins, 1989] Christopher J. Watkins. *Models of delayed reinforcement*. PhD thesis, Cambridge University, 1989.
- [Whitehead and Ballard, 1991] Steven D. Whitehead and Dana H. Ballard. Learning to perceive and act by trial and error. *Machine Learning*, 7:45–83, 1991.
- [Yanco and Stein, 1993] Holly Yanco and Lynn Andrea Stein. An adaptive communication protocol for cooperating mobile robots. In J.A. Meyer, H.L. Roitblat, and S.W. Wilson, editors, *From Animals to Animats: Proceedings of the 2nd International Conference on the Simulation of Adaptive Behavior*, pages 478–485. MIT Press/Bradford Books, 1993.
- [Yankelovich, 1994] Nicole Yankelovich. Talking vs. taking: speech access to remote computers. In *CHI '94 Conference Companion, ACM Conference on Human Factors in Computing Systems*, pages 275–276, Boston, 1994.
- [Zukerman and McConachy, 1995] Ingrid Zukerman and Richard McConachy. Generating discourse across several user models: maximizing belief while avoiding boredom and overload. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1251–1257, 1995.