To appear in the proc. of the International Conference on Computational Linguistics (COLING), 2012.

Hierarchical Dialogue Policy Learning Using Flexible State Transitions and Linear Function Approximation

Heriberto Cuayáhuitl[1], *Ivana Kruijff-Korbayová*[2], *Nina Dethlefs*[1]
[1]Heriot-Watt University, Edinburgh, Scotland, United Kingdom
[2]German Research Center for Artificial Intelligence, Saarbrücken, Germany
h.cuayahuitl@hw.ac.uk, ivana.kruijff@dfki.de, n.s.dethlefs@hw.ac.uk

Abstract

Conversational agents that use reinforcement learning for policy optimization in large domains often face the problem of limited scalability. This problem can be addressed either by using function approximation techniques that estimate an approximate true value function, or by using a hierarchical decomposition of a learning task into subtasks. In this paper, we present a novel approach for dialogue policy optimization that combines the benefits of hierarchical control with function approximation. The approach incorporates two concepts to allow flexible switching between sub-dialogues, extending current hierarchical reinforcement learning methods. First, hierarchical tree-based state representations initially represent a compact portion of the possible state space and are then dynamically extended in real time. Second, we allow state transitions across sub-dialogues to allow non-strict hierarchical control. Our approach is integrated, and tested with real users, in a robot dialogue system that learns to play Quiz games.

Keywords: spoken dialogue systems, reinforcement learning, hierarchical control, function approximation, user simulation, human-robot interaction, flexible interaction.

1 Introduction

The past decade has experienced important progress in spoken dialogue systems that learn their conversational behaviour. The Reinforcement Learning (RL) framework in particular has been an attractive alternative to hand-coded policies for the design of sophisticated and adaptive dialogue agents. An RL agent learns its behaviour from interaction with an environment, where situations are mapped to actions by maximizing a long-term reward signal (Sutton and Barto, 1998). While RL-based dialogue systems are promising, they still need to overcome several limitations to reach practical and wide-spread application. One of these limitations is the *curse of dimensionality*, the problem that the state space grows exponentially according to the variables taken into account. Another limitation is that attempts to address the first problem often involve rule-based reductions of the state space (Litman et al., 2000; Singh et al., 2002; Heeman, 2007; Williams, 2008; Cuayáhuitl et al., 2010b; Dethlefs et al., 2011) which can lead to reduced flexibility of system behaviour in terms of letting the user say and/or do anything at any time during the dialogue. Finally, even when function approximation techniques have been used to scale up in small-scale and single-task systems (Henderson et al., 2008; Li et al., 2009; Pietquin, 2011; Jurcícek et al., 2011), their application to more complex dialogue contexts has yet to be demonstrated.

Our motivation for increased *dialogue flexibility* in this paper is the assumption that users at times deviate from the system's expected user behaviour. In reduced state spaces this may lead to unseen dialogue states in which the system cannot react properly to the new situation, as is exemplified

by dialogue 3S in Figure 1. So whilst a full state space represents maximal flexibility (according to the state variables considered), but is often not scalable, reducing the state space for increased scalability simultaneously faces a risk of reducing dialogue flexibility.

This paper presents a novel approach for dialogue policy optimization that increases both dialogue flexibility and scalability. It is couched within a Hierarchical Reinforcement Learning (HRL) framework, a principled and scalable model for optimizing sub-behaviours (Barto and Mahadevan, 2003). We extend an existing HRL-algorithm with the following features: (1) dynamic tree-based state representations that can grow during a dialogue, *during the course of the dialogue*, according to the state variables used in the interaction; and (2) rather than imposing strict hierarchical dialogue control, we allow users to navigate more flexibly across the available sub-dialogues. A further extension is the representation of the dialogue policy using function approximation in order to generalize decision-making even for unseen situations.

2 Proposed Learning Approach

This section proposes two extensions of the hierarchical RL framework presented in (Cuayáhuitl and Dethlefs, 2011b): (1) dynamic tree-based state representations that grow over the course of a dialogue; and (2) state transitions across sub-dialogues for flexible dialogue control. In addition, we make use of function approximation to support decision-making for unseen situations.

2.1 Dynamic Tree-based State Representations

We treat each multi-step action as a separate SMDP as described in (Cuayáhuitl et al., 2007; Cuayáhuitl, 2009). To allow compact state representations to grow over the course of a dialogue, we redefine such SMDP models as tuples $M_{\beta}^{(i,j)} = <S_{\beta}^{(i,j)}, A_{\beta}^{(i,j)}, T_{\beta}^{(i,j)}, R_{\beta}^{(i,j)}, L_{\beta}^{(i,j)}>$, where $S_{\beta}^{(i,j)}$ is a finite set of dynamic tree-based states that grow from $S_{\beta}^{(i,j)}$ to $S_{\beta'}^{(i,j)}$ after new states are encountered.[1] This principle is illustrated in Figure 1. Since the state space changes continuously, the actions per state $A_{\beta}^{(i,j)}$, the corresponding state transitions $T_{\beta}^{(i,j)}$, reward functions $R_{\beta}^{(i,j)}$, and context-free grammar $L_{\beta}^{(i,j)}$, which defines the tree-based state representations, are affected by these changes. When an unknown (or unseen) state is observed, the current subtask $M_{\beta}^{(i,j)}$ is updated to support that new state, where β refers to the number of times the subtask $M^{(i,j)}$ has been updated. An update consists of the following steps:

1. extend the state space and corresponding grammar $L_{\beta}^{(i,j)}$ with the new states;
2. assign a set of actions to the new states, which satisfy pre-specified constraints (if any); and
3. extend the state transition function to support transitions to the new states.

Although our approach could also continuously extend the reward function to cover the new observed situations, we leave this point as future work. The solution to our redefined model consists in approximating or relearning an optimal policy for each agent in the hierarchy continuously often, i.e. whenever the state space has grown, according to

$$\pi_{\beta}^{*(i,j)}(s \in S_{\beta}^{(i,j)}) = \arg\max_{a \in A_{\beta}^{(i,j)}} Q_{\beta}^{*i,j}(s,a). \quad (1)$$

[1] No threshold is imposed on the growth of state spaces. In the worst case, it would represent all possible combinations of state variable values. Nevertheless, it is reasonable to assume that only a subset of it would be observed in real interactions.

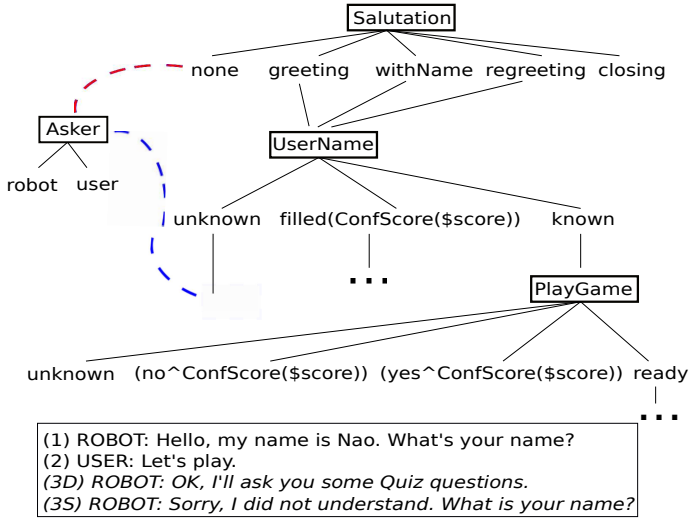

Figure 1: Fragment of our dialogue state space, where rectangles represent state variables that expand their domain values. A tree-branch (with expanded variables, here denoted as '$variable') is a unique dialogue state. The dashed lines illustrate a dynamically growing tree according to the example dialogue. Here, 3D represents a dialogue with dynamically growing state representations and 3S represents a dialogue with static state representations.

Note that the initial state space $S_{\beta=0}^{(i,j)}$ is compact so that unseen states can be observed during the agent-environment interaction from a knowledge-rich state k that maintains the dialogue history. k can also be seen as a knowledge base and is used to initialize states of each subtask, so that we can extend the state space for unseen states within a given set of state variables. We can approximate the optimal policies using linear function approximation. The policies are represented by a set of weighted linear functions expressed as

$$Q_{\theta}^{(i,j)}(s,a) = \theta_0^{(i,j)} + \theta_1^{(i,j)} f_1(s,a) + ... + \theta_n^{(i,j)} f_n(s,a) \quad (2)$$

with a set of state features $f_i \in F$ [2] and parameters $\theta^{(i,j)} = \{\theta_1^{(i,j)}, ...\theta_n^{(i,j)}\}$ for each agent in the hierarchy. A reinforcement learning agent can learn values for the parameters θ, where the utility functions $Q_{\theta}^{(i,j)}$ approximate to the true utility functions.

2.2 Flexible Navigation across Sub-Dialogues

To allow flexible navigation across sub-dialogues, we extend the previous models with tuples $M_{\beta}^{(i,j)} = <S_{\beta}^{(i,j)}, A_{\beta}^{(i,j)}, T_{\beta}^{(i,j)}, R_{\beta}^{(i,j)}, L_{\beta}^{(i,j)}, G_{\beta}^{(i,j)}>$, where the new element $G_{\beta}^{(i,j)} = P(m'|m,s,a)$ is a stochastic model transition function that specifies the next model or subtask $m' \in \mu$ given model $m \in \mu$, state s and action a. Here, μ refers to the set of all models. The new element $G_{\beta}^{(i,j)}$ is the mechanism to specify the currently active subtask for each state-action (see Figure 2). This is a relevant feature in dialogue agents in order to allow users to act freely at anytime and across

[2] A dialogue state is represented as a vector of binary features derived from the tree-based representations, where every possible variable-value pair in the tree is represented with 1 if present and 0 if absent.

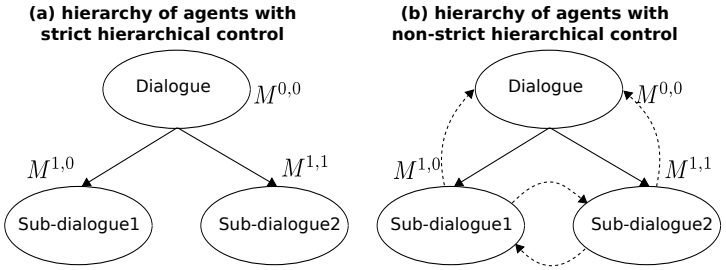

Figure 2: Hierarchies of agents with local and global state transitions.Whilst the straight arrows connecting models mean invoking a model and returning control after terminating its execution, the dashed arrows connecting models mean interrupting the execution of the current model and transition to another one to continue the interaction.

sub-dialogues, while dialogue agents should be able to react with appropriate actions. The goal of these extended SMDPs is to learn a similar mapping as in the previous section, expressed as

$$\pi_{\theta}^{*(i,j)}(s \in S_{\beta}^{(i,j)}) = \arg\max_{a \in A_{\beta}^{(i,j)}} Q_{\theta}^{*(i,j)}(s,a), \tag{3}$$

where we extend the HSMQ-learning algorithm (Dietterich, 2000; Cuayáhuitl et al., 2010b; Cuayáhuitl and Dethlefs, 2011a) to induce the action-value functions Q_{θ}, one per SMDP model.

3 Experimental Setting

To test if our approach can generate more flexible interactions than a baseline, we use a robot dialogue system in the Quiz domain, where the robot[3] can ask the user questions, or vice-versa, the user can ask the robot questions. In addition, to allow flexibility, the user or robot can switch roles or stop playing at any point during the interaction (e.g. from any dialogue state).

3.1 Characterization of the Learning Agents

We use a compact hierarchy of dialogue agents with one parent and two children agents ('robot asks' and 'user asks'), which is shown in Figure 2. (Cuayáhuitl and Dethlefs, 2012)-Table 1 shows the set of state variables for our system, each one modelled as a discrete probability distribution with updated parameters at runtime. Dialogue and game features are included to inform the agent of situations in the interaction. Our action set consists of 80 meaningful combinations of speech act types[4] and associated parameters[5]. We constrained the actions per state based on the grammars $L_{\beta}^{(i,j)}$, i.e. only a subset of actions was allowed per dialogue state (constraints omitted due to space). While our HRL agent with tree-based states grows up to 10^4 state-actions, a static, propositional representation (enumerating all variables and values) has 10^{12} state-action pairs. This makes the tree-based representation attractive for complex, large-scale systems.

[3]Our dialogue system has been fully implemented and tested using wizarded and speech-based user responses with the actual Nao humanoid robot. An evaluation with real users will be reported in a forthcoming paper.

[4]Set of speech act types: Salutation, Request, Apology, Confirm, Accept, SwitchRole, Acknowledgement, Provide, Stop, Feedback. Express, Classify, Retrieve, Provide.

[5]Greet, Closing, Name, PlayGame, Asker, KeepPlaying, GameFun, StopPlaying, Play, NoPlay, Fun, NoFun, GameInstructions, StartGame, Question, Answers, CorrectAnswer, IncorrectAnswer, GamePerformance, Answer, Success, Failure, GlobalGameScore, ContinuePlaying.

The **reward function** addressed efficient and effective interactions by encouraging to play and get the right answers as much as possible. It is defined by the following rewards for choosing action a in state s: $+10$ for reaching a terminal state or answering a question correctly, -10 for remaining in the same state (i.e. $s_{t+1} = s_t$ or $s_{t+1} = s_{t-1}$), and 0 otherwise. The **user simulation** used a set of user dialogue acts as responses to the system dialogue acts (footnotes 4-5). The user dialogue acts were estimated using conditional distributions $P(a^{usr}|a^{sys})$ with Witten-Bell discounting from 21 wizarded dialogues (900 user turns). The attribute-values were distorted based on an equally distributed speech recognition error rate of 20%. The confidence scores of attribute values were generated from beta probability distributions with parameters (α=2, β=6) for bad recognition and (α=6, β=2) for good recognition.

3.2 The Robot Dialogue System

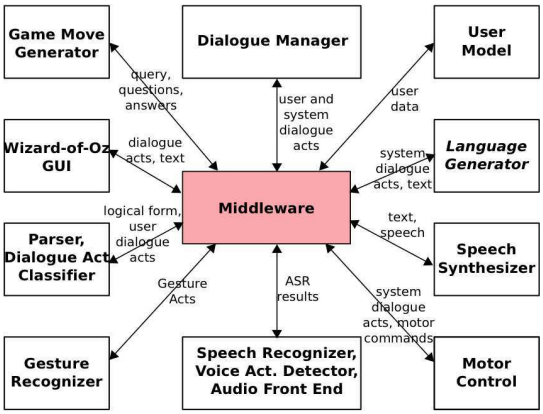

Figure 3: High-level architecture of the integrated system.

Our experiments were carried out using a dialogue system (Cuayáhuitl and Kruijff-Korbayová, 2012) running on the Nao robot [6]. The system integrates components for speech and gesture capture and interpretation, activity and interaction management, user modeling, speech and gesture production and robot motor control (see Figure 3). We use components developed by ourselves as well as off-the-shelf technologies such as Google speech recognition, OpenCV for gesture recognition, Acapela for speech synthesis, OpenCCG for language parsing and generation, and Weka and JavaBayes for maintaining a probabilistic personalized user profile. To bring all components together within a concurrent execution approach, we use the Urbi middleware (Baillie, 2005). More details on the system implementation are described in (Kruijff-Korbayová et al., 2012b,a).

During interactions, the user provided responses through a mobile device, which were processed by a Bayesian dialogue act classifier estimated from our Wizard-of-Oz data. Based on this, the next system action is selected by the Dialogue Manager component (as described in Sections 2 and 3). The dialogue act corresponding to the selected next system action is verbalized automatically by the Natural Language Generation component which produces text for the speech synthesizer. Nonverbal behavior planning and motor control are automatic communicative gestures assigned to specific types of dialogue acts (e.g., greetings, requests), and static key poses displaying emotions such as anger, sadness, fear, happiness, excitement and pride (Beck et al., 2010).

[6] www.aldebaran-robotics.com

Figure 4: Participants talking to the NAO robot through a smartphone while playing the Quiz game. The pieces of paper on the table are used by the users to ask the robot questions.

To summarize, the following features describe the system used in our experiments: (a) automatic speech and dialogue act recognition; (b) automatic system action selection; (c) user barge-ins in the form of interruptions of the robot's speech by an early user response; (d) automatically produced verbal output in English with many variations and expressive speech synthesis distinguishing sad, happy and neutral state; (e) automatically produced head and body poses and gestures; and (f) persistent user-specific interaction profile. This robot dialogue system has been evaluated with simulated and real users, see Figure 4. A comprehensive evaluation description and results analysis will be reported in a forthcoming paper.

4 Conclusion and Future Work

We have described a novel approach for optimizing dialogue systems by extending an existing HRL framework to support dynamic state spaces, non-strict hierarchical control, and linear function approximation. We evaluated our approach by incorporating it into a robot dialogue system that learns to play Quiz games. Our experimental results, based on simulation and experiments with human users (reported elsewhere), show that our approach is promising. It can yield more flexible interactions than a policy that uses strict control and is preferred by human users. We expect that our approach will represent an important step forward in the development of more sophisticated dialogue systems that combine the benefits of trainable, scalable and flexible interaction.

As future work, we suggest the following directions in order to equip spoken or multimodal dialogue systems with more flexible and adaptive conversational interaction: (1) to learn model state transition functions automatically so that the system can suggest how to navigate in the hierarchy of sub-dialogues; (2) to optimize dialogue control combining verbal and non-verbal behaviours (Cuayáhuitl and Dethlefs, 2012; Dethlefs et al., 2012b); (3) to optimize dialogue control jointly with natural language generation (Dethlefs and Cuayáhuitl, 2011b,a; Lemon, 2011); (4) to extend our approach with large hierarchies and partially observable SMDPs; (5) an application to situated dialogue systems (Cuayáhuitl et al., 2010a; Cuayáhuitl and Dethlefs, 2011a,b; Janarthanam et al., 2012); (6) an application to complex turn-taking phenomena in systems with multiple modalities for more natural and effective interactions (Chao and Thomaz, 2012); (7) an application to incremental dialogue processing (Schlangen and Skantze, 2009) using reinforcement learning (Dethlefs et al., 2012a), and (8) to induce the reward function during the course of the interaction for providing online adaptation.

5 Acknowledgments

Funding by the EU-FP7 projects ALIZ-E (ICT-248116, `www.aliz-e.org`) and Spacebook (270019, `http://www.spacebook-project.eu`) is gratefully acknowledged.

References

Baillie, J. (2005). Urbi: Towards a universal robotic low-level programming language. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3219–3224. IEEE.

Barto, A. and Mahadevan, S. (2003). Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems: Theory and Applications*, 13(1-2):41–77.

Beck, A., Cañamero, L., and Bard, K. (2010). Towards an affect space for robots to display emotional body language. In *Ro-Man 2010*, pages 464–469, Viareggio, Italy.

Chao, C. and Thomaz, A. L. (2012). Timing in multimodal reciprocal interactions: control and analysis using timed petri nets. *Journal of Human-Robot Interaction*, 1(1):4–25.

Cuayáhuitl, H. (2009). *Hierarchical Reinforcement Learning for Spoken Dialogue Systems*. PhD thesis, School of Informatics, University of Edinburgh.

Cuayáhuitl, H. and Dethlefs, N. (2011a). Spatially-aware dialogue control using hierarchical reinforcement learning. *ACM Transactions on Speech and Language Processing*, 7(3):5:1–5:26.

Cuayáhuitl, H. and Dethlefs, N. (2012). Hierarchical multiagent reinforcement learning for coordinating verbal and non-verbal actions in robots. In *ECAI Workshop on Machine Learning for Interactive Systems (MLIS)*, pages 27–29, Montpellier, France.

Cuayáhuitl, H., Dethlefs, N., Frommberger, L., Richter, K.-F., and Bateman, J. (2010a). Generating adaptive route instructions using hierarchical reinforcement learning. In *Proc. of the International Conference on Spatial Cognition (Spatial Cognition VII)*, Portland, OR, USA.

Cuayáhuitl, H. and Dethlefs, N. a. (2011b). Optimizing situated dialogue management in unknown environments. In *INTERSPEECH*, pages 1009–1012, Florence, Italy.

Cuayáhuitl, H. and Kruijff-Korbayová, I. (2012). An interactive humanoid robot exhibiting flexible sub-dialogues. In *HLT-NAACL*, pages 17–20, Montreal, Canada.

Cuayáhuitl, H., Renals, S., Lemon, O., and Shimodaira, H. (2007). Hierarchical dialogue optimization using semi-Markov decision processes. In *INTERSPEECH*, pages 2693–2696, Antwerp, Belgium.

Cuayáhuitl, H., Renals, S., Lemon, O., and Shimodaira, H. (2010b). Evaluation of a hierarchical reinforcement learning spoken dialogue system. *Computer Speech and Language*, 24(2):395–429.

Dethlefs, N. and Cuayáhuitl, H. (2011a). Combining hierarchical reinforcement learning and Bayesian networks for natural language generation in situated dialogue. In *ENLG*, Nancy, France.

Dethlefs, N. and Cuayáhuitl, H. (2011b). Hierarchical reinforcement learning and hidden Markov models for task-oriented natural language generation. In *ACL-HLT*, pages 654–659, Portland, OR, USA.

Dethlefs, N., Cuayáhuitl, H., and Viethen, J. (2011). Optimising Natural Language Generation Decision Making for Situated Dialogue. In *SIGdial*, Portland, Oregon, USA.

Dethlefs, N., Hastie, H., Rieser, V., and Lemon, O. (2012a). Optimising Incremental Dialogue Decisions Using Information Density for Interactive Systems. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-CoNLL)*, Jeju, South Korea.

Dethlefs, N., Rieser, V., Hastie, H., and Lemon, O. (2012b). Towards optimising modality allocation for multimodal output generation in incremental dialogue. In *ECAI Workshop on Machine Learning for Interactive Systems (MLIS)*, pages 31–36, Montpellier, France.

Dietterich, T. (2000). An overview of MAXQ hierarchical reinforcement learning. In *Symposium on Abstraction, Reformulation, and Approximation (SARA)*, pages 26–44.

Heeman, P. (2007). Combining reinforcement learning with information-state update rules. In *Human Language Technology Conference (HLT)*, pages 268–275, Rochester, NY, USA.

Henderson, J., Lemon, O., and Georgila, K. (2008). Hybrid reinforcement/supervised learning of dialogue policies from fixed data sets. *Computational Linguistics*, 34(4):487–511.

Janarthanam, S., Lemon, O., Liu, X., Bartie, P., Mackaness, W., Dalmas, T., and Goetze, J. (2012). Integrating location, visibility, and Question-Answering in a spoken dialogue system for pedestrian city exploration. In *SEMDIAL*, pages 134–136, Paris, France.

Jurcícek, F., Thompson, B., and Young, S. (2011). Natural actor and belief critic: Reinforcement algorithm for learning parameters of dialogue systems modelled as POMDPs. *ACM Transactions on Speech and Language Processing*, 7(3):6.

Kruijff-Korbayová, I., Cuayáhuitl, H., Kiefer, B., Schröder, M., Cosi, P., Paci, G., Sommavilla, G., Tesser, F., Sahli, H., Athanasopoulos, G., Wang, W., Enescu, V., and Verhelst, W. (2012a). Spoken language processing in a conversational system for child-robot interaction. In *Workshop on Child-Computer Interaction (WOCCI)*, Portland, OR, USA.

Kruijff-Korbayová, I., Cuayáhuitl, H., Kiefer, B., Schröder, M., Csi, P., Paci, G., Sommavilla, G., Tesser, F., Sahli, H., Athanasopoulos, G., Wang, W., Enescu, V., and Verhelst, W. (2012b). A conversational system for multi-session child-robot interaction with several games. In *German Conference on Artificial Intelligence (KI), Saarbruecken, Germany*.

Lemon, O. (2011). Learning what to say and how to say it: Joint optimization of spoken dialogue management and natural language generation. *Computer Speech and Language*.

Li, L., Williams, J., and Balakrishnan, S. (2009). Reinforcement learning for dialog management using least-squares policy iteration and fast feature selection. In *INTERSPEECH*, Brighton, UK.

Litman, D., Kearns, M., Singh, S., and Walker, M. (2000). Automatic optimization of dialogue management. In *COLING*, pages 502–508, Saarbrucken, Germany.

Pietquin, O. (2011). Batch reinforcement learning for spoken dialogue systems with sparse value function approximation. In *NIPS Workshop on Learning and Planning from Batch Time Series Data*, Vancouver, Canada.

Schlangen, D. and Skantze, G. (2009). A General, Abstract Model of Incremental Dialogue Processing. In *EACL*, Athens, Greece.

Singh, S., Litman, D., Kearns, M., and Walker, M. (2002). Optimizing dialogue management with reinforcement learning: Experiments with the NJFun system. *JAIR*, 16:105–133.

Sutton, R. and Barto, A. (1998). *Reinforcement Learning: An Introduction*. MIT Press.

Williams, J. (2008). The best of both worlds: Unifying conventional dialog systems and POMDPs. In *INTERSPEECH*, Brisbane, Australia.