Learning to Compose Effective Strategies from a Library of Dialogue Components

Martijn Spitters† **Marco De Boni**‡ **Jakub Zavrel**† **Remko Bonnema**†

† Textkernel BV, Nieuwendammerkade 28/a17, 1022 AB Amsterdam, NL
{spitters,zavrel,bonnema}@textkernel.nl
‡ Unilever Corporate Research, Colworth House, Sharnbrook, Bedford, UK MK44 1LQ
marco.de-boni@unilever.com

Abstract

This paper describes a method for automatically learning effective dialogue strategies, generated from a library of dialogue content, using reinforcement learning from user feedback. This library includes greetings, social dialogue, chit-chat, jokes and relationship building, as well as the more usual clarification and verification components of dialogue. We tested the method through a motivational dialogue system that encourages take-up of exercise and show that it can be used to construct good dialogue strategies with little effort.

1 Introduction

Interactions between humans and machines have become quite common in our daily life. Many services that used to be performed by humans have been automated by natural language dialogue systems, including information seeking functions, as in timetable or banking applications, but also more complex areas such as tutoring, health coaching and sales where communication is much richer, embedding the provision and gathering of information in e.g. social dialogue. In the latter category of dialogue systems, a high level of naturalness of interaction and the occurrence of longer periods of satisfactory engagement with the system are a prerequisite for task completion and user satisfaction.

Typically, such systems are based on a dialogue strategy that is manually designed by an expert based on knowledge of the system and the domain, and on continuous experimentation with test users. In this process, the expert has to make many design choices which influence task completion and user satisfaction in a manner which is hard to assess, because the effectiveness of a strategy depends on many different factors, such as classification/ASR performance, the dialogue domain and task, and, perhaps most importantly, personality characteristics and knowledge of the user.

We believe that the key to maximum dialogue effectiveness is to listen to the user. This paper describes the development of an adaptive dialogue system that uses the feedback of users to automatically improve its strategy. The system starts with a library of generic and task-/domain-specific dialogue components, including social dialogue, chit-chat, entertaining parts, profiling questions, and informative and diagnostic parts. Given this variety of possible dialogue actions, the system can follow many different strategies within the dialogue state space. We conducted training sessions in which users interacted with a version of the system which randomly generates a possible dialogue strategy for each interaction (restricted by global dialogue constraints). After each interaction, the users were asked to reward different aspects of the conversation. We applied reinforcement learning to use this feedback to compute the optimal dialogue policy.

The following section provides a brief overview of previous research related to this area and how our work differs from these studies. We then proceed with a concise description of the dialogue system used for our experiments in section 3. Section 4 is about the training process and the reward model. Section 5 goes into detail about dialogue policy op-

Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 792–799,
Prague, Czech Republic, June 2007.

timization with reinforcement learning. In section 6 we discuss our experimental results.

2 Related Work

Previous work has examined learning of effective dialogue strategies for information seeking spoken dialogue systems, and in particular the use of reinforcement learning methods to learn policies for action selection in dialogue management (see e.g. Levin et al., 2000; Walker, 2000; Scheffler and Young, 2002; Peek and Chickering, 2005; Frampton and Lemon, 2006), for selecting initiative and confirmation strategies (Singh et al., 2002); for detecting speech recognition problem (Litman and Pan, 2002); changing the dialogue according to the expertise of the user (Maloor and Chai, 2000); adapting responses according to previous interactions with the users (Rudary et al., 2004); optimizing mixed initiative in collaborative dialogue (English and Heeman, 2005), and optimizing confirmations (Cuayáhuitl et al., 2006). Other researchers have focussed their attention on the learning aspect of the task, examining, for example hybrid reinforcement/supervised learning (Henderson et al., 2005).

Previous work on learning dialogue management strategies has however generally been limited to well defined areas of the dialogue, in particular dealing with speech recognition and clarification problems, with small state spaces and a limited set of actions to choose from (Henderson et al., 2005). In a number of contexts, however, dialogues need to have a far greater degree of complexity not just in the number of states and possible actions but also in the variety of dialogue acts: for example in motivational dialogue systems where the task is not limited to information gathering, slot-filling or querying of a database, and where dialogues must contain more social and relational elements to be successful (for the usefulness of social dialogue see e.g. Bickmore, 2003; Liu and Picard, 2005). Only little effort has been directed to the question what dialogue components should make up the dialogue, involving decisions like how much and what type of social interaction should be used, different ways of forming a relationship with the user such as using chit-chat (for example asking about a user's hobbies or asking for the user's name), using humour, as well as the more conventional tasks of clarifying user input, establishing common ground and ensuring system replies are appropriate. Our work has focused on these aspects of dialogue strategy construction, in order to create good dialogue strategies incorporating appropriate levels of social interaction, humour, chit-chat, as well as successful information gathering and provision.

3 A Motivational Dialogue System

The domain of our system is physical exercise. The system is set up as an exercise advisor that asks the user what is preventing him/her from exercising more. After the user has worded his/her exercise 'barrier', the system will give motivational advice for how to overcome this barrier. As an illustration, Table 1 shows an example dialogue, generated by our system. Our dialogue system is *text*-based, so it does not involve speech recognition. This section describes the system's three main modules: the dialogue state/action manager, the utterance classifier and the dialogue component library.

Speaker	Utterance
system	*Good evening!*
system	*What's your name?*
user	**my name is Brenda**
system	*So Brenda, how are you today?*
user	**fine thank you**
system	*Tell me what is preventing you from exercising more.*
user	**I don't have anyone to exercise with...**
system	*So your barrier is Loneliness (e.g. You don't like exercising on your own).*
user	**sort of, yeah**
system	*A solution for this barrier is: Try going on your own and see whether it's really that different to going with a friend. Try talking to other people who are exercising, you're likely to find they welcome a little conversation!*
system	*Did you find this solution useful?*
user	**I guess...**
system	*Ok. Try to work on this to overcome your barrier.*
system	*Bye!*

Table 1: Example dialogue.

3.1 Dialogue state/action management

The dialogue state is unique at every stage of the conversation and is represented as a vector of feature-values. We use only a limited set of features because, as also noted in (Singh et al., 2002; Levin et al., 2000), it is important to keep the state space as small as possible (but with enough distinctive power to support learning) so we can construct a non-sparse Markov decision process (see section 5) based on our limited training dialogues. The state features are listed in Table 2.

Feature	Values	Description
curnode	$c \in N$	the current dialogue node
actiontype	$utt, trans$	action type
trigger	$t \in T$	utterance classifier category
confidence	$1, 0$	category confidence
problem	$1, 0$	communication problem earlier

Table 2: Dialogue state features.

In each dialogue state, the dialogue manager will look up the next action that should be taken. In our system, an action is either a *system utterance* or a *transition* in the dialogue structure. In the initial system, the dialogue structure was manually constructed. In many states, the next action requires a choice to be made. Dialogue states in which the system can choose among several possible actions are called *choice-states*. For example, in our system, immediately after greeting the user, the dialogue structure allows for different directions: the system can first ask some personal questions, or it can immediately discuss the main topic without any digressions. Utterance actions may also require a choice (e.g. *directive* versus *open* formulation of a question). In training mode, the system will make random choices in the choice-states. This approach will generate many different dialogue strategies, i.e. *paths* through the dialogue structure.

User replies are sent to an utterance classifier. The category assigned by this classifier is returned to the dialogue manager and triggers a transition to the next node in the dialogue structure. The system also accommodates a simple rule-based extraction module, which can be used to extract information from user utterances (e.g. the user's name, which is templated in subsequent system prompts in order to personalize the dialogue).

3.2 Utterance classification

The (memory-based) classifier uses a rich set of features for accurate classification, including words, phrases, regular expressions, domain-specific word-relations (using a taxonomy-plugin) and syntactically motivated expressions. For utterance parsing we used a memory-based shallow parser, called MBSP (Daelemans et al., 1999). This parser provides part of speech labels, chunk brackets, subject-verb-object relations, and has been enriched with detection of negation scope and clause boundaries.

The feature-matching mechanism in our classification system can match terms or phrases at specified positions in the token stream of the utterance, also in combination with syntactic and semantic class labels. This allows us to define features that are particularly useful for resolving confusing linguistic phenomena like ambiguity and negation. A base feature set was generated automatically, but quite a lot of features were manually tuned or added to cope with certain common dialogue situations. The overall classification accuracy, measured on the dialogues that were produced during the training phase, is 93.6%. Average precision/recall is 98.6/97.3% for the non-barrier categories (*confirmation*, *negation*, *unwillingness*, etc.), and 99.1/83.4% for the barrier categories (*injury*, *lack of motivation*, etc.).

3.3 Dialogue Component Library

The dialogue component library contains generic as well as task-/domain-specific dialogue content, combining different aspects of dialogue (task/topic structure, communication goals, etc.). Table 3 lists all components in the library that was used for training our dialogue system. A dialogue component is basically a coherent set of dialogue node representations with a certain dialogue function. The library is set up in a flexible, generic way: new components can easily be plugged in to test their usefulness in different dialogue contexts or for new domains.

4 Training the Dialogue System

4.1 Random strategy generation

In its training mode, the dialogue system uses random exploration: it generates different dialogue strategies by choosing randomly among the *allowed actions* in the choice-states. Note that dialogue generation is constrained to contain certain fixed actions that are essential for task completion (e.g. asking the exercise barrier, giving a solution, closing the session). This excludes a vast number of useless strategies from exploration by the system. Still, given all action choices and possible user reactions, the total number of unique dialogues that can be generated by

Component	Description	p_a	p_e
StartSession	Dialogue openings, including various greetings	•	•
PersonalQuestionnaire	Personal questions, e.g. name; age; hobbies; interests, *how are you today?*	•	
ElizaChitChat	Eliza-like chit-chat, e.g. *please go on...*		
ExerciseChitChat	Chit-chat about exercise, e.g. *have you been doing any exercise this week?*	◦	
Barrier	Prompts concerning the barrier, e.g. ask the barrier; barrier verification; ask a rephrase	•	•
Solution	Prompts concerning the solution, e.g. give the solution; verify usefulness	•	•
GiveBenefits	Talk about the benefits of exercising		
AskCommitment	Ask user to commit his implementation of the given solution	•	
Encourage	Encourage the user to work on the given solution	•	•
GiveJoke	The humor component: ask if the user wants to hear a joke; tell random jokes	◦	•
VerifyCloseSession	Verification for closing the session (*are you sure you want to close this session?*)	◦	◦
CloseSession	Dialogue endings, including various farewells	•	•

Table 3: Components in the dialogue component library. The last two columns show which of the components was used in the learned policy (p_a) and the expert policy (p_e), discussed in section 6. • means the component is always used, ◦ means it is sometimes used, depending on the dialogue state.

the system is approximately 345000 (many of which are unlikely to ever occur). During training, the system generated 490 different strategies. There are 71 choice-states that can actually occur in a dialogue. In our training dialogues, the opening state was obviously visited most frequently (572 times), almost 60% of all states was visited at least 50 times, and only 16 states were visited less than 10 times.

4.2 The reward model

When the dialogue has reached its final state, a survey is presented to the user for dialogue evaluation. The survey consists of five statements that can each be rated on a five-point scale (indicating the user's level of agreement). The responses are mapped to rewards of -2 to 2. The statements we used are partly based on the user survey that was used in (Singh et al., 2002). We considered these statements to reflect the most important aspects of conversation that are relevant for learning a good dialogue policy. The five statements we used are listed below.

M1 *Overall, this conversation went well*

M2 *The system understood what I said*

M3 *I knew what I could say at each point in the dialogue*

M4 *I found this conversation engaging*

M5 *The system provided useful advice*

4.3 Training set-up

Eight subjects carried out a total of 572 conversations with the system. Because of the variety of possible exercise barriers known by the system (52 in total) and the fact that some of these barriers are more complex or harder to detect than others, the system's classification accuracy depends largely on the user's barrier. To prevent classification accuracy distorting the user evaluations, we asked the subjects to act as if they had one of five predefined exercise barriers (e.g. *Imagine that you don't feel comfortable exercising in public. See what the advisor recommends for this barrier to your exercise*).

5 Dialogue Policy Optimization with Reinforcement Learning

Reinforcement learning refers to a class of machine learning algorithms in which an agent explores an environment and takes actions based on its current state. In certain states, the environment provides a reward. Reinforcement learning algorithms attempt to find the optimal policy, i.e. the policy that maximizes cumulative reward for the agent over the course of the problem. In our case, a policy can be seen as a mapping from the dialogue states to the possible actions in those states. The environment is typically formulated as a Markov decision process (MDP).

The idea of using reinforcement learning to automate the design of strategies for dialogue systems was first proposed by Levin et al. (2000) and has subsequently been applied in a.o. (Walker, 2000; Singh et al., 2002; Frampton and Lemon, 2006; Williams et al., 2005).

5.1 Markov decision processes

We follow past lines of research (such as Levin et al., 2000; Singh et al., 2002) by representing a dialogue as a trajectory in the state space, determined by the user responses and system actions: $s_1 \xrightarrow{a_1,r_1} s_2 \xrightarrow{a_2,r_2} \ldots s_n \xrightarrow{a_n,r_n} s_{n+1}$, in which $s_i \xrightarrow{a_i,r_i} s_{i+1}$ means that the system performed action a_i in state s_i, received[1] reward r_i and changed to state s_{i+1}. In our system, a state is a dialogue context vector of feature values. This feature vector contains the available information about the dialogue so far that is relevant for deciding what action to take next in the current dialogue state. We want the system to learn the optimal decisions, i.e. to choose the actions that maximize the expected reward.

5.2 Q-value iteration

The field of reinforcement learning includes many algorithms for finding the optimal policy in an MDP (see Sutton and Barto, 1998). We applied the algorithm of (Singh et al., 2002), as their experimental set-up is similar to ours, consisting of: generation of (limited) exploratory dialogue data, using a training system; creating an MDP from these data and the rewards assigned by the training users; off-line policy learning based on this MDP.

The Q-function for a certain action taken in a certain state describes the total reward expected between taking that action and the end of the dialogue. For each state-action pair (s, a), we calculated this expected cumulative reward $Q(s, a)$ of taking action a from state s, with the following equation (Sutton and Barto, 1998; Singh et al., 2002):

$$Q(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a) \max_{a'} Q(s',a') \quad (1)$$

where: $P(s'|s,a)$ is the probability of a transition from state s to state s' by taking action a, and $R(s,a)$ is the expected reward obtained when taking action a in state s. γ is a weight ($0 \leq \gamma \leq 1$), that discounts rewards obtained later in time when it is set to a value < 1. In our system, γ was set to 1. Equation 1 is recursive: the Q-value of a certain state is computed in terms of the Q-values of its successor states. The Q-values can be estimated to within a desired threshold using Q-value iteration (Sutton and Barto, 1998). Once the value iteration process is completed, by selecting the action with the maximum Q-value (the maximum expected future reward) at each choice-state, we can obtain the optimal dialogue policy π.

[1] In our experiments, we did not make use of immediate rewarding (e.g. at every turn) during the conversation. Rewards were given after the final state of the dialogue had been reached.

6 Results and Discussion

6.1 Reward analysis

Figure 1 shows a graph of the distribution of the five different evaluation measures in the training data (see section 4.2 for the statement wordings). M1 is probably the most important measure of success. The distribution of this reward is quite symmetrical, with a slightly higher peak in the positive area. The distribution of M2 shows that M1 and M2 are related. From the distribution of M4 we can conclude that the majority of dialogues during the training phase was not very engaging. Users obviously had a good feeling about what they could say at each point in the dialogue (M3), which implies good quality of the system prompts. The judgement about the usefulness of the provided advice is pretty average, tending a bit more to negative than to positive. We do think that this measure might be distorted by the fact that we asked the subjects to *imagine* that they have the given exercise barriers. Furthermore, they were sometimes confronted with advice that had already been presented to them in earlier conversations.

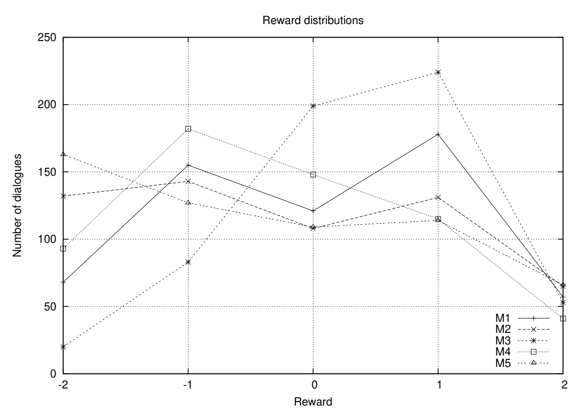

Figure 1: Reward distributions in the training data.

In our analysis of the users' rewarding behavior, we found several significant correlations. We found that longer dialogues (> 3 user turns) are appreciated more than short ones (< 4 user turns), which seems rather logical, as dialogues in which the user barely gets to say anything are neither natural nor engaging.

We also looked at the relationship between user input verification and the given rewards. Our intuition is that the choice of barrier verification is one of the most important choices the system can make in the dialogue. We found that it is much better to first verify the detected barrier than to immediately give advice. The percentage of appropriate advice provided in dialogues with barrier verification is significantly higher than in dialogues without verification.

In several states of the dialogue, we let the system choose from different wordings of the system prompt. One of these choices is whether to use an open question to ask what the user's barrier is (*How can I help you?*), or a directive question (*Tell me what is preventing you from exercising more.*). The motivation behind the open question is that the user gets the initiative and is basically free to talk about anything he/she likes. Naturally, the advantage of directive questions is that the chance of making classification errors is much lower than with open questions because the user will be better able to assess what kind of answer the system expects. Dialogues in which the key-question (asking the user's barrier) was directive, were rewarded more positively than dialogues with the open question.

6.2 Learned dialogue policies

We learned a different policy for each evaluation measure separately (by only using the rewards given for that particular measure), and a policy based on a combination (sum) of the rewards for all evaluation measures. We found that the learned policy based on the combination of all measures, and the policy based on measure M1 alone (*Overall, this conversation went well*) were nearly identical. Table 4 compares the most important decisions of the different policies. For convenience of comparison, we only listed the main, structural choices. Table 3 shows which of the dialogue components in the library were used in the learned and the expert policy.

Note that, for the sake of clarity, the state descriptions in Table 4 are basically summaries of a set of more specific states since a state is a specific representation of the dialogue context at a particular moment (composed of the values of the features listed in Table 2). For instance, in the p_a policy, the decision in the last row of the table (give a joke or not), depends on whether or not there has been a classification failure (i.e. a communication problem earlier in the dialogue). If there has been a classification failure, the policy prescribes the decision *not* to give a joke, as it was not appreciated by the training users in that context. Otherwise, if there were no communication problems during the conversation, the users *did* appreciate a joke.

6.3 Evaluation

We compared the learned dialogue policy with a policy which was independently hand-designed by experts[2] for this system. The decisions made in the learned strategy were very similar to the ones made by the experts, with only a few differences, indicating that the automated method would indeed perform as well as an expert. The main differences were the inclusion of a personal questionnaire for relation building at the beginning of the dialogue and a commitment question at the end of the dialogue. Another difference was the more restricted use of the humour element, described in section 6.2 which turns out to be intuitively better than the expert's decision to simply always include a joke. Of course, we can only draw conclusions with regard to the effectiveness of these two policies if we empirically compare them with real test users. Such evaluations are planned as part of our future research.

As some additional evidence against the possibility that the learned policy was generated by chance, we performed a simple experiment in which we took several random samples of 300 training dialogues from the complete training set. For each sample, we learned the optimal policy. We mutually compared these policies and found that they were very similar: only in 15-20% of the states, the policies disagreed on which action to take next. On closer inspection we found that this disagreement mainly concerned states that were poorly visited (1-10 times) in these samples. These results suggest that the learned policy is unreliable at infrequently visited states. Note however, that all main decisions listed in Table 4 are

[2]The experts were a team made up of psychologists with experience in the psychology of health behaviour change and a scientist with experience in the design of automated dialogue systems.

State description	Action choices	p_1	p_2	p_3	p_4	p_5	p_a	p_e
After greeting the user	- ask the exercise barrier		●			●		●
	- ask personal information	●		●	●		●	
	- chit-chat about exercise							
When asking the barrier	- use a directive question	●	●	●	●	●	●	●
	- use an open question							
User gives exercise barrier	- verify detected barrier	●	●	●	●	●	●	●
	- give solution							
User rephrased barrier	- verify detected barrier	●		●	●	●	●	●
	- give solution		●					
Before presenting solution	- ask if the user wants to see a solution for the barrier			●				
	- give a solution	●	●		●	●	●	●
After presenting solution	- verify solution usefulness	●	●	●	●		●	●
	- encourage the user to work on the given solution					●		
	- ask user to commit solution implementation							
User found solution useful	- encourage the user to work on the solution			●	●	●		●
	- ask user to commit solution implementation	●	●				●	
User found solution not useful	- give another solution	●	●	●	●	●	●	●
	- ask the user wants to propose his own solution							
After giving second solution	- verify solution usefulness	●						●
	- encourage the user to work on the given solution		●		●	●	●	
	- ask user to commit solution implementation			●				
End of dialogue	- close the session		●		●	●		
	- ask if the user wants to hear a joke	●		●			●	●

Table 4: Comparison of the most important decisions made by the learned policies. p_n is the policy based on evaluation measure n; p_a is the policy based on all measures; p_e contains the decisions made by experts in the manually designed policy.

made at frequently visited states. The only disagreement in frequently visited states concerned system-prompt choices. We might conclude that these particular (often very subtle) system-prompt choices (e.g. careful versus direct formulation of the exercise barrier) are harder to learn than the more noticable dialogue structure-related choices.

7 Conclusions and Future Work

We have explored reinforcement learning for automatic dialogue policy optimization in a question-based motivational dialogue system. Our system can automatically compose a dialogue strategy from a library of dialogue components, that is very similar to a manually designed expert strategy, by learning from user feedback.

Thus, in order to build a new dialogue system, dialogue system engineers will have to set up a rough dialogue template containing several 'multiple choice'-action nodes. At these nodes, various dialogue components or prompt wordings (e.g. entertaining parts, clarification questions, social dialogue, personal questions) from an existing or self-made library can be plugged in without knowing beforehand which of them would be most effective.

The automatically generated dialogue policy is very similar (see Table 4) –but arguably improved in many details– to the hand-designed policy for this system. Automatically learning dialogue policies also allows us to test a number of interesting issues in parallel, for example, we have learned that users appreciated dialogues that were longer, starting with some personal questions (e.g *What is your name?, What are your hobbies?*). We think that altogether, this relation building component gave the dialogue a more natural and engaging character, although it was left out in the expert strategy.

We think that the methodology described in this paper may be able to yield more effective dialogue policies than experts. Especially in complicated dialogue systems with large state spaces. In our system, state representations are composed of multiple context feature values (e.g. communication problem earlier in the dialogue, the confidence of the utterance classifier). Our experiments showed that sometimes different decisions were learned in dialogue contexts where only one of these features was different (for example use humour only if the system has been successful in recognising a user's exercise barrier): all context features are implicitly used to learn the optimal decisions and when hand-designing a dialogue policy, experts can impossibly take into account all possible different dialogue contexts.

With respect to future work, we plan to examine the impact of different state representations. We did not yet empirically compare the effects of each feature on policy learning or experiment with other features than the ones listed in Table 2. As Tetreault and Litman (2006) show, incorporating more or different information into the state representation might however result in different policies.

Furthermore, we will evaluate the actual genericity of our approach by applying it to different domains. As part of that, we will look at automatically mining libraries of dialogue components from existing dialogue transcript data (e.g. available scripts or transcripts of films, tv series and interviews containing real-life examples of different types of dialogue). These components can then be plugged into our current adaptive system in order to discover what works best in dialogue for new domains. We should note here that extending the system's dialogue component library will automatically increase the state space and thus policy generation and optimization will become more difficult and require more training data. It will therefore be very important to carefully control the size of the state space and the global structure of the dialogue.

Acknowledgements

The authors would like to thank Piroska Lendvai Rudenko, Walter Daelemans, and Bob Hurling for their contributions and helpful comments. We also thank the anonymous reviewers for their useful comments on the initial version of this paper.

References

Timothy W. Bickmore. 2003. *Relational Agents: Effecting Change through Human-Computer Relationships*. Ph.D. Thesis, MIT, Cambridge, MA.

Heriberto Cuayáhuitl, Steve Renals, Oliver Lemon, and Hiroshi Shimodaira. 2006. Learning multi-goal dialogue strategies using reinforcement learning with reduced state-action spaces. *Proceedings of Interspeech-ICSLP*.

Walter Daelemans, Sabine Buchholz, and Jorn Veenstra. 1999. Memory-Based Shallow Parsing. *Proceedings of CoNLL-99*, Bergen, Norway.

Michael S. English and Peter A. Heeman 2005. Learning Mixed Initiative Dialog Strategies By Using Reinforcement Learning On Both Conversants. *Proceedings of HLT/NAACL*.

Matthew Frampton and Oliver Lemon. 2006. Learning More Effective Dialogue Strategies Using Limited Dialogue Move Features. *Proceedings of the Annual Meeting of the ACL*.

James Henderson, Oliver Lemon, and Kallirroi Georgila. 2005. Hybrid Reinforcement/Supervised Learning for Dialogue Policies from COMMUNICATOR Data. *IJCAI workshop on Knowledge and Reasoning in Practical Dialogue Systems*.

Esther Levin, Roberto Pieraccini, and Wieland Eckert. 2000. A Stochastic Model of Human-Machine Interaction for Learning Dialog Strategies. *IEEE Trans. on Speech and Audio Processing*, Vol. 8, No. 1, pp. 11-23.

Diane J. Litman and Shimei Pan. 2002. Designing and Evaluating an Adaptive Spoken Dialogue System. *User Modeling and User-Adapted Interaction,* Volume 12, Issue 2-3, pp. 111-137.

Karen K. Liu and Rosalind W. Picard. 2005. Embedded Empathy in Continuous, Interactive Health Assessment. *CHI Workshop on HCI Challenges in Health Assessment,* Portland, Oregon.

Preetam Maloor and Joyce Chai. 2000. Dynamic User Level and Utility Measurement for Adaptive Dialog in a Help-Desk System. *Proceedings of the 1st Sigdial Workshop*.

Tim Paek and David M. Chickering. 2005. The Markov Assumption in Spoken Dialogue Management. *Proceedings of SIGDIAL 2005*.

Matthew Rudary, Satinder Singh, and Martha E. Pollack. 2004. Adaptive cognitive orthotics: Combining reinforcement learning and constraint-based temporal reasoning. *Proceedings of the 21st International Conference on Machine Learning*.

Konrad Scheffler and Steve Young. 2002. Automatic learning of dialogue strategy using dialogue simulation and reinforcement learning. *Proceedings of HLT-2002*.

Satinder Singh, Diane Litman, Michael Kearns, and Marilyn Walker. 2002. Optimizing Dialogue Management with Reinforcement Learning: Experiments with the NJFun System. *Journal of Artificial Intelligence Research (JAIR)*, Volume 16, pages 105-133.

Richard S. Sutton and Andrew G. Barto. 1998. *Reinforcement Learning*. MIT Press.

Joel R. Tetreault and Diane J. Litman 2006. Comparing the Utility of State Features in Spoken Dialogue Using Reinforcement Learning. *Proceedings of HLT/NAACL*, New York.

Marilyn A. Walker 2000. An Application of Reinforcement Learning to Dialogue Strategy Selection in a Spoken Dialogue System for Email. *Journal of Artificial Intelligence Research*, Vol 12., pp. 387-416.

Jason D. Williams, Pascal Poupart, and Steve Young. 2005. Partially Observable Markov Decision Processes with Continuous Observations for Dialogue Management. *Proceedings of the 6th SigDial Workshop*, September 2005, Lisbon.