Comparing the Utility of State Features in Spoken Dialogue Using Reinforcement Learning

Joel R. Tetreault
University of Pittsburgh
Learning Research and Development Center
Pittsburgh PA, 15260, USA
tetreaul@pitt.edu

Diane J. Litman
University of Pittsburgh
Department of Computer Science
Learning Research and Development Center
Pittsburgh PA, 15260, USA
litman@cs.pitt.edu

Abstract

Recent work in designing spoken dialogue systems has focused on using Reinforcement Learning to automatically learn the best action for a system to take at any point in the dialogue to maximize dialogue success. While policy development is very important, choosing the best features to model the user state is equally important since it impacts the actions a system should make. In this paper, we compare the relative utility of adding three features to a model of user state in the domain of a spoken dialogue tutoring system. In addition, we also look at the effects of these features on what type of a question a tutoring system should ask at any state and compare it with our previous work on using feedback as the system action.

1 Introduction

A host of issues confront spoken dialogue system designers, such as choosing the best system action to perform given any user state, and also selecting the right features to best represent the user state. While recent work has focused on using Reinforcement Learning (RL) to address the first issue (such as (Walker, 2000), (Henderson et al., 2005), (Williams et al., 2005a)), there has been very little empirical work on the issue of feature selection in prior RL approaches to dialogue systems. In this paper, we use a corpus of dialogues of humans interacting with a spoken dialogue tutoring system to show the comparative utility of adding the three features of *concept repetition*, *frustration level*, and *student performance*. These features are not just unique to the tutoring domain but are important to dialogue systems in general. Our empirical results show that these features all lead to changes in what action the system should take, with concept repetition and frustration having the largest effects.

This paper extends our previous work (Tetreault and Litman, 2006) which first presented a methodology for exploring whether adding more complex features to a representation of student state will beneficially alter tutor actions with respect to feedback. Here we present an empirical method of comparing the effects of each feature while also generalizing our findings to a different action choice of what type of follow-up question should a tutor ask the student (as opposed to what type of feedback should the tutor give). In complex domains such as tutoring, testing different policies with real or simulated students can be time consuming and costly so it is important to properly choose the best features before testing, which this work allows us to do. This in turn aids our long-term goal of improving a spoken dialogue system that can effectively adapt to a student to maximize their learning.

2 Background

We follow past lines of research (such as (Levin and Pieraccini, 1997) and (Singh et al., 1999)) for describing a dialogue d as a trajectory within a Markov Decision Process (MDP) (Sutton and Barto, 1998).

A MDP has four main components: 1: *states* s, 2: *actions* A, 3: a *policy* π, which specifies what is the best action to take in a state, and 4: a *reward function* R which specifies the worth of the entire process. Dialogue management is easily described using a MDP because one can consider the actions as actions made by the system, the state as the dialogue context (which can be viewed as a vector of features, such as ASR confidence or dialogue act), and a reward which for many dialogue systems tends to be task completion success or dialogue length.

Another advantage of using MDP's to model a dialogue space, besides the fact that the primary MDP parameters easily map to dialogue parameters, is the notion of delayed reward. In a MDP, since rewards are often not given until the final states, dynamic programming is used to propagate the rewards back to the internal states to weight the value of each state (called the V-value), as well as to develop an optimal policy π for each state of the MDP. This propagation of reward is done using the *policy iteration* algorithm (Sutton and Barto, 1998) which iteratively updates the V-value and best action for each state based on the values of its neighboring states.

The V-value of each state is important for our purposes not only because it describes the relative worth of a state within the MDP, but as more data is added when building the MDP, the V-values should stabilize, and thus the policies stabilize as well. Since, in this paper, we are comparing policies in a fixed data set it is important to show that the policies are indeed reliable, and not fluctuating.

For this study, we used the MDP infrastructure designed in our previous work which allows the user to easily set state, action, and reward parameters. It then performs policy iteration to generate a policy and V-values for each state. In the following sections, we discuss our corpus, methodology, and results.

3 Corpus

For our study, we used an annotated corpus of 20 human-computer spoken dialogue tutoring sessions (for our work we use the ITSPOKE system (Litman and Silliman, 2004) which uses the text-based Why2-ATLAS dialogue tutoring system as its "back-end" (VanLehn et al., 2002)). The content of the system, and all possible dialogue paths, were authored by physics experts. Each session consists of an interaction with one student over 5 different college-level physics problems, for a total of 100 dialogues. Before each session, the student is asked to read physics material for 30 minutes and then take a pretest based on that material. Each problem begins with the student writing out a short essay response to the question posed by the computer tutor. The fully-automated system assesses the essay for potential flaws in the reasoning and then starts a dialogue with the student, asking questions to help the student understand the confused concepts. The tutor's response and next question is based only on the correctness of the student's last answer. Informally, the dialogue follows a question-answer format. Once the student has successfully completed the dialogue section, he is asked to correct the initial essay. Each of the dialogues takes on average 20 minutes and 60 turns. Finally, the student is given a posttest similar to the pretest, from which we can calculate their normalized learning gain: $NLG = \frac{posttest - pretest}{1 - pretest}$.

State Feature	Values
Certainty	Certain (cer) Uncertain (unc) Neutral (neu)
Frustration	Frustrated (F) Neutral (N),
Correctness	Correct (C) Partially Correct (PC) Incorrect (I)
Percent Correct	50-100% (H)igh 0-49% (L)ow
Concept Repetition	Concept is new (0) Concept is repeated (R)

Table 1: Potential Student State Features in MDP

Prior to our study, the corpus was annotated for Tutor Moves, which can be viewed as Dialogue Acts (Forbes-Riley et al., 2005) [1] and consisted of Tutor Feedback, Question and State Acts. In this corpus, a turn can consist of multiple utterances and thus can be labeled with multiple moves. For example, a tutor can give positive feedback and then ask a question in the same turn. What type of question to ask will be the action choice addressed in this paper.

As for features to include in the student state, we annotated five features as shown in Table 1. Two

[1] The Dialogue Act annotation had a Kappa of 0.67.

Action	Example Turn
SAQ	"Good. What is the direction of that force relative to your fi st?"
CAQ	"What is the defi nition of Newton's Second Law?"
Mix	"Good. If it doesn't hit the center of the pool what do you know about the magnitude of its displacement from the center of the pool when it lands? Can it be zero? Can it be nonzero?"
NoQ	"So you can compare it to my response..."

Table 2: Tutor Actions for MDP

emotion related features, certainty and frustration, were annotated manually prior to this study (Forbes-Riley and Litman, 2005) [2]. Certainty describes how confident a student seemed to be in his answer, while frustration describes how frustrated the student seemed to be when he responded. We include three other automatically extracted features for the Student state: (1) Correctness: whether the student was correct or not; (2) Percent Correct: percentage of correctly answered questions so far for the current problem; (3) Concept Repetition: whether the system is forced to cover a concept again which reflects an area of difficulty for the student.

4 Experimental Method

The goal of this study is to quantify the utility of adding a feature to a baseline state space. We use the following four step process: (1) establish an action set and reward function to be used as constants throughout the test since the state space is the one MDP parameter that will be changed during the tests; (2) establish a baseline state and policy, and (3) add a new feature to that state and test if adding the feature results in policy changes. Every time we create a new state, we make sure that the generated V-values converge. Finally, (4), we evaluate the effects of adding a new feature by using three metrics: (1) number of policy changes (diffs), (2) % policy change, and (3) Expected Cumulative Reward. These three metrics are discussed in more detail in Section 5.2. In this section we focus on the first three steps of the methodology.

4.1 Establishing Actions and Rewards

We use questions as our system action A in our MDP. The action size is 4 (tutor can ask a simple answer question (SAQ), a complex answer question (CAQ), or a combination of the two (Mix), or not ask a question (NoQ)). Examples from our corpus can be seen in Table 2. We selected this as the action because what type of question a tutor should ask is of great interest to the Intelligent Tutoring Systems community, and it generalizes to dialogue systems since asking users questions of varying complexity can elicit different responses.

For the dialogue reward function R we did a median split on the 20 students based on their normalized learning gain, which is a standard evaluation metric in the Intelligent Tutoring Systems community. So 10 students and their respective 5 dialogues were assigned a positive reward of 100 (high learners), and the other 10 students and their respective 5 dialogues were assigned a negative reward of -100 (low learners). The final student turns in each dialogue were marked as either a positive final state (for a high learner) or a negative final state (for a low learner). The final states allow us to propagate the reward back to the internal states. Since no action is taken from the final states, their V-values remain the same throughout policy iteration.

4.2 Establishing a Baseline State and Policy

Currently, our tutoring system's response to a student depends only on whether or not the student answered the last question correctly, so we use correctness as the sole feature in our baseline dialogue state. A student can either be correct, partially correct, or incorrect. Since partially correct responses occur infrequently compared to the other two, we reduced the state size to two by combining Incorrect and Partially Correct into one state (IPC) and keeping Correct (C).

With the actions, reward function, and baseline state all established, we use our MDP tool to generate a policy for both states (see Table 3). The second column shows the states, the third, the policy determined by our MDP toolkit (i.e. the optimal action to

[2] In a preliminary agreement study, a second annotator labeled the entire corpus for *uncertain* versus *other*, yielding 90% inter-annotator agreement (0.68 Kappa).

take in that state with respect to the final reward) and finally how many times the state occurs in our data (state size). So if a student is correct, the best action is to give something other than a question immediately, such as feedback. If the student is incorrect, the best policy is to ask a combination of short and complex answer questions.

#	State	Policy	State Size
1	C	NoQ	1308
2	IPC	Mix	872

Table 3: Baseline Policy

The next step in our experiment is to test whether the policies generated are indeed reliable. Normally, the best way to verify a policy is to conduct experiments and see if the new policy leads to a higher reward for new dialogues. In our context, this would entail running more subjects with the augmented dialogue manager, which could take months. So, instead we check if the polices and values for each state are indeed converging as we add data to our MDP model. The intuition here is that if both of those parameters were varying between a corpus of 19 students versus one of 20 students, then we can't assume that our policy is stable, and hence is not reliable.

To test this out, we made 20 random orderings of our students to prevent any one ordering from giving a false convergence. Each ordering was then passed to our MDP infrastructure such that we started with a corpus of just the first student of the ordering and then determined a MDP policy for that cut, then incrementally added one student at a time until we had added all 20 students. So at the end, 20 random orderings with 20 cuts each provides 400 MDP trials. Finally, we average each cut across the 20 random orderings. The first graph in Figure 1 shows a plot of the average V-values against a cut. The state with the plusses is the positive final state, and the one at the bottom is the negative final state. However, we are most concerned with how the non-final states converge. The plot shows that the V-values are fairly stable after a few initial cuts, and we also verified that the policies remained stable over the 20 students as well (see our prior work (Tetreault and Litman, 2006) for details of this method). Thus we can be sure that our baseline policy is indeed reliable for our corpus.

5 Results

In this section, we investigate whether adding more information to our student state will lead to interesting policy changes. First, we add certainty to our baseline of correctness because prior work (such as (Bhatt et al., 2004), (Liscombe et al., 2005) and (Forbes-Riley and Litman, 2005)) has shown the importance of considering certainty in tutoring systems. We then compare this new baseline's policy (henceforth Baseline 2) with the policies generated when frustration, concept repetition, and percent correctness are included.

We'll first discuss the new baseline state. There are three types of certainty: certain (cer), uncertain (unc), and neutral (neu). Adding these to our state representation increases state size from 2 to 6. The new policy is shown in Table 4. The second and third columns show the original baseline states and their policies. The next column shows the new policy when splitting the original state into the three new states based on certainty (with the policies that differ from the baseline shown in bold). The final column shows the size of each new state. So the first row indicates that if the student is correct and certain, one should give a combination of a complex and short answer question; if the student is correct and neutral, just ask a SAQ; and else if the student is correct and uncertain, give a Mix. The overall trend of adding the certainty feature is that if the student exhibits some emotion (either they are certain or uncertain), the best response is Mix, but for neutral do something else.

#	State	Baseline	Baseline 2	B2 State Size
1	C	NoQ	certain:C **Mix**	663
			neutral:C **SAQ**	480
			uncertain:C **Mix**	165
2	IPC	Mix	certain:IPC Mix	251
			neutral:IPC **NoQ**	377
			uncertain:IPC Mix	244

Table 4: Baseline 2 Policy

We assume that if a feature is important to include in a state representation it should change the policies of the old states. For example, if certainty did not impact how well students learned (as deemed by the MDP) then the policies for certainty, uncertainty,

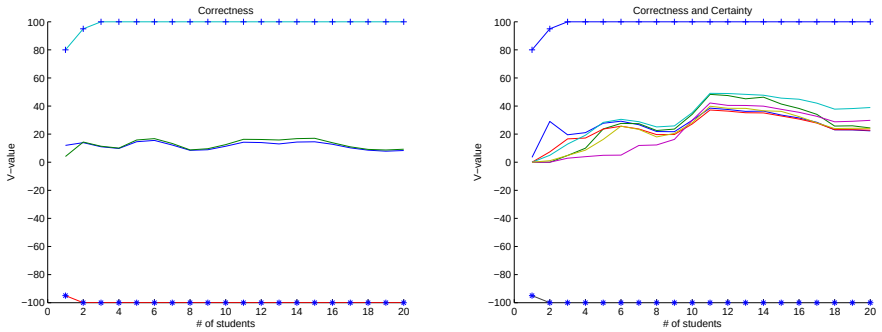

Figure 1: Baseline 1 and 2 Convergence Plots

and neutral would be the same as the original policy for Correct (C) or Incorrect (IPC). However, the figures show otherwise. When certainty is added to the state, only two new states (incorrect while being certain or uncertain) retain the old policy of having the tutor give a mix of SAQ and CAQ. The right graph in Figure 1 shows that for Baseline 2, V-values tend to converge around 10 cuts.

Next, we add Concept Repetition, Frustration, and Percent Correct features individually to Baseline 2. For each of the three features we repeated the reliability check of plotting the V-value convergence and found that the graphs showed convergence around 15 students.

5.1 Feature Addition Results

Policies for the three new features are shown in Table 5 with the policies that differ from Baseline 2's shown in bold. The numbers in parentheses refer to the size of the new state (so for the first +Concept state, there are 487 instances in the data of a student being correct, certain after hearing a new concept).

Concept Repetition Feature As shown in column 4, the main trend of incorporating concept repetition usually is to give a complex answer question after a concept has been repeated, and especially if the student is correct when addressing a question about the repeated concept. This is intuitive because one would expect that if a concept has been repeated, it signals that the student did not grasp the concept initially and a clarification dialogue was initiated to help the student learn the concept. Once the student answers the repeated concept correctly, it signals that the student understands the concept and that the tutor can once again ask more difficult questions to challenge the student. Given the amount of differences in the new policy and the original policy (10 out of 12 possible), including concept repetition as a state feature has a significant impact on the policy generated.

Frustration Feature Our results show that adding frustration changes the policies the most when the student is frustrated, but when the student isn't frustrated (neutral) the policy stays the same as the baseline with the exception of when the student is Correct and Certain (state 1), and Incorrect and Uncertain (state 6). It should be noted that for states 2 through 6, that frustration occurs very infrequently so the policies generated (CAQ) may not have enough data to be totally reliable. However in state 1, the policy when the student is confident and correct but also frustrated is to simply give a hint or some other form of feedback. In short, adding the frustration feature results in a change in 8 out of 12 policies.

Percent Correctness Feature Finally, the last column, shows the new policy generated for incorporating a simple model of current student performance within the dialog. The main trend is to give a Mix of SAQ and CAQ's. Since the original policy was to give a lot of Mix's in the first place, adding this feature does not result in a large policy change, only 4 differences.

5.2 Feature Comparison

To compare the utility of each of the features, we use three metrics: (1) Diff's (2) % Policy Change, and (3) Expected Cumulative Reward. # of Diff's are the number of states whose policy differs from the baseline policy, The second column of Table 6

#	State	Baseline 2	+Concept	+Frustration	+ % Correctness
1	certain:C	Mix (663)	0: **CAQ** (487) R: **CAQ** (176)	N: **SAQ** (558) F: **NoQ** (105)	H: Mix (650) L: Mix (13)
2	certain:IPC	Mix (251)	0: **SAQ** (190) R: **NoQ** (61)	N: Mix (215) F: **CAQ** (36)	H: Mix (217) L: Mix (34)
3	neutral:C	SAQ (480)	0: **CAQ** (328) R: **CAQ** (152)	N: SAQ (466) F: **CAQ** (14)	H: **Mix** (468) L: **Mix** (12)
4	neutral:IPC	NoQ (377)	0: NoQ (289) R: **Mix** (88)	N: NoQ (364) F: **CAQ** (13)	H: NoQ (320) L: **Mix** (57)
5	uncertain:C	Mix (165)	0: Mix (127) R: **CAQ** (38)	N: Mix (151) F: **CAQ** (14)	H: Mix (156) L: Mix (9)
6	uncertain:IPC	Mix (244)	0: **SAQ** (179) R: **CAQ**(65)	N: **CAQ** (209) F: **CAQ** (35)	H: **CAQ** (182) L: Mix (62)

Table 5: Question Policies

summarizes the amount of Diff's for each new feature compared to Baseline 2. Concept Repetition has the largest number of differences: 10, followed by Frustration, and then Percent Correctness. However, counting the number of differences does not completely describe the effect of the feature on the policy. For example, it is possible that a certain feature may impact the policy for several states that occur infrequently, resulting in a lot of differences but the overall impact may actually be lower than a certain feature that only impacts one state, since that state occurs a majority of the time in the data. So we weight each difference by the number of times that state-action sequence actually occurs in the data and then divide by the total number of state-action sequences. This weighting, % Policy Change (or % P.C.), allows us to more accurately depict the impact of adding the new feature. The third columns shows the weighted figures of % Policy Change. As an additional confirmation of the ranking, we use Expected Cumulative Reward (E.C.R.). One issue with % Policy Change is that it is possible that frequently occurring states have very low V-values so the expected utility from starting the dialogue could potentially be lower than a state feature with low % Policy Change. E.C.R. is calculated by normalizing the V-value of each state by the number of times it occurs as a start state in a dialogue and then summing over all states. The upshot of both metrics is the ranking of the three features remains the same with Concept Repetition effecting the greatest change in what a tutoring system should do; Percent Correctness has the least effect.

We also added a random feature to Baseline 2

State Feature	# Diff's	% P.C.	E.C.R
Concept Repetition	10	80.2%	39.52
Frustration	8	66.4%	31.30
Percent Correctness	4	44.3%	28.17

Table 6: Question Act Results

State Feature	# Diff's	% P.C.	E.C.R
Concept Repetition	4	34.6%	43.43
Frustration	3	6.0%	25.80
Percent Correctness	3	10.3%	26.41

Table 7: Feedback Act Results

with one of two values (0 and 1) to serve as a baseline for the # of Diff's. In a MDP with a large enough corpus to explore, a random variable would not alter the policy, however with a smaller corpus it is possible for such a variable to alter policies. We found that by testing a random feature 40 times and averaging the diffs from each test, resulted in an average diff of 5.1. This means that Percent Correctness effects a smaller amount of change than this random baseline and thus is fairly useless as a feature to add since the random feature is probably capturing some aspect of the data that is more useful. However, the Concept Repetition and Frustration cause more change in the policies than the random feature baseline so one can view them as fairly useful still.

As a final test, we investigated the utility of each feature by using a different tutor action - whether or not the tutor should give simple feedback (SimFeed), or a complex feedback response(ComFeed), or a combination of the two (Mix) (Tetreault and Litman, 2006). The policies and distributions for all features from this previous work are shown in Ta-

#	State	Baseline 2	+Concept	+Frustration	+ % Correctness
1	certain:C	ComFeed (663)	0: ComFeed (487) R: **SimFeed** (176)	N: ComFeed (558) F: **SimFeed** (105)	H: ComFeed (650) L: ComFeed (13)
2	certain:IPC	ComFeed (251)	0: **Mix** (190) R: **Mix** (61)	N: ComFeed (215) F: ComFeed (36)	H: ComFeed (217) L: ComFeed (34)
3	neutral:C	SimFeed (480)	0: **Mix** (328) R: SimFeed (152)	N: SimFeed (466) F: **ComFeed** (14)	H: SimFeed (468) L: **ComFeed** (12)
4	neutral:IPC	Mix (377)	0: Mix (289) R: Mix (88)	N: Mix (364) F: **ComFeed** (13)	H: Mix (320) L: **ComFeed** (57)
5	uncertain:C	ComFeed (165)	0: ComFeed (127) R: ComFeed (38)	N: ComFeed (151) F: ComFeed (14)	H: **Mix** (156) L: ComFeed (9)
6	uncertain:IPC	ComFeed (244)	0: ComFeed (179) R: ComFeed (65)	N: ComFeed (209) F: ComFeed (35)	H: ComFeed (182) L: ComFeed (62)

Table 8: Feedback Policies (summarized from (Tetreault and Litman, 2006))

bles 7 and 8. Basically, we wanted to see if the relative rankings of the three features remained the same for a different action set and whether different action sets evoked different changes in policy. The result is that although the amount of policy change is much lower than when using Questions as the tutor action, the relative ordering of the features is still about the same with Concept Repetition still having the greatest impact on the policy. Interestingly, while Frustration and Percent Correctness have lower diffs, % policy changes, and E.C.R. then their question counterparts (which indicates that those features are less important when considering what type of feedback to give, as opposed to what type of question to give), the E.C.R. for concept repetition with feedback is actually higher than the question case.

6 Related Work

RL has been applied to improve dialogue systems in past work but very few approaches have looked at which features are important to include in the dialogue state. Paek and Chickering's (2005) work on testing the Markov Assumption for Dialogue Systems showed how the state space can be learned from data along with the policy. One result is that a state space can be constrained by only using features that are relevant to receiving a reward. Henderson et al.'s (2005) work focused on learning the best policy by using a combination of reinforcement and supervised learning techniques but also addressed state features by using linear function approximation to deal with large state spaces. Singh et al. (1999) and Frampton et al. (2005) both showed the effect of adding one discourse feature to the student state (dialogue length and user's last dialogue act, respectively) whereas in our work we compare the worth of multiple features. Although Williams et al.'s (2005b) work did not focus on choosing the best state features, they did show that in a noisy environment, Partially-Observable MDP's could be used to build a better model of what state the user is in, over traditional MDP and hand-crafted methods. One major difference between all this related work and ours is that usually the work is focused on how to best deal with ASR errors. Although this is also important in the tutoring domain, our work is novel because it focuses on more semantically-oriented questions.

7 Discussion

In this paper we showed that incorporating more information into a representation of the student state has an impact on what actions the tutor should take. Specifically, we proposed three metrics to determine the relative weight of the three features. Our empirical results indicate that Concept Repetition and Frustration are the most compelling since adding them to the baseline resulted in major policy changes. Percent Correctness had a negligible effect since it resulted in only minute changes to the baseline policy. In addition, we also showed that the relative ranking of these features generalizes across different action sets.

While these features may appear unique to tutoring systems they also have analogs in other dialogue systems as well. Repeating a concept (whether it be a physics term or travel information) is important because it is an implicit signal that there might be some

confusion and a different action is needed when the concept is repeated. Frustration can come from difficulty of questions or from the more frequent problem for any dialogue system, speech recognition errors, so the manner in dealing with it will always be important. Percent Correctness can be viewed as a specific instance of tracking user performance such as if they are continuously answering questions properly or are confused by what the system requests.

With respect to future work, we are annotating more human-computer dialogue data and will triple the size of our test corpus allowing us to create more complicated states since more states will have been explored, and test out more complex tutor actions, such as when to give Hints and Restatements. In the short term, we are investigating whether other metrics such as entropy and confidence bounds can better indicate the usefulness of a feature. Finally, it should be noted that the certainty and frustration feature scores are based on a manual annotation. We are investigating how well an automated certainty and frustration detection algorithm will impact the % Policy Change. Previous work such as (Liscombe et al., 2005) has shown that certainty can be automatically generated with accuracy as high as 79% in comparable human-human dialogues. In our corpus, we achieve an accuracy of 60% in automatically predicting certainty.

8 Acknowledgments

We would like to thank the ITSPOKE and Pitt NLP groups, Pam Jordan, James Henderson, and the three anonymous reviewers for their comments. Support for this research was provided by NSF grants #0325054 and #0328431.

References

K. Bhatt, M. Evens, and S. Argamon. 2004. Hedged responses and expressions of affect in human/human and human computer tutorial interactions. In *Proc. Cognitive Science*.

K. Forbes-Riley and D. Litman. 2005. Using bigrams to identify relationships between student certainness states and tutor responses in a spoken dialogue corpus. In *SIGDial*.

K. Forbes-Riley, D. Litman, A. Huettner, and A. Ward. 2005. Dialogue-learning correlations in spoken dialogue tutoring. In *Artificial Intelligence in Education*.

M. Frampton and O. Lemon. 2005. Reinforcement learning of dialogue strategies using the user's last dialogue act. In *IJCAI Wkshp. on K&R in Practical Dialogue Systems*.

J. Henderson, O. Lemon, and K. Georgila. 2005. Hybrid reinforcement/supervised learning for dialogue policies from communicator data. In *IJCAI Wkshp. on K&R in Practical Dialogue Systems*.

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

J. Liscombe, J. Hirschberg, and J. Venditti. 2005. Detecting certainness in spoken tutorial dialogues. In *Interspeech*.

D. Litman and S. Silliman. 2004. Itspoke: An intelligent tutoring spoken dialogue system. In *HLT/NAACL*.

T. Paek and D. Chickering. 2005. The markov assumption in spoken dialogue management. In *6th SIGDial Workshop on Discourse and Dialogue*.

S. Singh, M. Kearns, D. Litman, and M. Walker. 1999. Reinforcement learning for spoken dialogue systems. In *Proc. NIPS '99*.

R. Sutton and A. Barto. 1998. *Reinforcement Learning*. The MIT Press.

J. Tetreault and D. Litman. 2006. Using reinforcement learning to build a better model of dialogue state. In *EACL*.

K. VanLehn, P. Jordan, C. Rosé, D. Bhembe, M. Bottner, A. Gaydos, M Makatchev, U. Pappuswamy, M. Ringenberg, A. Roque, S. Siler, R. Srivastava, and R. Wilson. 2002. The archictecture of why2-atlas: A coach for qualitative physics essay writing. In *Intelligent Tutoring Systems*.

M. Walker. 2000. An application of reinforcement learning to dialogue strategy selection in a spoken dialogue system for email. *JAIR*, 12.

J. Williams, P. Poupart, and S. Young. 2005a. Factored partially observable markov decision processes for dialogue management. In *IJCAI Wkshp. on K&R in Practical Dialogue Systems*.

J. Williams, P. Poupart, and S. Young. 2005b. Partially obervable markov decision processes with continuous observations for dialogue management. In *SIGDial*.