Learning Dialogue Agents with Bayesian Relational State Representations

Heriberto Cuayáhuitl
German Research Center for Artificial Intelligence (DFKI)
heriberto.cuayahuitl@dfki.de

Abstract

A new approach is developed for representing the search space of reinforcement learning dialogue agents. This approach represents the state-action space of a reinforcement learning dialogue agent with relational representations for fast learning, and extends it with belief state variables for dialogue control under uncertainty. Our approach is evaluated, using simulation, on a spoken dialogue system for situated indoor wayfinding assistance. Experimental results showed rapid adaptation to an unknown speech recognizer, and more robust operation than without Bayesian-based states.

Introduction

Reinforcement learning dialogue agents have a promising application for adaptive conversational interfaces. Unfortunately, three main problems affect their practical application. The first, *the curse of dimensionality*, causes the state space to grow exponentially in the number of state variables. This problem has been addressed by function approximation techniques (Denecke, Dohsaka, and Nakano 2004; Henderson, Lemon, and Georgila 2005; Chandramohan, Geist, and Pietquin 2010); and by divide-and-conquer approaches (Cuayáhuitl et al. 2010; Lemon 2011). Second, the dialogue agent *operates under uncertainty* (the most obvious source is automatic speech recognition errors, but not the only source). This problem has been addressed by sequential decision-making models under uncertainty (Roy, Pineau, and Thrun 2000; Williams 2006; Thomson 2009; Young et al. 2010). Third, reinforcement learning methods usually require many dialogues to find optimal policies, resulting in *slow learning*. This last problem has been addressed by incorporating prior knowledge into the decision making process (Singh et al. 2002; Heeman 2007; Williams 2008; Cuayáhuitl 2009). Because of such problems, the current practice in dialogue optimization consists in inducing behaviour offline, from a corpus of real dialogues or from simulations. When the learnt policies are then deployed they behave with frozen optimization. The rest of the paper contributes to tackle these problems by proposing a new approach to represent the agent's state-action space.

Dialogue Optimization Under Uncertainty

A human-machine dialogue can be defined as a finite sequence of information units conveyed between conversants, where the information can be described at different levels of communication such as speech signals, words, and dialogue acts. Figure 1 illustrates a model of human-machine interaction. An interaction under uncertainty between both conversants can be briefly described as follows: the machine receives a distorted user speech signal $\tilde{x}_t$ from which it extracts a user dialogue act $\tilde{u}_t$ and enters it into its knowledge base; the machine then updates its belief dialogue state b_t (i.e. a probability distribution over dialogue states) with information extracted from its knowledge base; this dialogue state is received by the spoken dialogue manager in order to choose a machine dialogue act a_t, which is received by the response generation module to generate the corresponding machine speech signal conveyed to the user.

A conversation follows the sequence of interactions above in an iterative process between both conversants until one of them terminates it. Assuming that the machine receives a numerical reward r_t for executing action a_t when the conversational environment makes a transition from belief state b_t to state b_{t+1}, a dialogue can be expressed as $D = \{b_1, a_1, r_2, b_2, a_2, r_3, ..., b_{T-1}, a_{T-1}, r_T, b_T\}$, where T is the final time step. Such sequences can be used by a reinforcement learning agent to optimize the machine's dialogue behaviour. Although human-machine conversations can be used for optimizing dialogue behaviour, a more common practice is to use simulations.

A reinforcement learning dialogue agent aims to learn its behaviour from interaction with an environment, where situations are mapped to actions by maximizing a long-term reward signal (see (Sutton and Barto 1998) for an introduction to reinforcement learning). Briefly, the reinforcement learning paradigm works by using the formalism of Markov Decision Processes (MDPs). An MDP is characterized by a finite set of states S, a finite set of actions A, a probabilistic state transition function, and a reward function that rewards the agent for each selected action. Solving the MDP means finding a mapping from observable states to actions corresponding to $\pi^*(s_t) = \arg\max_{a_t \in A} Q^*(s_t, a_t)$, where the Q-function specifies the cumulative rewards for each state-action pair. The optimal policy can be learnt by dynamic programming or reinforcement learning algorithms.

Proceedings of the 7th IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, pages 9–15,
Barcelona, July 2011

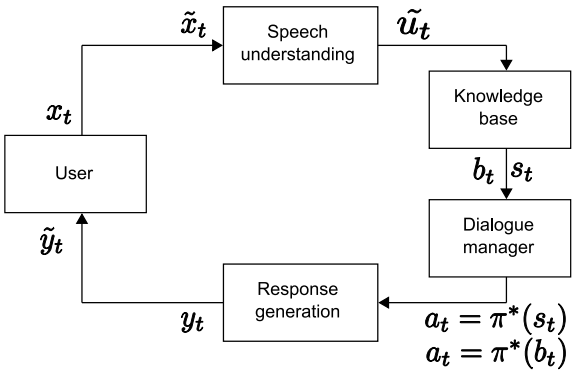

Figure 1: A pipeline model of human-machine interaction, where observable dialogue state s_t or belief dialogue state b_t is used by the dialogue manager to choose action a_t.

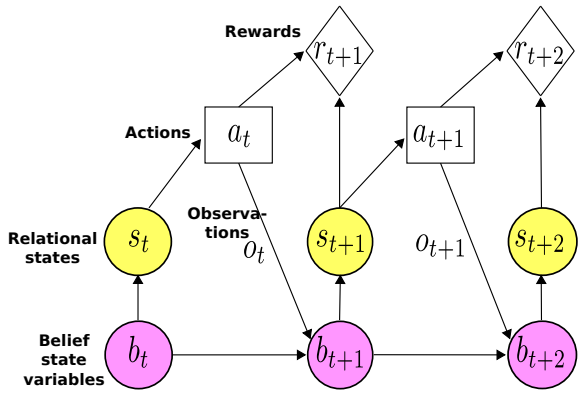

Figure 2: Dynamics of an MDP-based dialogue manager using Bayesian Relational state representations.

An alternative but more computationally intensive model for sequential decision-making under uncertainty is the Partially Observable Markov Decision Process (POMDP) model. In a POMDP the dialogue state is not known with certainty (as opposed to an MDP); i.e. since the agent does not know the state s exactly, it must maintain a belief state over the possible states S (Young et al. 2010). The characterization of a POMDP extends an MDP with a set of observations or perceptions from the environment (e.g. keywords from the user utterances) $\Omega = \{o_1, o_2, ..., o_n\}$, and an observation function $O(s, a, o)$ that specifies a perceived observation o from selecting action a in state s with probability $P(o|s, a)$. Thus, a POMDP can be seen as an MDP over a belief space, where the observable states are replaced by belief states. Solving the POMDP can be described as finding a mapping from belief states to actions corresponding to $\pi^*(b_t) = \arg\max_{a_t \in A} Q^*(b_t, a_t)$, where the Q-function specifies the cumulative rewards for each belief state and action. The rest of the paper describes an approach that extends MDP-based reinforcement learning conversational agents with beliefs states, which can be seen as learning agents with a characterization between MDPs and POMDPs.

A Bayesian-Relational Approach for Dialogue Control Under Uncertainty

Figure 2 shows the presented approach which unifies two concepts: (a) *relational representations* imposed on an MDP state-action space; and (b) *belief state variables* extending the fully-observed state variables by using partition-based Bayesian networks.

Dialogue as a Relational MDP

An MDP is typically represented with propositional representations (e.g. a set of binary features), which result into exponential growth. A relational MDP mitigates that problem by using tree-based and high-level representations resulting in the following benefits: (a) compression and more expressive description of the state-action space, (b) straightforward incorporation of prior-knowledge into the policy, (c) generalization for reusable behaviours, and (d) fast learning.

A relational MDP is a generalization of an MDP specified with representations based on a logical language (van Otterlo 2009). A relational MDP can be defined as a 5-tuple $<S, A, T, R, L>$, where element L is a language that provides the mechanism to express logic-based representations. We describe L as a context-free grammar to represent formulas compounded by predicates, variables, constants and connectives similar to (Russell and Norvig 2003), Chaper 8. Whilst the state set S is generated from an enumeration of all logical forms in grammar L, the actions A available in a given state are constrained by the logical forms in L. A sample relational state is expressed by a set of predicates: '$Salutation(greeting) \wedge Slot(x, confirmed) \wedge SlotsToConfirm(none) \wedge DatabaseTuples(none)$'. This representation indicates that slot x has been confirmed, there are no slots to confirm and no database tuples. A sample relational action is expressed as follows: '$request \leftarrow Salutation(greeting) \wedge Slot(x, unfilled) \wedge SlotsToConfirm(none)$'. This expression indicates that the action 'request' is valid if the logical expression is true.

Relational MDPs with Belief States

Because dialogue states are not known with certainty, POMDPs have been adopted for policy optimization under uncertainty (Roy, Pineau, and Thrun 2000; Williams 2006; Henderson and Lemon 2008; Thomson 2009; Young et al. 2010). Moreover, because POMDPs are computationally intensive and hard to scale up, in this paper we propose to approximate the belief states of a relational MDP with belief state variables. This approximation is used to scale up to more complex conversational systems. The belief states can be defined as $b(s) = \frac{1}{Z} \Pi p(X_i \in s)$, where $p(X_i \in s)$ is the probability distribution of predicate X_i in state s, and Z is a normalization constant.

For the belief states, we maintain a Bayesian Network (BN) for each predicate $X_i \in s$. A BN models a joint probability distribution over a set of random variables and their dependencies based on a directed acyclic graph, where each node represents a variable Y_j with parents $pa(Y_j)$

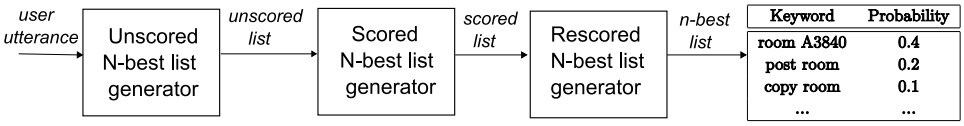

Figure 3: Block diagram for generating N-best list. Whilst scored lists are based on beta distributions and ASR error rates (see Fig. 5), re-scored n-best lists are based on posterior distributions (useful for the belief states) derived from Bayesian Networks.

(Jensen 1996). The Markov condition implies that each variable is only dependent on its parents, resulting in a unique joint probability distribution expressed as $p(Y) = \Pi p(Y_j|pa(Y_j))$, where every variable is associated with a conditional probability distribution $p(Y_j|pa(Y_j))$. Such a network is used for probabilistic reasoning, i.e. the calculation of posterior probabilities given some observed evidence. To that end, we use efficient implementations of the variable elimination and junction tree algorithms (Cozman 2000). In addition, because the size of domain values D for each variable can be large (which results in high computational expense), we use random variables with partitions $D = \{\tilde{D}_i\}$ expressed as

$$D = \begin{cases} \tilde{D}_0 \leftarrow \text{item}_1, \text{item}_2, \text{item}_3 \ldots \text{item}_N, \text{other} \\ \tilde{D}_1 \leftarrow \text{item}_{N+2}, \text{item}_{N+3}, \text{item}_{N+4} \ldots \text{item}_{N'}, \text{other} \\ \ldots \\ \tilde{D}_M \leftarrow \text{item}_{N'+2}, \text{item}_{N'+3} \ldots \text{item}_{N''}, \text{other} \end{cases}$$

where $|\tilde{D}_k| \leq \max$. The entry 'other' is initialized with probability 1, which changes with belief updating during the course of the interaction. At each time step, the networks and corresponding posteriors are updated based on the perceived observations (i.e. ASR N-best lists) from the environment. The N-Best lists were generated according to the procedure shown in Figure 3. Once the posteriors are updated, their 1-best hypotheses are used in the relational states of the MDP.

Belief Updating of the Dialogue State

The partition-based Bayesian Networks (BNs) described above use multiple minimal BNs defined by $p(V_k^i|R_k^i, P_k^i)$, where index i denotes a predicate in the dialogue state and index k denotes a partition in predicate i. The meaning of such random variables is as follows: R_k^i is used for speech recognition at time step t, P_k^i is used for speech recognition at time step $t-1$, and V_k^i is the belief of predicate i. The belief updating procedure is as follows. First, compute an N-best list for each keyword in the user utterance. For each entry in the N-best list, get the partition of the current entry denoted as $\tilde{D}_k^i$. Assign the corresponding probabilities to the random variable R_k^i. Update the probability of entry 'other' according the new observations. If $t = 0$ then assign the probability distribution of R_k^i to P_k^i, else assign the probability distribution of V_k^i to P_k^i so that it can maintain the previous beliefs. Finally, the state with the highest probability in the random variables $V_{\forall k}^i$—computed by combining partitions omitting the entry 'other' and redistributing probability mass accordingly—is used in predicate X_i of dialogue state s. This implies that there is a single belief for each predicate, even if it appears in multiple dialogue states.

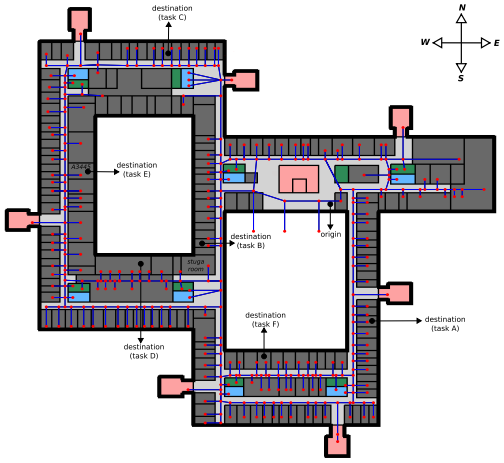

Figure 4: Map of the navigation environment including a superimposed route graph specifying the navigational space. The black circles represent origin and destination locations.

Experiments and Results

We tested our approach in a learning agent that collects information for situated indoor navigation using simulated speech-based interactions. The task of the user is to navigate from an origin to a destination based on instructions received from a dialogue system. After each instruction the user has to say where he/she is and the agent has to guide the user to the goal location (see Figure 4, and (Cuayáhuitl and Dethlefs 2011a) for a dialogue system of this type but without belief monitoring). This scenario represents at least the following sources of uncertainty: What did the user say? Where is the user? What does the user know? This paper focuses its attention on the first source of uncertainty.

The Simulated Conversational Environment

The system and user verbal contributions are based on the Dialogue Act (DA) types shown in Table 1 combined with the attributes {origin, destination}. This makes a set of 10 user DAs and 14 system DAs. We used the conditional probability distribution $p(u|a)$ for simulating user dialogue acts u given the last machine dialogue acts a. The user responses were coherent with probability 0.9 and random otherwise, a speech recognition error rate of 20% was simulated and ambiguity of domain values of 10%.

In addition, we modelled Automatic Speech Recognition (ASR) events from *beta* continuous probability distri-

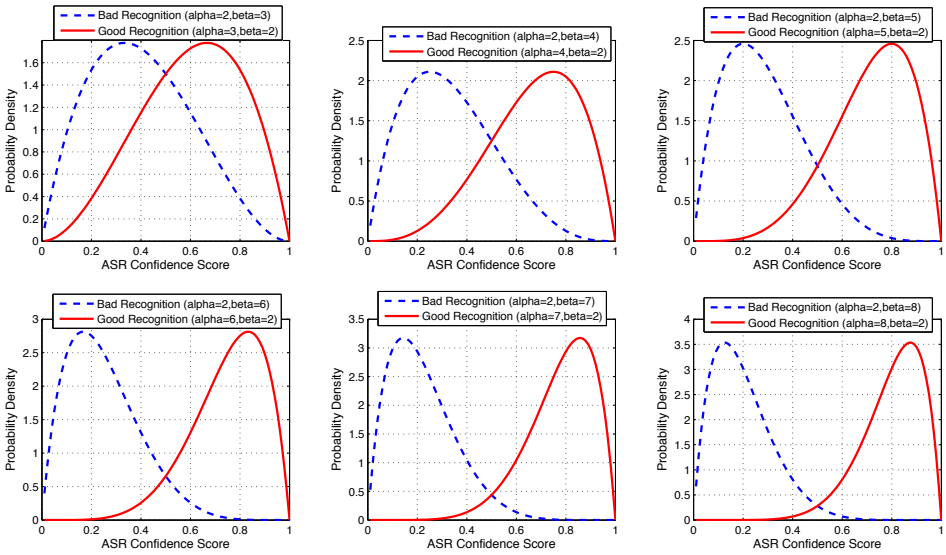

Figure 5: Beta probability distributions for modelling speech recognition events in simulation-based dialogue strategy learning.

butions (see Figure 5), which have been applied to statistical dialogue modelling by (Williams and Balakrishnan 2009; Williams 2010). The *beta* distribution is defined in the interval (0, 1) and it is parameterized by two positive shape parameters referred to as α and β. The probability density function of a *beta* distribution is expressed as

$$f(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\int_0^1 x^{\alpha-1}(1-x)^{\beta-1}\,dx},$$

where the denominator represents the beta function, α and β are positive real numbers (which can be estimated from data), and $0 \leq x \leq 1$. Our simulations used (α=2,β=5; α=5,β=2) for bad and good recognition, respectively.

Characterization of the Learning Agent

Figure 6 shows the context-free grammar specifying the language for the relational states in our learning agent. Whilst the enumeration using a propositional representation represents a total of $100^2 \times 3^3 = 270$ thousand states (100^2 recognized locations for each confidence score from 0.01 to 1.0; 3 values for unfilled, filled, confirmed origin; 3 values for unfilled, filled, confirmed destination; and 3 values for ambiguous user dialogue act), the relational representation only required 21 thousand combinations (7.7% of the propositional representation). The actions constrained with the relational states (i.e. logical forms in grammar L) are expressed as

$$A = \begin{cases} \text{Request(origin,destination)} \leftarrow l_{01} \\ \text{Request(origin)} \leftarrow l_{03} \vee l_{12} \\ \text{Request(destination)} \leftarrow l_{02} \vee l_{08} \\ \text{Apology(origin,destination)+} \\ \qquad \text{Request(origin,destination)} \leftarrow l_{04} \\ \text{Apology(origin)+Request(destination)} \leftarrow l_{02} \vee l_{11} \\ \text{Apology(destination)+Request(origin)} \leftarrow l_{03} \vee l_{09} \\ \text{ImpConf(origin)+Request(destination)} \leftarrow l_{02} \\ \text{ImpConf(destination)+Request(origin)} \leftarrow l_{03} \\ \text{ExpConf(origin)} \leftarrow l_{02} \vee l_{11} \\ \text{ExpConf(destination)} \leftarrow l_{03} \vee l_{09} \\ \text{ExpConf(origin,destination)} \leftarrow l_{04} \\ \text{Clarify(origin)} \leftarrow l_{05} \vee l_{13} \\ \text{Clarify(destination)} \leftarrow l_{04} \vee l_{10} \\ \text{Clarify(origin,destination)} \leftarrow l_{08}. \end{cases}$$

It can be observed that whilst the propositional state-action space would use $100^2 \times 3^3 \times 14$ =3.8 million state-actions, the constrained state-action space only uses 32 thousand (less than 1% of the propositional one). The goal state is defined when the origin and destination locations are confirmed (a sample dialogue is shown in Table 2). In addition, the Bayesian networks (with semi-hand-crafted structure and parameters based on the spatial environment) for modelling the beliefs of predicates in the relational states are shown in Figure 7. Since the posteriors can have a large number of probabilities (e.g. the conditional probability table for predicate 'UserOrigin' has $200^3 \times 2$=16 million entries), we partitioned large networks with entries based on locations per navigation segment (from one junction to another) allowing a maximum of domain values $\max \leq 30$ (i.e. multiple instantiations of a Bayesian net with smaller conditional probability tables). Finally, the reward function is defined by the following rewards: 0 for reaching the goal

$L := l_1\ l_2\ l_3\ l_4\ l_5\ l_6\ l_7\ l_8\ l_9\ l_{10}\ l_{11}\ l_{12}\ l_{13}\ l_{14}$
l_1:= UserOrigin(unfilled) ∧ UserDestination(unfilled) ∧ AmbiguousUserDialogueAct(unknown)
l_2:= UserOrigin(filled,score) ∧ UserDestination(unfilled) ∧ AmbiguousUserDialogueAct(no)
l_3:= UserOrigin(unfilled) ∧ UserDestination(filled,score) ∧ AmbiguousUserDialogueAct(no)
l_4:= UserOrigin(filled,score) ∧ UserDestination(filled,score) ∧ AmbiguousUserDialogueAct(no)
l_5:= UserOrigin(filled,score) ∧ UserDestination(unfilled) ∧ AmbiguousUserDialogueAct(yes)
l_6:= UserOrigin(unfilled) ∧ UserDestination(filled,score) ∧ AmbiguousUserDialogueAct(yes)
l_7:= UserOrigin(filled,score) ∧ UserDestination(filled,score) ∧ AmbiguousUserDialogueAct(yes)
l_8:= UserOrigin(confirmed) ∧ UserDestination(unfilled) ∧ AmbiguousUserDialogueAct(no)
l_9:= UserOrigin(confirmed) ∧ UserDestination(filled,score) ∧ AmbiguousUserDialogueAct(no)
l_{10}:= UserOrigin(confirmed) ∧ UserDestination(filled,score) ∧ AmbiguousUserDialogueAct(yes)
l_{11}:= UserOrigin(filled,score) ∧ UserDestination(confirmed) ∧ AmbiguousUserDialogueAct(no)
l_{12}:= UserOrigin(unfilled) ∧ UserDestination(confirmed) ∧ AmbiguousUserDialogueAct(no)
l_{13}:= UserOrigin(filled,score) ∧ UserDestination(confirmed) ∧ AmbiguousUserDialogueAct(yes)
l_{14}:= UserOrigin(confirmed) ∧ UserDestination(confirmed) ∧ AmbiguousUserDialogueAct(no)
score := $0.01\ \vee\ 0.02\ \vee\ 0.03\ \vee 0.04\ \vee 0.05\ \vee ...\ \vee\ 0.97\ \vee\ 0.98\ \vee\ 0.99 \vee 1$

Figure 6: Context-free grammar defining the language L for collecting information in the wayfinding domain. See (Cuayáhuitl and Dethlefs 2011b) for a more complete state representation of the wayfinding interaction (including information presentation).

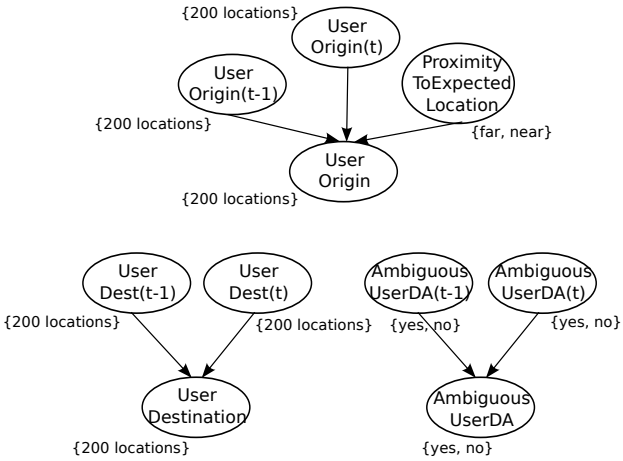

Figure 7: Bayesian networks for modelling the beliefs of predicates in the relational states. The domain values of each random variable is shown in curly brackets. Notice that the top and bottom left networks use multiple networks (partitions) to handle smaller conditional probability tables.

state and -10 otherwise. We used the Q-Learning algorithm (Sutton and Barto 1998). The learning rate parameter α decays from 1 to 0 according to $\alpha = 100/(100 + \tau)$, where τ represents elapsed time-steps. The action selection strategy used ϵ-Greedy with $\epsilon = .01$, undiscounted rewards, and Q-values initialized to 0.

Experimental Results

Figure 8 shows the learning curves of induced behaviour with the proposed approach. One thing to notice is that reinforcement learning with relational representations using a constrained state-action space is dramatically faster than without constraints. Whilst the latter requires five orders of magnitude to learn a stable policy, the former only requires three orders of magnitude. Two key characteristics of relational state-action spaces are (1) they are easy to specify and to read, and (2) they offer the mechanism to generate coherent dialogues (even before learning). Surprisingly, the relational representations have been ignored in the learning dialogue systems field. Another thing to notice is that learnt policies with belief state variables help to improve performance more (due to more accurate recognitions) than without tracking beliefs from the environment. We measured the average system turns of the last 1000 training dialogues and found that constrained learning with belief states outperforms its counterpart (constrained learning without belief states) by an absolute 15% in terms of average system turns. We also compared the average system turns of the first 1000 training dialogues and the last 1000 training dialogues for the best policy (with beliefs), and found that the latter phase outperformed the first one by 2 system turns. This indicates that the hand-coded policy with relational representations was improved by policy learning.

Furthermore, our approach scales up to larger domain values because (a) the size of the relational state-action space is location-independent, and (b) even when the Bayesian Networks (BNs) are slot-dependent, the partitioned approach makes them scalable. It remains to be investigated the scalability limits of our approach with larger and more complex BNs. Nonetheless, the partition-based BNs reduce computational demands for loading, updating and querying beliefs in comparison to non-partitioned BNs. Although the results above require an evaluation in a realistic environment, the proposed approach is promising for optimizing dialogue behaviour in unknown and uncertain environments (which require fast learning with continuous belief tracking).

Conclusions and Future Work

We have described a unified approach for representing search spaces of reinforcement learning dialogue agents, which aims for efficient and robust operation combined with straightforward design. To this end we use logic-based representations in the state-action space, and extend them with belief states by using partition-based Bayesian networks.

Dialogue Acts	Sample Utterance
Provide(ori)	I am in front of room B3090
Provide(des)	How do I get to Dr. Watson's office?
Provide(ori,des)	How do I get from room B3090 to Dr. Watson's office?
Reprovide(ori)	I said in front of room B3090
Reprovide(des)	I meant to Dr. Watson's office?
Reprovide(ori,des)	I asked how do I get from room B3090 to Dr. Watson's office?
Confirm(ori)	Yes, I did.
Confirm(des)	Yes, I said that.
Confirm(ori,des)	Yes, please.
Silence()	[*remain in silence*]
Request(ori,des)	What is your origin and destination?
Request(ori)	Where are you?
Request(des)	Where would you like to go?
Apology(ori,des)+ Request(ori,des)	Sorry, from where to where?
Apology(ori)+ Request(ori)	Sorry, where are you?
Apology(des)+ Request(des)	Sorry, what is your destination?
ImpConf(ori)+ Request(des)	Okay, from room B3090, to where ?
ImpConf(des)+ Request(ori)	Okay, to room B3090, where are you?
ExpConf(ori,des)	Yes
ExpConf(ori)	No
ExpConf(des)	Yes I did
Clarify(ori)	Do you mean James Watson or Peter Watson?
Clarify(des)	Do you mean Copy room or Post room?
Clarify(ori,des)	Do you want to go to the Copy room or Post room?

Table 1: Dialogue Acts for collecting information in the situated navigation domain, where ori=origin and des=destination. The groups correspond to user and system dialogue acts, respectively.

Our experimental results provide evidence to conclude that our method is promising because it combines fast learning with robust operation. By proposing relational state-action spaces, it makes a concrete contribution to conversational interfaces which learn their dialogue behaviour. Although this approach scales up to large domain values, it can be extended with hierarchical control to deal with large relational states and optimization of large-scale conversational interfaces; e.g, hierarchical reinforcement learning dialogue agents such as (Cuayáhuitl et al. 2010; Cuayáhuitl and Dethlefs 2011a) can be extended with Bayesian relational representations.

Related work closest to ours is the following. (Lecoeuche 2001) used reinforcement learning with relational representations, but he did not model beliefs. (Horvitz and Paek 1999; Paek and Horvitz 2000; Bohus and Rudnicky 2005; 2006; Skantze 2007) modelled beliefs in dialogue systems, but they did not optimize conversations using reinforcement learning. In general, our approach lies between the

Agent	Dialogue Act	Utterance
Sys	Request(ori,des)	What is your origin and destination?
Usr	Provide(ori,des)	*I want to go from room B3090 to Dr. Watson's office?*
Sys	ImpConf(ori) +Request(des)	Okay, from room B3090, to where?
Usr	Reprovide(des)	*Dr. Watson's office?*
Sys	Clarify(des)	Do you mean James Watson or Peter Watson?
Usr	Reprovide(des)	*Peter Watson*
Sys		[provides a route instruction]
Usr	Provide(ori)	[executes the route instruction] I am in front of the lifts
Sys	Apology(ori) +Request(ori)	Sorry, where are you?
Usr	Reprovide(ori)	*In the corridor of the lifts*
Sys	Clarify(des)	Do you mean the lifts next to the language learning center?
Usr	Confirm(ori)	*Ehhh, yes*
Sys		[provides a route instruction]
Usr	Provide(ori)	[executes the route instruction] *Okay, now I can see offices B3280 and B3285*
Sys		[provides a route instruction]
...		[and so on until reaching the goal]

Table 2: Fragment of a conversation in the situated wayfinding domain. This dialogue focuses its attention on collecting information as the user carries out the navigation task. We assume that the user carries a hand-held device with him/her to communicate with the system using spoken interaction.

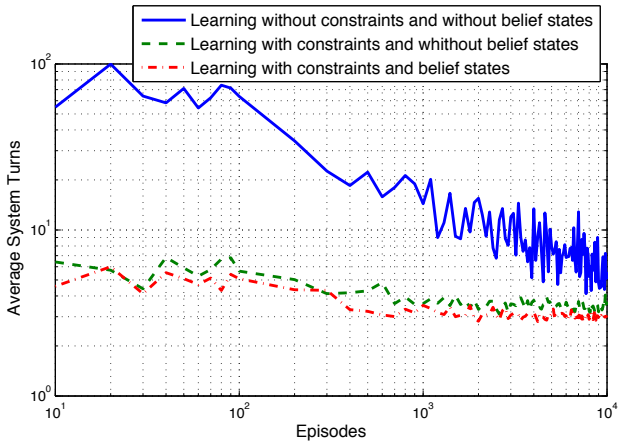

Figure 8: Learning curves of induced dialogue behaviour (averaged over 10 runs), where all agents started to learn after 100 episodes (to plot the performance before learning). Learning with constraints and belief states (i.e. the best learning curve) outperforms its counterpart (learning with constraints and without belief states) by an absolute 15% in terms of average system turns due to more accurate speech recognition. The best learnt dialogue policy improved the hand-coded constraints from 5 to 3 system turns, derived from a comparison of the first and the last 1000 dialogues.

MDP and POMDP models (Roy, Pineau, and Thrun 2000; Williams 2006; Thomson 2009; Young et al. 2010). Since we model beliefs of predicates (with short histories) in the dialogue state instead of beliefs of entire dialogue states (with long histories), our approach is expected to be less robust than the POMDP model but at the same time more scalable. A theoretical and experimental comparison between our and a POMDP-based approach is left as future work. Another future direction is to use (non-)linear function approximation for tackling very large relational state-action-spaces, when hierarchical control would not be sufficient to control the rapid state space growth. Finally, the proposed approach can be assessed in larger, more complex systems.

References

Bohus, D., and Rudnicky, A. 2005. Constructing accurate beliefs in spoken dialogue systems. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 272–277.

Bohus, D., and Rudnicky, A. 2006. A 'K hypothesis + other' belief updating model. In *AAAI Workshop on Statistical and Empirical Approaches for Spoken Dialogue Systems*, 13–18.

Chandramohan, S.; Geist, M.; and Pietquin, O. 2010. Optimizing spoken dialogue management from data corpora with fitted value iteration. In *INTERSPEECH*, 86–89.

Cozman, F. G. 2000. Generalizing variable elimination in Bayesian networks. In *IBERAMIA/SBIA, Workshop on Probabilistic Reasoning in Artificial Intelligence*, 27–32.

Cuayáhuitl, H., and Dethlefs, N. 2011a. Spatially-aware dialogue control using hierarchical rienforcement learning. *ACM Transactions on Speech and Language Processing (Special Issue on Machine Learning for Robust and Adaptive Spoken Dialogue Systems)* 7(3):5:1–5:26.

Cuayáhuitl, H., and Dethlefs, N. 2011b. Optimizing situated dialogue management in unknown environments. In *INTERSPEECH*.

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

Cuayáhuitl, H. 2009. *Hierarchical Reinforcement Learning for Spoken Dialogue Systems*. Ph.D. Dissertation, School of Informatics, University of Edinburgh.

Denecke, M.; Dohsaka, K.; and Nakano, M. 2004. Fast reinforcement learning of dialogue policies using stable function approximation. In *International Joint Conference on Natural Language Processing (IJCNLP)*, 1–11.

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

Henderson, J., and Lemon, O. 2008. Mixture model POMDPs for efficient handling of uncertainty in dialogue management. In *International Conference on Computational Linguistics (ACL)*, 73–76.

Henderson, J.; Lemon, O.; and Georgila, K. 2005. Hybrid reinforcement/supervised learning for dialogue policies from communicator data. In *Workshop on Knowledge and Reasoning in Practical Dialogue Systems (IJCAI)*, 68–75.

Horvitz, E., and Paek, T. 1999. A computational architecture for conversation. In *International Conference on User Modelling (UM)*, 201–210.

Jensen, F. 1996. *An Introduction to Bayesian Networks*. Springer Verlag, New York.

Lecoeuche, R. 2001. Learning optimal dialogue management rules by using reinforcement learning and inductive logic programming. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

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*.

Paek, T., and Horvitz, E. 2000. Conversation and action under uncertainty. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 455–464.

Roy, N.; Pineau, J.; and Thrun, S. 2000. Spoken dialogue management using probabilistic reasoning. In *International Conference on Computational Linguistics (ACL)*, 93–100.

Russell, S., and Norvig, P. 2003. *Artificial Intelligence: A Modern Approach*. Pearson Education.

Singh, S.; Litman, D.; Kearns, M.; and Walker, M. 2002. Optimizing dialogue management with reinforcement learning: Experiments with the NJFun system. *Journal of AI Research* 16:105–133.

Skantze, G. 2007. *Error Handling in Spoken Dialogue Systems: Managing Uncertainty, Grounding and Miscommunication*. Ph.D. Dissertation, KTH - Royal Institute of Technology.

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

Thomson, B. 2009. *Statistical methods for spoken dialogue management*. Ph.D. Dissertation, University of Cambridge.

van Otterlo, M. 2009. *The Logic of Adaptive Behaviour: Knowledge Representation and Algorithms for Adaptive Sequential Decision Making under Uncertainty in First-Order and Relational Domains*. IOS Press.

Williams, J., and Balakrishnan, S. 2009. Estimating probability of correctness for ASR N-Best lists. In *Workshop on Discourse and Dialogue (SIGDIAL)*.

Williams, J. 2006. *Partially Observable Markov Decision Processes for Spoken Dialogue Management*. Ph.D. Dissertation, Cambridge University.

Williams, J. 2008. Integrating expert knowledge into POMDP optimization for spoken dialogue systems. In *AAAI Workshop on Advancements in POMDP Solvers*.

Williams, J. 2010. Incremental partition recombination for efficient tracking of multiple dialog states. In *ICASSP*.

Young, Y.; Gasic, M.; Keizer, S.; Mairesse, F.; Schatzmann, J.; B., T.; and Yu, K. 2010. The hidden information state model: a practical framework for POMDP-based spoken dialogue management. *Computer Speech and Language* 24(2):150–174.