POMDP CONCEPT POLICIES AND TASK STRUCTURES FOR HYBRID DIALOG MANAGEMENT

Sebastian Varges, Giuseppe Riccardi, Silvia Quarteroni, Alexei V. Ivanov

Department of Information Engineering and Computer Science
University of Trento, 38050 Povo di Trento, Italy
{varges|riccardi|silviaq|ivanov}@disi.unitn.it

ABSTRACT

We address several challenges for applying statistical dialog managers based on Partially Observable Markov Models to real world problems: to deal with large numbers of concepts, we use individual POMDP policies for each concept. To control the use of the concept policies, the dialog manager uses explicit task structures. The POMDP policies model the confusability of concepts at the value level. In contrast to previous work, we use explicit confusability statistics including confidence scores based on real world data in the POMDP models. Since data sparseness becomes a key issue for estimating these probabilities, we introduce a form of smoothing the observation probabilities that maintains the overall concept error rate. We evaluated three POMDP-based dialog systems and a rule-based one in a phone-based user evaluation in a tourist domain. The results show that a POMDP that uses confidence scores, in combination with an improved SLU module, achieves the highest concept precision.

Index Terms— dialog management, POMDP, reinforcement learning, task structure

1. INTRODUCTION

In probabilistic dialog systems, the interaction between a system and its environment (which includes the user) is modeled similarly to probabilistic robotics. The interaction is characterized as a dynamic system which manipulates its environment by performing dialog actions and perceives feedback from the environment through its sensors. The original sensory information is obtained from the speech recognition (ASR) results. These are typically processed by a spoken language understanding module (SLU) before being passed on to the dialog manager.

The seminal work of [1] modeled dialog management as a Markov Decision Process (MDP). Using reinforcement learning as the general learning paradigm, an MDP-based dialog manager incrementally acquires a policy by obtaining rewards about actions it performed in specific dialog states. However, MDPs do not take into account the observational uncertainty of the speech recognition results, a key challenge in spoken dialog systems. Partially Observable Markov Decision Process (POMDPs) address this issue by explicitly modeling how the distribution of observations is governed by states and actions [2, 3, 4]. On the one hand, POMDPs provide a principled treatment of decision making under uncertainty. On the other hand, computational complexity becomes an issue when applying POMDPs to real world scenarios.

In this work, we implement and evaluate a hybrid divide-and-conquer approach to dialog management by optimizing policies for acquiring individual concepts separately. This makes optimization much easier and allows us to model the confusability of concrete concept *values* explicitly without having to abstract them away. The use of individual policies is orchestrated by an explicit task structure that activates and de-activates them. The resulting division of labor between learned knowledge and explicitly encoded knowledge is used to automatically optimize action decisions that are hard to encode manually, in particular handling uncertainty via clarification questions, but leave the control of well-known task knowledge in the hand of the developer. To address data sparseness issues that arise when dealing with concrete values, we introduce a novel form of smoothing the observation counts that maintains the overall concept error rate.The approach has been evaluated in several user studies in a tourist information domain involving 20 subjects and four different dialog systems.

2. RELATED WORK

In the well-known Hidden Information State approach to dialog management [2] the summary space contains the probabilities of the top two hypotheses and information about the user act and supporting database entries. By dealing with individual values, the presented approach pursues a very different strategy for dealing with the tractability challenges of applying POMDPs to real world dialog management. This makes our approach more related to [4] who use individual concept policies, albeit with Dynamic Decision Networks. The use of confidence measures in POMDP frameworks has been proposed in [3]. However, a difference to all three above-

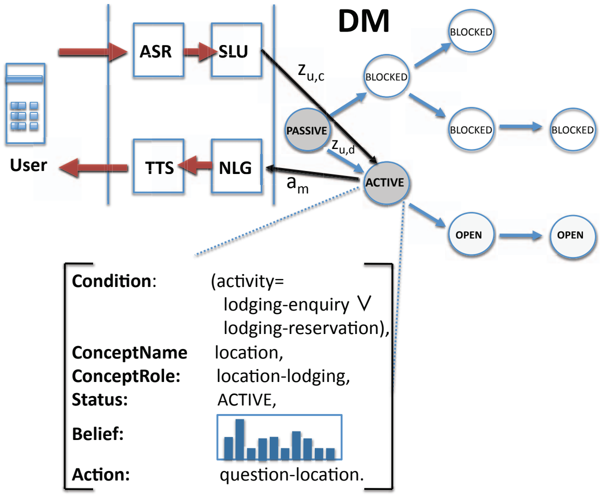

Figure 1: System architecture with Task Structure (active task node in detailed view)

mentioned approaches seems to be our use of real data and users. To our knowledge error-rate adjusted smoothing has not been employed in dialog modeling before. Moreover, although task structures have been used for dialog management [5], they are not used to control POMDP policies.

3. TASK STRUCTURE AND DIALOG MANAGEMENT

Our domain is a tourist information system that deploys 5 different policies that can be used in 8 different task roles (see below). The dialog manager uses a task structure to determine which POMDPs are active. The task structure is essentially a directed AND-OR graph with a common root node. The dialog manager maintains a separate belief distribution for each concept. Each node in the task structure consists of a belief representation, a status variable (ACTIVE|PASSIVE|OPEN|BLOCKED), the concept class (e.g. month) and its 'role' (e.g. check-in month vs check-out month), a condition for entering the node in the first place, plus some book keeping variables.

Figure 1 shows the general system architecture with a schematic view of the task structure, and additionally a more detailed view of an active location node. In the example, the root node has already finished and the system is currently obtaining the location for a lodging task.

At the beginning of a dialog, the task structure is initialized by activating the root node. A top level function activates nodes of the task structure and passes control to that node. Each node maintains a belief b_c for a concept c, which is used to rank the available actions by computing the inner product of policy vectors and belief. The top-ranked action a_m is selected by the system, i.e. it is exploiting the policy, and passed to the natural language generator (NLG). Next, the top-ranked SLU results for the active node and concept are used as observation $z_{u,c}$ to update the belief to b'_c:

$$b'_c(s') = \sum_{s \in S} b_c(s)\, T(s, a_m, s')\, O(a, s', z_{u,c}) / p_{z_{u,c}} \qquad (1)$$

where probability $b'_c(s')$ is the updated belief of being in state s', which is computed as the sum of the probabilities of transitioning from all previous belief points s to s' by taking machine action a_m ($T(s, a_m, s')$) and observing $z_{u,c}$ with probability $O(a_m, s', z_{u,c})$. Normalization to obtain a valid probability distribution is performed by dividing by the probability of the observation $p_{z_{u,c}}$. Probability $O(a_m, s', z_{u,c})$ is smoothed (section 4). We note a mismatch between turn taking in the dialog system and performing updates according to equation (1): each turn starts with a recognition phase, followed by action selection of the DM. Therefore, belief updating takes place after speech recognition/SLU but uses the action of the *previous* turn, $a_{m,t-1}$.

A concept remains active until a submit action is selected. At that point, the next active node is retrieved from the task structure and immediately used for action selection with an initially uniform belief. Submit actions are not communicated to the user but collected and used for the database query at the end of the dialog.

Overanswering, i.e. the user providing more information than directly asked for, is handled by *delayed belief updating*: the SLU results are stored until the first concept of a matching type becomes active. This is a heuristic rule designed to ensure that a concept is interpreted in its correct role. Operationally, unused SLU results $z_{u,d}$ (where concept $d \neq c$) are passed on to the next activated task node (see also figure 1).

4. CONCEPT-LEVEL POMDPS

The system has a number of actions to obtain further information from the environment which it can try and repeat in any order. Ultimately, however, it needs to commit to a decision, which in our case means 'submitting' a specific concept value in a database. A POMDP is generally characterized as the tuple (S, A, T, R, O, Z). In our approach, the state set S consists of the concept values, for example location names (trento). The action set A consists of a concept question, a set of clarification actions, and a set of submit actions (question-location, verify-trento, submit-trento etc). The set of observation symbols Z consist of the concept values and additionally yes and no. When using SLU confidences in the POMDP, the observations represent two confidence bins (trento_HIGH/trento_LOW). The bins are defined by the median confidences in a previous data collection. Transitions T do not change the state (see above) and are thus described as 'identity'. Rewards R assign a small cost to all question and clarification actions, a large cost to submitting the wrong value (e.g. choosing action submit-trento in state maderno) and large reward to submitting the correct value.

The most important aspect of this model is the observation probability matrix O. It is based on empirical data about the actual confusability of concept values, which we obtained from a previous data collection experiment [6]. However, there are serious data sparseness issues: many values are only rarely mentioned or not at all. A first idea might be to use simple add-one smoothing, i.e. to assume that each observation is made at least once. However, this implies unrealistic recognition error rates. Our idea for smoothing is to 'match' the average error rate for the concept: *"class error rate adjusted add-one smoothing"*. The smoothing algorithm proceeds in three steps:

1. compute class error rate E_c of concept c,
2. add one to all incorrect observation counts m_z,
3. take the probability of the correct observation to be 1-E_c and scale all other probabilities to fit the remaining probability mass E_c by computing $\frac{m_z+1}{\sum_{i \in Z}^{n} m_{z_i}+1} E_c$.

We optimized the policies individually with the `APPL` solver that employs the `SARSOP` sampling-based POMDP algorithm [7].

5. EXPERIMENTS

We conducted experimental studies involving four spoken dialog systems:

1. RULE: a rule-based dialog manager [6],
2. POMDP-Ia a POMDP based DM with a simple SLU component,
3. POMDP-Ib: a POMDP based DM with an advanced SLU component,
4. POMDP-IIb: a POMDP based DM that uses SLU confidences with an advanced SLU component.

The simple SLU module performs a turn-level parse of the user's utterance, regardless of the current dialog context. The ASR interpretation for the opening prompt, obtained via a language model, is parsed by a Finite State Transducer to obtain the user's intended task and related concepts such as locations and event types. The other prompts are addressed via concept-specific grammars.

The advanced SLU module produces a full interpretation of the user's utterance in the context of the current task and conversation progress (e.g. using previous turn history to identify that the month mentioned at the current turn is the checkout month of a lodging enquiry). Moreover, it performs a more precise interpretation of the opening prompt utterance thanks to regular expressions on the language model output, and a more careful concept confidence estimation.

	Lodging Task		Event Enquiry		ALL
	TCR	#turns	TCR	#turns	TCR
RULE	70.3% (26/37)	13.0 (σ=3.5)	66.7% (28/42)	8.7 (σ=2.5)	68.4% (54/79)
POMDP-Ia	79.0% (45/57)	22.0 (σ=5.8)	84.3% (27/32)	14.4 (σ=4.3)	80.9% (72/89)
POMDP-Ib	91.4% (74/81)	19.8 (σ=4.1)	94.2% (65/69)	13.3 (σ=2.9)	92.7% (139/150)
POMDP-IIb	88.8% (87/98)	21.7 (σ=5.5)	86.5% (45/52)	14.0 (σ=5.2)	88.0% (132/150)

Table 1: Task completion and length metrics

All systems use the same Voice XML platform to drive ASR and TTS components as described in [8]. Speech recognition is based on statistical language models for the opening prompt, and is grammar-based otherwise.

For all data collections we used the same tasks and drew subjects from a fixed pool users (40 overall). The dialog systems were evaluated in two batches: RULE vs POMDP-Ia in the first batch and POMDP-Ib vs POMDP-IIb in the second. The POMDP-Ia and POMDP-Ib used confusion data of an earlier data collection with RULE [6]; POMDP-IIb (confidences) used additionally data obtained from RULE of the first batch.

Table 1 shows task completion rates ('TCR', with actual call counts in brackets) and durations ('#turns') for the POMDP and rule-based systems. Task completion in this metric is defined as the number of tasks of a certain type that were successfully concluded. Duration is measured in the number of turn pairs consisting of a system action followed by a user action. Table 1 shows that the POMDP-DMs successfully conclude more and longer lodging and event tasks.

In general, the POMDP policies can be described as more cautious than the rule-based system although obviously the dialog length of the rule system depends on the (heuristically set) thresholds. In order to measure the effect of the dialog strategies in both systems, we computed concept precisions for two different mentions of the concept in the dialog (table 2): first mentions and final values after clarifications and similar strategies. The latter is the concept value that the system ultimately obtained from the user and used in the database query. Row 'ALL' in table 2 refers to the average weighted precision of the four concepts. Table 2 also shows the standard deviation of these numbers, obtained by subsampling from the distribution (25 random samples of the initial dialog data, each $\frac{1}{2}$ the data size).

As table 2 clearly shows, the use of clarification strategies has a positive effect on concept precision in all four systems. The overall final precision increases from RULE to POMDP-IIb. The relative improvement of POMDP-Ib is largest (13.3%), presumably because it starts from a lower first mention precision than POMDP-IIb. The average first mention precisions, which in general should be similar for

	RULE			POMDP-Ia			POMDP-Ib			POMDP-IIb		
	first	final	$\delta\%$	first	final	$\delta\%$	first	final	$\delta\%$	first	final	$\delta\%$
activity	78	74	-4.1	83	88	5.0	83	96	15.7	84*	84*	0.0
loction	64	74	15.8	69	73	6.3	54	69	28.0	66	76	14.3
starrating	67	70	3.4	90	97	7.7	87	96	10.0	94	96	2.6
month	85	89	4.3	76	86	12.7	76	83	9.0	92	93	1.6
day	70	76	8.3	61	76	25.3	74	82	10.0	76	90	18.3
ALL	**74**	**78**	**5.2**	**74**	**83**	**12.1**	**74**	**84**	**13.3**	**82**	**88**	**7.4**
σ=	(0.02)	(0.02)	(2.11)	(0.03)	(0.03)	(2.19)	(0.03)	(0.03)	(2.50)	(0.02)	(0.02)	(2.09)

Table 2: Improvement of concept precision due to dialog strategies: first vs final value, relative change

RULE and II, and for POMDP-Ib and IV due to their shared SLU components, are different in the latter case. We found a different distribution over values chosen by the users to be the likely explanation. The precision of concept activity requires two comments. First, in RULE it decreases: the dialog strategy of the system is to reprompt rather than verify, and the second value obtained may be incorrect but above the confidence threshold – after all, the rule system does not maintain a belief distribution over values. Secondly, the precision of the activity values of POMDP-IIb (marked '*') is not directly comparable because of the learned dialog strategy: the user only needs to confirm/disconfirm and does not need to name the activity explicitly.

We conducted two-sample statistical significance tests by computing the delta in the form of three values for individual data points, i.e. dialogs, and assigned `+1` for all changes from non-match to match, `-1` for a change in the opposite direction and `0` for everything else (e.g. from mismatch to mismatch). We found that the change from RULE and POMDP-Ib to POMDP-IIb is statistically significant ($p<0.05$) whereas the change from POMDP-Ia to POMDP-IIb is not (p=0.115).

6. DISCUSSION AND CONCLUSIONS

The results show that a POMDP that uses confidence scores, in combination with an improved SLU module, achieves the highest concept precision. The flipside of using POMDP-based DMs is the greater dialog duration that the larger number of actions entails. There is a clear trade-off between task completion and dialog length. In future work, we would like to introduce a length penalty into the reward function of the POMDPs (which, however, would not optimize globally). Furthermore, we would like to model the activation of the POMDPs with a separate meta-POMDP. However our current systems are largely using system initiative so that this does not yet appear to be a major problem.

7. ACKNOWLEDGEMENTS

This work was partially supported by the European Commission Marie Curie Excellence Grant for the ADAMACH project (contract No. 022593).

8. REFERENCES

[1] E. Levin, R. Pieraccini, and W. Eckert, "A stochastic model of human-machine interaction for learning dialog strategies," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 1, 2000.

[2] S. Young, M. Gasic, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu, "The Hidden Information State Model: a practical framework for POMDP-based spoken dialogue management," *Computer Speech and Language*, vol. 24, pp. 150–174, 2010.

[3] Jason D. Williams, Pascal Poupart, and Steve Young, "Partially Observable Markov Decision Processes with Continuous Observations for Dialogue Management," in *Proc. 6th SIGdial Workshop on Discourse and Dialogue (SIGDIAL)*, 2005.

[4] T.H. Bui, M. Poel, A. Nijholt, and J. Zwiers, "A tractable hybrid DDN-POMDP approach to affective dialogue modeling for probabilistic frame-based dialogue systems," *Natural Language Engineering*, vol. 15, no. 2, pp. 273–307, 2009.

[5] D. Bohus and A. Rudnicky, "RavenClaw: Dialog Management Using Hierarchical Task Decomposition and an Expectation Agenda," in *Proc. Eurospeech*, Geneva, Switzerland, 2003.

[6] S. Varges, S. Quarteroni, G. Riccardi, A.V. Ivanov, and P. Roberti, "Combining POMDPs trained with User Simulations and Rule-based Dialogue Management in a Spoken Dialogue System," in *Proc. of ACL-IJCNLP 2009 Software Demonstrations*, Suntec, Singapore, 2009.

[7] H. Kurniawati, D. Hsu, and W.S. Lee, "SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces," in *Proc. Robotics: Science and Systems*, 2008.

[8] S. Varges and G. Riccardi, "A Data-centric Architecture for Data-driven Spoken Dialogue Systems," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Kyoto, Japan, 2007.