Training and Evaluation of the HIS POMDP Dialogue System in Noise

M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, K. Yu, S. Young
Machine Intelligence Laboratory
Engineering Department
Cambridge University
United Kingdom

Abstract

This paper investigates the claim that a dialogue manager modelled as a Partially Observable Markov Decision Process (POMDP) can achieve improved robustness to noise compared to conventional state-based dialogue managers. Using the Hidden Information State (HIS) POMDP dialogue manager as an exemplar, and an MDP-based dialogue manager as a baseline, evaluation results are presented for both simulated and real dialogues in a Tourist Information Domain. The results on the simulated data show that the inherent ability to model uncertainty, allows the POMDP model to exploit alternative hypotheses from the speech understanding system. The results obtained from a user trial show that the HIS system with a trained policy performed significantly better than the MDP baseline.

1 Introduction

Conventional spoken dialogue systems operate by finding the most likely interpretation of each user input, updating some internal representation of the dialogue state and then outputting an appropriate response. Error tolerance depends on using confidence thresholds and where they fail, the dialogue manager must resort to quite complex recovery procedures. Such a system has no explicit mechanisms for representing the inevitable uncertainties associated with speech understanding or the ambiguities which naturally arise in interpreting a user's intentions. The result is a system that is inherently fragile, especially in noisy conditions or where the user is unsure of how to use the system.

It has been suggested that Partially Observable Markov Decision Processes (POMDPs) offer a natural framework for building spoken dialogue systems which can both model these uncertainties and support policies which are robust to their effects (Young, 2002; Williams and Young, 2007a). The key idea of the POMDP is that the underlying dialogue state is hidden and dialogue management policies must therefore be based not on a single state estimate but on a distribution over all states.

Whilst POMDPs are attractive theoretically, in practice, they are notoriously intractable for anything other than small state/action spaces. Hence, practical examples of their use were initially restricted to very simple domains (Roy et al., 2000; Zhang et al., 2001). More recently, however, a number of techniques have been suggested which do allow POMDPs to be scaled to handle real world tasks. The two generic mechanisms which facilitate this scaling are factoring the state space and performing policy optimisation in a reduced *summary state space* (Williams and Young, 2007a; Williams and Young, 2007b).

Based on these ideas, a number of real-world POMDP-based systems have recently emerged. The most complex entity which must be represented in the state space is the user's goal. In the *Bayesian Update of Dialogue State (BUDS)* system, the user's goal is further factored into conditionally independent *slots*. The resulting system is then modelled as a dynamic Bayesian network (Thomson et al., 2008). A similar approach is also developed in

(Bui et al., 2007a; Bui et al., 2007b). An alternative approach taken in the *Hidden Information State (HIS)* system is to retain a complete representation of the user's goal, but partition states into equivalence classes and prune away very low probability partitions (Young et al., 2007; Thomson et al., 2007; Williams and Young, 2007b).

Whichever approach is taken, a key issue in a real POMDP-based dialogue system is its ability to be robust to noise and that is the issue that is addressed in this paper. Using the HIS system as an exemplar, evaluation results are presented for a real-world tourist information task using both simulated and real users. The results show that a POMDP system can learn noise robust policies and that N-best outputs from the speech understanding component can be exploited to further improve robustness.

The paper is structured as follows. Firstly, in Section 2 a brief overview of the HIS system is given. Then in Section 3, various POMDP training regimes are described and evaluated using a simulated user at differing noise levels. Section 4 then presents results from a trial in which users conducted various tasks over a range of noise levels. Finally, in Section 5, we discuss our results and present our conclusions.

2 The HIS System

2.1 Basic Principles

A POMDP-based dialogue system is shown in Figure 1 where s_m denotes the (unobserved or hidden) machine state which is factored into three components: the last user act a_u, the user's goal s_u and the dialogue history s_d. Since s_m is unknown, at each time-step the system computes a belief state such that the probability of being in state s_m given belief state b is $b(s_m)$. Based on this current belief state b, the machine selects an action a_m, receives a reward $r(s_m, a_m)$, and transitions to a new (unobserved) state s'_m, where s'_m depends only on s_m and a_m. The machine then receives an observation o' consisting of an N-best list of hypothesised user actions. Finally, the belief distribution b is updated based on o' and a_m as follows:

$$b'(s'_m) = kP(o'|s'_m, a_m) \sum_{s_m \in S_m} P(s'_m|a_m, s_m) b(s_m) \quad (1)$$

where k is a normalisation constant (Kaelbling et al., 1998). The first term on the RHS of (1) is called the *observation model* and the term inside the summation is called the *transition model*. Maintaining this belief state as the dialogue evolves is called *belief monitoring*.

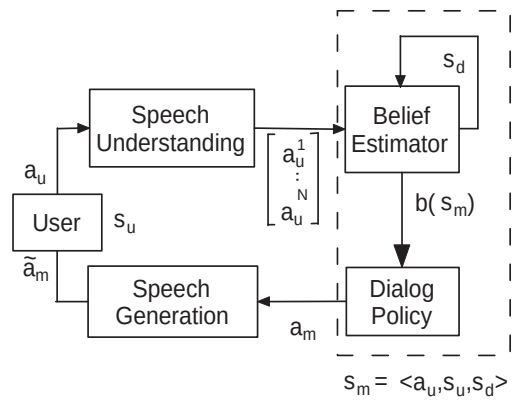

Figure 1: Abstract view of a POMDP-based spoken dialogue system

At each time step t, the machine receives a reward $r(b_t, a_{m,t})$ based on the current belief state b_t and the selected action $a_{m,t}$. Each action $a_{m,t}$ is determined by a policy $\pi(b_t)$ and building a POMDP system involves finding the policy π^* which maximises the discounted sum R of the rewards

$$R = \sum_{t=0}^{\infty} \lambda^t r(b_t, a_{m,t}) \quad (2)$$

where λ^t is a discount coefficient.

2.2 Probability Models

In the HIS system, user goals are partitioned and initially, all states $s_u \in S_u$ are regarded as being equally likely and they are placed in a single partition p_0. As the dialogue progresses, user inputs result in changing beliefs and this root partition is repeatedly split into smaller partitions. This splitting is binary, i.e. $p \rightarrow \{p', p - p'\}$ with probability $P(p'|p)$. By replacing s_m by its factors (s_u, a_u, s_d) and making reasonable independence assumptions, it can be shown (Young et al., 2007) that in parti-

tioned form (1) becomes

$$b'(p', a'_u, s'_d) = k \cdot \underbrace{P(o'|a'_u)}_{\text{observation model}} \underbrace{P(a'_u|p', a_m)}_{\text{user action model}} \cdot \sum_{s_d} \underbrace{P(s'_d|p', a'_u, s_d, a_m)}_{\text{dialogue model}} \underbrace{P(p'|p)b(p, s_d)}_{\text{partition splitting}} \quad (3)$$

where p is the parent of p'.

In this equation, the *observation model* is approximated by the normalised distribution of confidence measures output by the speech recognition system. The *user action model* allows the observation probability that is conditioned on a'_u to be scaled by the probability that the user would speak a'_u given the partition p' and the last system prompt a_m. In the current implementation of the HIS system, user dialogue acts take the form $act(a = v)$ where act is the dialogue type, a is an attribute and v is its value [for example, *request(food=Chinese)*]. The user action model is then approximated by

$$P(a'_u|p', a_m) \approx P(\mathcal{T}(a'_u)|\mathcal{T}(a_m))P(\mathcal{M}(a'_u)|p') \quad (4)$$

where $\mathcal{T}(\cdot)$ denotes the *type* of the dialogue act and $\mathcal{M}(\cdot)$ denotes whether or not the dialogue act *matches* the current partition p'. The dialogue model is a deterministic encoding based on a simple grounding model. It yields probability one when the updated dialogue hypothesis (i.e., a specific combination of p', a'_u, s_d and a_m) is consistent with the history and zero otherwise.

2.3 Policy Representation

Policy representation in POMDP-systems is non-trivial since each action depends on a complex probability distribution. One of the simplest approaches to dealing with this problem is to discretise the state space and then associate an action with each discrete grid point. To reduce quantisation errors, the HIS model first maps belief distributions into a reduced *summary space* before quantising. This summary space consists of the probability of the top two hypotheses plus some status variables and the user act type associated with the top distribution. Quantisation is then performed using a simple distance metric to find the nearest grid point. Actions in summary space refer specifically to the top two hypotheses, and unlike actions in master space, they are limited to a small finite set: *greet*, *ask*, *explicit_confirm*, *implicit_confirm*, *select_confirm*, *offer*, *inform*, *find_alternative*, *query_more*, *goodbye*. A simple heuristic is then used to map the selected next system action back into the full *master* belief space.

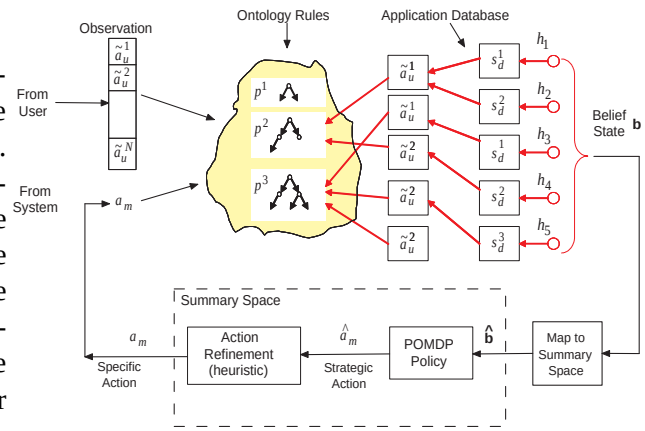

Figure 2: Overview of the HIS system dialogue cycle

The dialogue manager is able to support negations, denials and requests for alternatives. When the selected summary action is to offer the user a venue, the summary-to-master space mapping heuristics will normally offer a venue consistent with the most likely user goal hypothesis. If this hypothesis is then rejected its belief is substantially reduced and it will no longer be the top-ranking hypothesis. If the next system action is to make an alternative *offer*, then the new top-ranking hypothesis may not be appropriate. For example, if an expensive French restaurant near the river had been offered and the user asks for one nearer the centre of town, any alternative offered should still include the user's confirmed desire for an expensive French restaurant. To ensure this, all of the grounded features from the rejected hypothesis are extracted and all user goal hypotheses are scanned starting at the most likely until an alternative is found that matches the grounded features. For the current turn only, the summary-to-master space heuristics then treat this hypothesis as if it was the top-ranking one. If the system then offers a venue based on this hypothesis, and the user accepts it, then, since system outputs are appended to user inputs for the purpose of belief updating, the

alternative hypothesis will move to the top, or near the top, of the ranked hypothesis list. The dialogue then typically continues with its focus on the newly offered alternative venue.

2.4 Summary of Operation

To summarise, the overall processing performed by the HIS system in a single dialogue turn (i.e. one cycle of system output and user response) is as shown in Figure 2. Each user utterance is decoded into an N-best list of dialogue acts. Each incoming act plus the previous system act are matched against the forest of user goals and partitions are split as needed. Each user act a_u is then duplicated and bound to each partition p. Each partition will also have a set of dialogue histories s_d associated with it. The combination of each p, a_u and updated s_d forms a new dialogue hypothesis h_k whose beliefs are evaluated using (3). Once all dialogue hypotheses have been evaluated and any duplicates merged, the master belief state b is mapped into summary space $\hat{b}$ and the nearest policy belief point is found. The associated summary space machine action $\hat{a}_m$ is then heuristically mapped back to master space and the machine's actual response a_m is output. The cycle then repeats until the user's goal is satisfied.

3 Training and Evaluation with a Simulated User

3.1 Policy optimisation

Policy optimisation is performed in the discrete summary space described in the previous section using on-line batch ϵ-greedy policy iteration. Given an existing policy π, dialogs are executed and machine actions generated according to π except that with probability ϵ a random action is generated. The system maintains a set of belief points $\{\hat{\boldsymbol{b}}_i\}$. At each turn in training, the nearest stored belief point $\hat{\boldsymbol{b}}_k$ to $\hat{\boldsymbol{b}}$ is located using a distance measure. If the distance is greater than some threshold, $\hat{\boldsymbol{b}}$ is added to the set of stored belief points. The sequence of points $\hat{\boldsymbol{b}}_k$ traversed in each dialogue is stored in a list. Associated with each $\hat{\boldsymbol{b}}_i$ is a function $Q(\hat{\boldsymbol{b}}_i, \hat{a}_m)$ whose value is the expected total reward obtained by choosing summary action $\hat{a}_m$ from state $\hat{\boldsymbol{b}}_i$. At the end of each dialogue, the total reward is calculated and added to an accumulator for each point in the list, discounted by λ at each step. On completion of a batch of dialogs, the Q values are updated according to the accumulated rewards, and the policy updated by choosing the action which maximises each Q value. The whole process is then repeated until the policy stabilises.

In our experiments, ϵ was fixed at 0.1 and λ was fixed at 0.95. The reward function used attempted to encourage short successful dialogues by assigning $+20$ for a successful dialogue and -1 for each dialogue turn.

3.2 User Simulation

To train a policy, a user simulator is used to generate responses to system actions. It has two main components: a *User Goal* and a *User Agenda*. At the start of each dialogue, the goal is randomly initialised with requests such as "name", "addr", "phone" and constraints such as "type=restaurant", "food=Chinese", etc. The agenda stores the dialogue acts needed to elicit this information in a stack-like structure which enables it to temporarily store actions when another action of higher priority needs to be issued first. This enables the simulator to refer to previous dialogue turns at a later point. To generate a wide spread of realistic dialogs, the simulator reacts wherever possible with varying levels of patience and arbitrariness. In addition, the simulator will relax its constraints when its initial goal cannot be satisfied. This allows the dialogue manager to learn negotiation-type dialogues where only an approximate solution to the user's goal exists. Speech understanding errors are simulated at the dialogue act level using confusion matrices trained on labelled dialogue data (Schatzmann et al., 2007).

3.3 Training and Evaluation

When training a system to operate robustly in noisy conditions, a variety of strategies are possible. For example, the system can be trained only on noise-free interactions, it can be trained on increasing levels of noise or it can be trained on a high noise level from the outset. A related issue concerns the generation of grid points and the number of training iterations to perform. For example, allowing a very large number of points leads to poor performance due to over-fitting of the training data. Conversely, having too few point leads to poor performance due to a lack

of discrimination in its dialogue strategies.

After some experimentation, the following training schedule was adopted. Training starts in a noise free environment using a small number of grid points and it continues until the performance of the policy levels off. The resulting policy is then taken as an initial policy for the next stage where the noise level is increased, the number of grid points is expanded and the number of iterations is increased. This process is repeated until the highest noise level is reached. This approach was motivated by the observation that a key factor in effective reinforcement learning is the balance between exploration and exploitation. In POMDP policy optimisation which uses dynamically allocated grid points, maintaining this balance is crucial. In our case, the noise introduced by the simulator is used as an implicit mechanism for increasing the exploration. Each time exploration is increased, the areas of state-space that will be visited will also increase and hence the number of available grid points must also be increased. At the same time, the number of iterations must be increased to ensure that all points are visited a sufficient number of times. In practice we found that around 750 to 1000 grid points was sufficient and the total number of simulated dialogues needed for training was around 100,000.

A second issue when training in noisy conditions is whether to train on just the 1-best output from the simulator or train on the N-best outputs. A limiting factor here is that the computation required for N-best training is significantly increased since the rate of partition generation in the HIS model increases exponentially with N. In preliminary tests, it was found that when training with 1-best outputs, there was little difference between policies trained entirely in no noise and policies trained on increasing noise as described above. However, policies trained on 2-best using the incremental strategy did exhibit increased robustness to noise. To illustrate this, Figures 3 and 4 show the average dialogue success rates and rewards for 3 different policies, all trained on 2-best: a hand-crafted policy (hdc), a policy trained on noise-free conditions (noise_free) and a policy trained using the incremental scheme described above (increm). Each policy was tested using 2-best output from the simulator across a range of error rates. In addition, the noise-free policy was also tested on 1-best output.

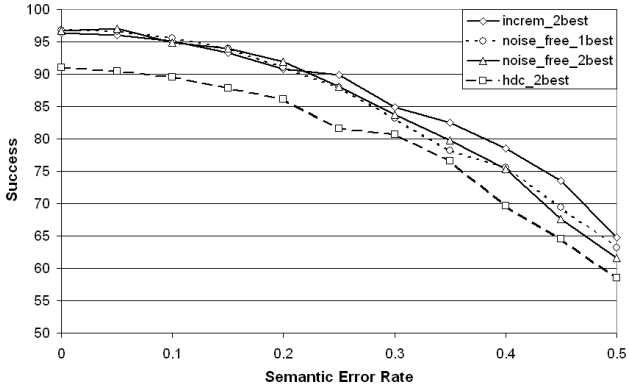

Figure 3: Average simulated dialogue success rate as a function of error rate for a hand-crafted (hdc), noise-free and incrementally trained (increm) policy.

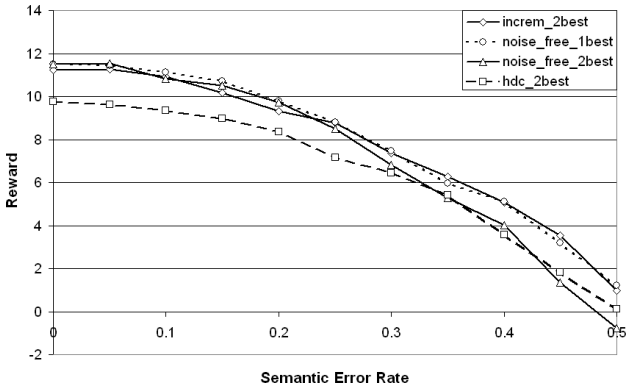

Figure 4: Average simulated dialogue reward as a function of error rate for a hand-crafted (hdc), noise-free and incrementally trained (increm) policy.

As can be seen, both the trained policies improve significantly on the hand-crafted policies. Furthermore, although the average rewards are all broadly similar, the success rate of the incrementally trained policy is significantly better at higher error rates. Hence, this latter policy was selected for the user trial described next.

4 Evaluation via a User Trial

The HIS-POMDP policy (HIS-TRA) that was incrementally trained on the simulated user using 2-best lists was tested in a user trial together with a hand-crafted HIS-POMDP policy (HIS-HDC). The strategy used by the latter was to first check the most likely hypothesis. If it contains sufficient grounded

keys to match 1 to 3 database entities, then *offer* is selected. If any part of the hypothesis is inconsistent or the user has explicitly asked for another suggestion, then *find_alternative* action is selected. If the user has asked for information about an offered entity then *inform* is selected. Otherwise, an ungrounded component of the top hypothesis is identified and depending on the belief, one of the confirm actions is selected.

In addition, an MDP-based dialogue manager developed for earlier trials (Schatzmann, 2008) was also tested. Since considerable effort has been put in optimising this system, it serves as a strong baseline for comparison. Again, both a trained policy (MDP-TRA) and a hand-crafted policy (MDP-HDC) were tested.

4.1 System setup and confidence scoring

The dialogue system consisted of an ATK-based speech recogniser, a Phoenix-based semantic parser, the dialogue manager and a diphone based speech synthesiser. The semantic parser uses simple phrasal grammar rules to extract the dialogue act type and a list of attribute/value pairs from each utterance.

In a POMDP-based dialogue system, accurate belief-updating is very sensitive to the confidence scores assigned to each user dialogue act. Ideally these should provide a measure of the probability of the decoded act given the true user act. In the evaluation system, the recogniser generates a 10-best list of hypotheses at each turn along with a compact confusion network which is used to compute the inference evidence for each hypothesis. The latter is defined as the sum of the log-likelihoods of each arc in the confusion network and when exponentiated and renormalised this gives a simple estimate of the probability of each hypothesised utterance. Each utterance in the 10-best list is passed to the semantic parser. Equivalent dialogue acts output by the parser are then grouped together and the dialogue act for each group is then assigned the sum of the sentence-level probabilities as its confidence score.

4.2 Trial setup

For the trial itself, 36 subjects were recruited (all British native speakers, 18 male, 18 female). Each subject was asked to imagine himself to be a tourist in a fictitious town called Jasonville and try to find particular hotels, bars, or restaurants in that town. Each subject was asked to complete a set of predefined tasks where each task involved finding the name of a venue satisfying a set of constraints such as food type is Chinese, price-range is cheap, etc., and getting the value of one or more additional attributes of that venue such as the address or the phone number.

For each task, subjects were given a scenario to read and were then asked to solve the task via a dialogue with the system. The tasks set could either have one solution, several solutions, or no solution at all in the database. In cases where a subject found that there was no matching venue for the given task, he/she was allowed to try and find an alternative venue by relaxing one or more of the constraints.

In addition, subjects had to perform each task at one of three possible noise levels. These levels correspond to signal/noise ratios (SNRs) of 35.3 dB (low noise), 10.2 dB (medium noise), or 3.3 dB (high noise). The noise was artificially generated and mixed with the microphone signal, in addition it was fed into the subject's headphones so that they were aware of the noisy conditions.

An instructor was present at all times to indicate to the subject which task description to follow, and to start the right system with the appropriate noise-level. Each subject performed an equal number of tasks for each system (3 tasks), noise level (6 tasks) and solution type (6 tasks for each of the types 0, 1, or multiple solutions). Also, each subject performed one task for all combinations of system and noise level. Overall, each combination of system, noise level, and solution type was used in an equal number of dialogues.

4.3 Results

In Table 1, some general statistics of the corpus resulting from the trial are given. The semantic error rate is based on substitutions, insertions and deletions errors on semantic items. When tested after the trial on the transcribed user utterances, the semantic error rate was 4.1% whereas the semantic error rate on the ASR input was 25.2%. This means that 84% of the error rate was due to the ASR.

Tables 2 and 3 present success rates (Succ.) and average performance scores (Perf.), comparing the two HIS dialogue managers with the two MDP base-

Number of dialogues	432
Number of dialogue turns	3972
Number of words (transcriptions)	18239
Words per utterance	4.58
Word Error Rate	32.9
Semantic Error Rate	25.2
Semantic Error Rate transcriptions	4.1

Table 1: General corpus statistics.

line systems. For the success rates, also the standard deviation (std.dev) is given, assuming a binomial distribution. The success rate is the percentage of successfully completed dialogues. A task is considered to be fully completed when the user is able to find the venue he is looking for and get all the additional information he asked for; if the task has no solution and the system indicates to the user no venue could be found, this also counts as full completion. A task is considered to be partially completed when only the correct venue has been given. The results on partial completion are given in Table 2, and the results on full completion in Table 3. To mirror the reward function used in training, the performance for each dialogue is computed by assigning a reward of 20 points for full completion and subtracting 1 point for the number of turns up until a successful recommendation (i.e., partial completion).

Partial Task Completion statistics			
System	Succ. (std.dev)	#turns	Perf.
MDP-HDC	68.52 (4.83)	4.80	8.91
MDP-TRA	70.37 (4.75)	4.75	9.32
HIS-HDC	74.07 (4.55)	7.04	7.78
HIS-TRA	84.26 (3.78)	4.63	12.22

Table 2: Success rates and performance results on partial completion.

Full Task Completion statistics			
System	Succ. (std.dev)	#turns	Perf.
MDP-HDC	64.81 (4.96)	5.86	7.10
MDP-TRA	65.74 (4.93)	6.18	6.97
HIS-HDC	63.89 (4.99)	8.57	4.20
HIS-TRA	78.70 (4.25)	6.36	9.38

Table 3: Success rates and performance results on full completion.

The results show that the trained HIS dialogue manager significantly outperforms both MDP based dialogue managers. For success rate on partial completion, both HIS systems perform better than the MDP systems.

4.3.1 Subjective Results

In the user trial, the subjects were also asked for a subjective judgement of the systems. After completing each task, the subjects were asked whether they had found the information they were looking for (yes/no). They were also asked to give a score on a scale from 1 to 5 (best) on how natural/intuitive they thought the dialogue was. Table 4 shows the results for the 4 systems used. The performance of the HIS systems is similar to the MDP systems, with a slightly higher success rate for the trained one and a slightly lower score for the handcrafted one.

System	Succ. Rate (std.dev)	Score
MDP-HDC	78 (4.30)	3.52
MDP-TRA	78 (4.30)	3.42
HIS-HDC	71 (4.72)	3.05
HIS-TRA	83 (3.90)	3.41

Table 4: Subjective performance results from the user trial.

5 Conclusions

This paper has described recent work in training a POMDP-based dialogue manager to exploit the additional information available from a speech understanding system which can generate ranked lists of hypotheses. Following a brief overview of the Hidden Information State dialogue manager and policy optimisation using a user simulator, results have been given for both simulated user and real user dialogues conducted at a variety of noise levels.

The user simulation results have shown that although the rewards are similar, training with 2-best rather than 1-best outputs from the user simulator yields better success rates at high noise levels. In view of this result, we would have liked to investigate training on longer N-best lists, but currently computational constraints prevent this. We hope in the future to address this issue by developing more efficient state partitioning strategies for the HIS system.

The overall results on real data collected from the user trial clearly indicate increased robustness by the HIS system. We would have liked to be able to plot performance and success scores as a function of noise level or speech understanding error rate, but there is great variability in these kinds of complex real-world dialogues and it transpired that the trial data was insufficient to enable any statistically meaningful presentation of this form. We estimate that we need at least an order of magnitude more trial data to properly investigate the behaviour of such systems as a function of noise level. The trial described here, including transcription and analysis consumed about 30 man-days of effort. Increasing this by a factor of 10 or more is not therefore an option for us, and clearly an alternative approach is needed.

We have also reported results of subjective success rate and opinion scores based on data obtained from subjects after each trial. The results were only weakly correlated with the measured performance and success rates. We believe that this is partly due to confusion as to what constituted success in the minds of the subjects. This suggests that for subjective results to be meaningful, measurements such as these will only be really useful if made on live systems where users have a real rather than imagined information need. The use of live systems would also alleviate the data sparsity problem noted earlier.

Finally and in conclusion, we believe that despite the difficulties noted above, the results reported in this paper represent a first step towards establishing the POMDP as a viable framework for developing spoken dialogue systems which are significantly more robust to noisy operating conditions than conventional state-based systems.

Acknowledgements

This research was partly funded by the UK EPSRC under grant agreement EP/F013930/1 and by the EU FP7 Programme under grant agreement 216594 (CLASSIC project: www.classic-project.org).

References

TH Bui, M Poel, A Nijholt, and J Zwiers. 2007a. A tractable DDN-POMDP Approach to Affective Dialogue Modeling for General Probabilistic Frame-based Dialogue Systems. In *Proc 5th Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, pages 34–57.

TH Bui, B van Schooten, and D Hofs. 2007b. Practical dialogue manager development using POMDPs . In *8th SIGdial Workshop on Discourse and Dialogue*, Antwerp.

LP Kaelbling, ML Littman, and AR Cassandra. 1998. Planning and Acting in Partially Observable Stochastic Domains. *Artificial Intelligence*, 101:99–134.

N Roy, J Pineau, and S Thrun. 2000. Spoken Dialogue Management Using Probabilistic Reasoning. In *Proc ACL*.

J Schatzmann, B Thomson, and SJ Young. 2007. Error Simulation for Training Statistical Dialogue Systems. In *ASRU 07*, Kyoto, Japan.

J Schatzmann. 2008. *Statistical User and Error Modelling for Spoken Dialogue Systems*. Ph.D. thesis, University of Cambridge.

B Thomson, J Schatzmann, K Weilhammer, H Ye, and SJ Young. 2007. Training a real-world POMDP-based Dialog System. In *HLT/NAACL Workshop "Bridging the Gap: Academic and Industrial Research in Dialog Technologies"*, Rochester.

B Thomson, J Schatzmann, and SJ Young. 2008. Bayesian Update of Dialogue State for Robust Dialogue Systems. In *Int Conf Acoustics Speech and Signal Processing ICASSP*, Las Vegas.

JD Williams and SJ Young. 2007a. Partially Observable Markov Decision Processes for Spoken Dialog Systems. *Computer Speech and Language*, 21(2):393–422.

JD Williams and SJ Young. 2007b. Scaling POMDPs for Spoken Dialog Management. *IEEE Audio, Speech and Language Processing*, 15(7):2116–2129.

SJ Young, J Schatzmann, K Weilhammer, and H Ye. 2007. The Hidden Information State Approach to Dialog Management. In *ICASSP 2007*, Honolulu, Hawaii.

SJ Young. 2002. Talking to Machines (Statistically Speaking). In *Int Conf Spoken Language Processing*, Denver, Colorado.

B Zhang, Q Cai, J Mao, E Chang, and B Guo. 2001. Spoken Dialogue Management as Planning and Acting under Uncertainty. In *Eurospeech*, Aalborg, Denmark.