

# User study of the Bayesian Update of Dialogue State approach to dialogue management

B. Thomson, M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, K. Yu, S. Young

Engineering Department, Cambridge University, CB2 1PZ, UK

{brmt2, mg436, sk561, farm2, js532, ky219, sjy}@eng.cam.ac.uk

## Abstract

This paper presents the results of a comparative user evaluation of various approaches to dialogue management. The major contribution is a comparison of traditional systems against a system that uses a Bayesian Update of Dialogue State approach. This approach is based on the Partially Observable Markov Decision Process (POMDP), which has previously been shown to give improved robustness in simulation experiments. Results from this paper show that the benefits demonstrated in simulation experiments are also obtained when testing a live system with real users.

## 1. Introduction

Recent research on the theory of dialogue systems has suggested the use of Partially Observable Markov Decision Processes (POMDPs) as a suitable approach. At low error rates, POMDP-based systems are expected to give equivalent performance to traditional finite state approaches. In the presence of noise, however, they should provide increased robustness to errors by handling uncertainty in a more principled way.

The expected benefits of the POMDP approach have been verified in simulation experiments by various researchers [1, 2, 3]. Unfortunately, this does not necessarily mean that the POMDP-based systems are better with real users since it is well documented that simulation results are not always reliable [4]. The aim of this paper is to show that the benefits shown in simulation experiments are confirmed by experiments with real users.

Past efforts to evaluate POMDP-based systems have been hindered by various obstacles. The POMDP relies heavily on the confidence scores it obtains from the semantic processing component. Unless the confidence scores give a good indication of the true probability of the semantics of a user utterance, the system has little information with which to update its beliefs. It is also very difficult to scale POMDP systems to the number of states required by a real-world dialogue manager. Without efficient approximation techniques the updating and learning quickly becomes intractable. Several approaches have been suggested to overcome this including the Composite Summary Point Based Value Iteration (CSPBVI) algorithm and the Hidden Information State model [5, 6].

This paper presents a user evaluation of an alternative to these scaling techniques called the Bayesian Update of Dialogue State (BUDS) model [3]. The paper is organised as follows. Section 2 explains the POMDP model as well as the BUDS approach to scaling. The following section then explains the details of the dialogue manager implementations along with simulated results. Section 4 gives a user-based evaluation of the system and section 5 concludes the paper.

## 2. Bayesian Update of Dialogue State

A standard model for dialogue management is that the system consists of a set of internal states, allowable machine actions  $\mathcal{A}$  and a set of observations  $\mathcal{O}$ . The machine actions,  $m \in \mathcal{A}$ , represent the allowable machine utterances while the observations,  $o \in \mathcal{O}$ , give the possible results after semantic processing of the user's speech. Decisions about which action to take are dependent on the internal state of the system.

When using a POMDP approach, the concept of internal system state is equated to the system's *beliefs* about the user and environment. The true state of the user and environment,  $s \in \mathcal{S}$ , is considered hidden and must be inferred from the observations received. This inference is done under the Markov assumption: states are conditionally dependent on only their previous value and the last machine action. Given an observation function  $P(o|s)$  and a state transition function  $P(s'|s, m)$ , a belief distribution over states can then be updated using Bayes' Theorem [2]. The set of internal system states is then the set of probability distributions over  $\mathcal{S}$ , sometimes called the *belief space*.

To make this update equation more tractable the Bayesian Update of Dialogue State (BUDS) framework makes several further assumptions. As suggested by [2], the hidden state is factored into three components: the true user goal  $g$ , the true user action  $u$  and the true dialogue history  $h$ . The BUDS approach then further factorises according to a set of slots,  $i \in \mathcal{I}$ . The goal becomes  $g = (g_i)_{i \in \mathcal{I}}$ , the user act  $u = (u_i)_{i \in \mathcal{I}}$  and the dialogue history  $h = (h_i)_{i \in \mathcal{I}}$ . Next some conditional independence assumptions are taken and are modeled as a Bayesian Network. Figure 1 shows the resulting network for two time-slices of a two-slot system based on this idea.

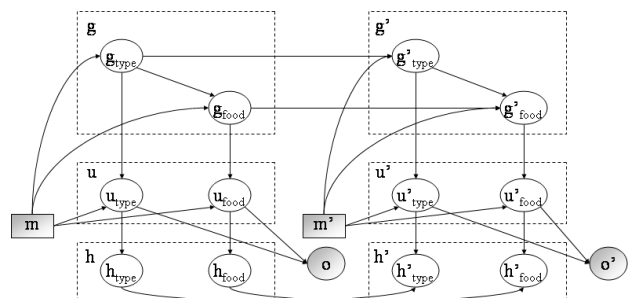


Figure 1: The two time-slice network for a BUDS based system with two slots: type of venue and food. The food slot only applies if the type of venue is restaurant and is assumed to take the value “N/A” otherwise.

One of the advantages of factoring the belief state is that handcrafted policies are reasonably simple to implement. The

system designer specifies which action to take as a function of the belief distributions over each node in the network. As an example, the system might request information about a node whenever there is no single value with a high probability. Once a sufficient number of nodes have highly probable values the system will offer information about a relevant database item.

An alternative to handcrafting the policy is to learn it automatically. One assigns rewards to the various stages of the dialogue and learning proceeds using the episodic Natural Actor Critic algorithm [7]. This algorithm follows a parametric stochastic policy which is updated by calculating a natural policy gradient and performing gradient descent. The parameters are factored so that each node in the network has its own parameters which contribute to the overall choice of action. Further details may be found in [3]

### 3. System implementation

Four dialogue systems were built for the Tourist Information domain to compare the POMDP approach against traditional alternatives and to compare handcrafted policies with learned ones. The task in this domain is to provide information and recommendations for restaurants, hotels and bars in a fictitious town. Users may request specific values for nine different slots: type of venue, number of stars, cuisine, area, drinks, music, price range, name and nearness to a specific location. They may also then ask for information about the address, phone number and a comment on the venue.

The first system is based on the BUDS approach using a handcrafted policy (BUDS-HDC). The system starts by checking the number of venues that are likely to match the user's requirements by checking the probabilities of the venue's slot values. If the total number of likely matches is below a threshold then one of these venues is offered to the user. Alternatively, the system finds a slot where there is high uncertainty and requests the slot, confirms the most likely value or asks the user to select between two options. The choice between these actions is decided by simple heuristics based on the probabilities of the most likely value and second most likely value.

A trained policy (BUDS-TRA) was implemented for the BUDS approach as described previously [3]. Policy learning requires several thousand interactive dialogues so it is infeasible to do this with real users. Instead an agenda based simulated user was built for the task [8]. The user simulator assumes a predefined user goal and then interacts with the dialogue manager at a semantic level. Planned dialogue acts are pushed and popped from a stack according to the actions of the dialogue manager, providing a mechanism for the simulator to maintain consistency through the dialogue.

When running simulations on a dialogue manager another important issue is the simulation of errors. In order to handle errors appropriately the system must experience them during training. For a given dialogue act decided by the user simulator, three confused acts were generated. A confusion rate,  $e$ , was chosen and for each output act the original act was chosen with probability  $(1 - e)$ . Otherwise the act was corrupted by changing the dialogue act type, substituting attribute values or types and / or inserting or deleting attribute-value pairs. Based on the number of times an act appeared in the confused list, a confidence score was generated and this was then passed on to the dialogue manager.

The third and fourth systems implemented are based on the traditional approach of using a finite number of states to represent the dialogue state. The handcrafted system (MDP-HDC),

fills a set of slots with dialogue acts as they are received and requests unfilled slots until it finds a suitable venue. Note that this approach can only make use of one hypothesis at a time so any additional information regarding the uncertainty is lost. Using the same internal state with a standard Markov Decision Process model allowed the implementation of a learned policy for this system (MDP-TRA) [9]. Policy learning used the same user and error simulator as the BUDS trained system.

Preliminary testing used simulations to compare the four approaches by using the same user simulator and error channel as for training. The confusion rate was varied between 0% and 50%, with 5000 dialogues simulated at each rate. As can be seen from Figure 2, the systems perform equally well when there is no confusion but as the error rate increases differences begin to develop. The two BUDS systems significantly outperform the traditional approaches and the learned policies give slight improvements over their handcrafted counterparts<sup>1</sup>. The next section discusses an evaluation based on real user data.

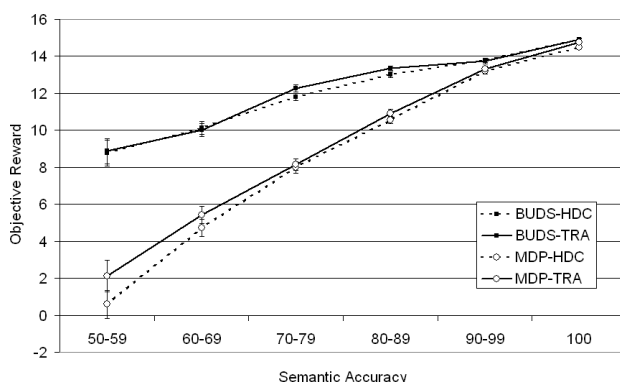


Figure 2: Simulation results of the four systems. The objective reward is calculated by assigning 20 points for a successful dialogue and subtracting one for each turn. Dialogues are grouped into bins according to the semantic accuracy using only the top hypothesis for each turn. Error-bars show one standard error on each side of the mean.

### 4. User Evaluation

The four dialogue managers discussed above were evaluated by 36 native English speakers in an attempt to investigate system performance on real users. The subjects, who had not been involved in previous dialogue system experiments, were asked to complete three tasks on each of the four systems. Tasks consisted of finding a venue matching a set of constraints, along with some information about it. For each system, every user was given one task where there was no suitable venue, one where there was more than one and one where there was exactly one matching venue. 108 dialogues were recorded for each system.

To investigate system robustness varying levels of synthetic noise were added to the speech signal before speech recognition. Three levels of zero, medium and high noise were used, corresponding to signal to noise ratios (SNR) of 35.3db, 10.2db

<sup>1</sup>Note that the difference between the BUDS trained and handcrafted policies is smaller than in previous experiments because of a change in evaluation metric[3]. In the case of tasks with no matching venue, a dialogue is considered successful if the system says there is no matching venue. In previous experiments the user simulator would change its goal and require a venue matching the new goal but this is difficult to test in a real environment.

and 3.3db. The users were given a different noise level for each of the three tasks for each system.

Three metrics were used to evaluate the system performance. At the end of each dialogue users were asked whether they felt they had found what they were looking for, resulting in a subjective success rate (SSR). An objective success rate (OSR) was calculated by examining the transcripts and checking if the task requirements had been met. From this the reward was calculated by giving twenty points for a successful dialogue, zero for an unsuccessful one and subtracting the number of turns until success or dialogue completion.

It was important to try and eliminate as much extra variability as possible so all systems used the same speech recogniser, semantic decoder, output generator and text-to-speech engine. The speech recogniser uses the ATK toolkit with a tri-phone acoustic model and a dictionary of around 2000 in-domain words. A 10-best list is output and passed to the semantic decoder along with sentence-level inference evidence scores, which are the sum of the sentence arc log-likelihoods in the confusion network. Semantic decoding is implemented using a hand-crafted Phoenix-based parser which computes for each sentence a dialogue act type along with a sequence of attribute value pairs. A confidence for each resulting dialogue act is calculated by exponentiating the inference evidence, adding the score for sentences that result in the same dialogue act and renormalising so that they sum to one. Output generation is handcrafted and text-to-speech is implemented with the FLite synthesis engine.

The overall results of the trial are given in Table 1. As was the case in simulations, the BUDS systems significantly outperformed the MDP-based systems in this trial. Interestingly, the BUDS trained policy fared significantly better than the hand-crafted policy in terms of subjective success but worse on the objective metrics. Preliminary investigation of the transcripts seems to indicate that this was because the trained policy sometimes offered venues before all the information had been given. While the simulated user always ensured its constraints were met, the real users did not necessarily do this. This may be one reason for the reduced performance of the trained policy.

System	OSR	Reward	SSR
BUDS-HDC	$0.84 \pm 0.04$	$11.83 \pm 0.94$	$0.84 \pm 0.04$
BUDS-TRA	$0.75 \pm 0.04$	$8.89 \pm 1.09$	$0.88 \pm 0.03$
MDP-TRA	$0.66 \pm 0.05$	$6.97 \pm 1.23$	$0.81 \pm 0.04$
MDP-HDC	$0.65 \pm 0.05$	$7.10 \pm 1.21$	$0.78 \pm 0.04$

Table 1: Objective Success Rate (OSR), Objective Reward and Subjective Success Rate (SSR) for the different systems. Error values give one standard error of the mean. They are calculated by assuming that success rates and rewards follow binomial and Gaussian distributions respectively.

Figure 3 shows the subjective success rates of the systems when separating the various noise levels. The overall trend is that the BUDS systems do outperform the traditional approaches but these differences are not significant. Due to the small number of samples the results are difficult to interpret, particularly as the trained systems performed better with medium noise than they did with no noise added. The objective success rates gave similar peculiarities; the main difference being that the BUDS-HDC policy performed better than the BUDS-TRA policy. Part of the reason for these peculiarities is that the added synthetic noise did not always result in increased semantic errors. The next subsection discusses how the error rates actually observed during the trial affected the systems.

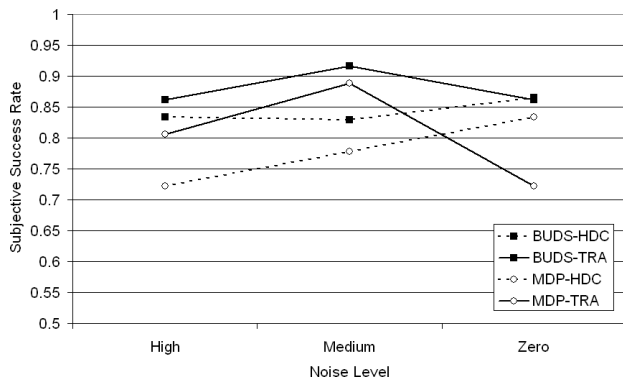


Figure 3: Subjective success rates for the different systems at varying noise levels.

#### 4.1. Effect of Semantic Errors

The major claim of POMDP-based systems is that they are more robust to errors than traditional approaches. Given sufficient data and noting that semantic error rates are affected by the choice of system actions, dialog performance should ideally be plotted directly as a function of noise level. However, as shown in the previous section, the correlation of error rates with noise level can be very weak and when sample sets are small, the results can be misleading. In this case, greater insight can be obtained by plotting performance as function of the measured error rate.

Error rates form a continuum so it is inappropriate to give observed success rates for any given level of error. Either the error rates must be grouped into bins or one must estimate a functional form for the success. Both of these approaches are presented in this section.

The results of binning the error rates are shown in figures 4 and 5. Accuracy rates are calculated for the top semantic hypothesis only and are defined as one minus the sum of the number of insertions, deletions and substitutions of semantic items divided by the reference total. Semantic items include the dialogue act type as well as the attribute-value pairs. Bins were chosen to obtain approximately equal numbers of dialogues in each bin and were split into accuracies less than 70%, between 70% and 85% and over 85%. Since the exact distribution of accuracies for each may be different the figures plot the success rate against the average top hypothesis semantic error rate for the bin.

Assuming a constant probability of success for each bin and system, the results show that the BUDS systems significantly outperform the MDP at higher error levels. Based on subjective success, the BUDS trained policy performs the best, while on the objective metric the handcrafted approach is better. Performance under medium error conditions has similar trends but the differences are less significant.

One problem with grouping data into bins is that the success rate for the bin may depend on the exact distribution of values obtained. It may be preferable to assume that the probability of success is some function of the semantic accuracy and estimate this function instead. A standard approach for this is to use logistic regression but this restricts the particular form of function allowed. A more flexible approach is to use a Gaussian Process (GP) classification model, where the probability of classification is replaced with the probability of dialogue suc-

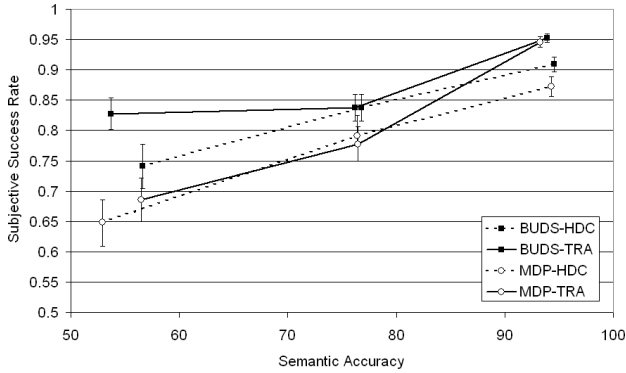


Figure 4: Subjective success rates for the different systems in varying semantic error bins. Error-bars show one standard error on each side of the mean.

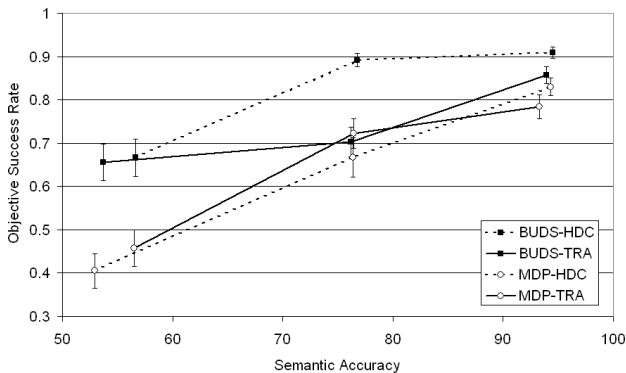


Figure 5: Objective success rates for the different systems in varying semantic error bins. Error-bars show one standard error on each side of the mean.

cess. This model allows for a large class of functions relating semantic accuracy with probability of success. Further details can be found in [10].

Results using the GP approach are shown in figure 6<sup>2</sup>. The data is insufficient to draw any significant conclusions but the trend gives some indication that the learned policies have a smaller decrease in performance for a given drop in accuracy.

## 5. Conclusion

This paper has discussed the results of a comparative evaluation of four dialogue systems, comparing the Bayesian Update of Dialogue State (BUDS) approach with traditional finite state techniques. Both simulations and a user evaluation indicate that the POMDP-based BUDS approach improves performance, particularly in the presence of errors. In the trial the BUDS systems were shown to significantly outperform overall as well as when dialogues were binned as having low semantic accuracy. The effect of noise is less significant, but is probably due to a lack of test data. In future work we hope to test performance of a deployed system, thereby both increasing the number of users

<sup>2</sup>Inference was done with the expectation propagation algorithm and hyperparameters were chosen with type II maximum likelihood. GP models also require the choice of a covariance function for which the Matern class (with  $\nu = \frac{3}{2}$ ) was chosen.

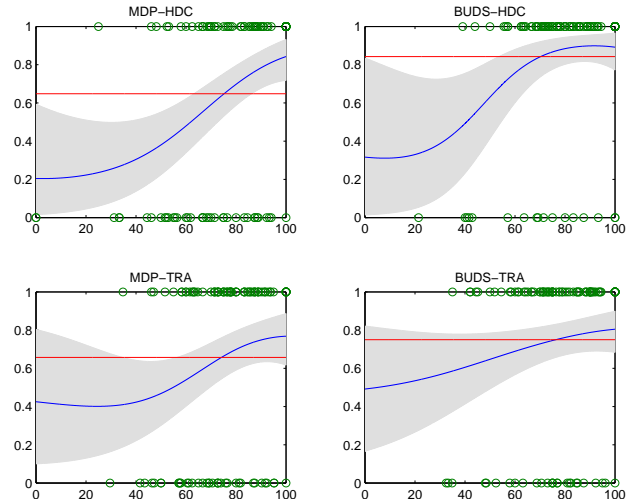


Figure 6: Graphs of the posterior probability of objective dialogue success as a function of semantic accuracy. The horizontal lines show average success. The shaded area gives a 95% confidence interval for the posterior probability of success.

and removing the need to simulate user task scenarios.

## 6. Acknowledgements

This research was partly funded by a St John's Benefactors Scholarship, 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](http://www.classic-project.org)). Thanks to Ulrich Paquet for his help in using Gaussian Processes.

## 7. References

- [1] N. Roy, J. Pineau, and S. Thrun, "Spoken Dialogue Management Using Probabilistic Reasoning," in *Proc. of the ACL*, 2000.
- [2] J. Williams, P. Poupart, and S. Young, "Factored partially observable markov decision processes for dialogue management," in *Workshop on Knowledge and Reasoning in Practical Dialog Systems, IJCAI*, Edinburgh, 2005.
- [3] B. Thomson, J. Schatzmann, and S. Young, "Bayesian update of dialogue state for robust dialogue systems," in *ICASSP*, Las Vegas, NV, 2008.
- [4] J. Schatzmann, M. N. Stuttle, K. Weilhammer, and S. Young, "Effects of the user model on simulation-based learning of dialogue strategies," in *Proc. of ASRU*, 2005.
- [5] J. Williams and S. Young, "Scaling POMDPs for spoken dialog management," *IEEE Transactions on audio, speech and language*, September 2007.
- [6] S. Young, J. Schatzmann, K. Weilhammer, and H. Ye, "The Hidden Information State Approach to Dialog Management," in *Proc. of ICASSP*, Honolulu, Hawaii, 2007.
- [7] J. Peters, S. Vijayakumar, and S. Schaal, "Natural actor-critic," in *Proc. of ECML*, 2005, pp. 280–291, Springer.
- [8] J. Schatzmann, B. Thomson, K. Weilhammer, H. Ye, and S. Young, "Agenda-based user simulation for bootstrapping a POMDP dialogue system," in *Proc. of HLT/NAACL*, 2007.
- [9] E. Levin, R. Pieraccini, and W. Eckert, "A Stochastic Model of Human-Machine Interaction for Learning Dialog Strategies," *IEEE Trans Speech and Audio Processing*, vol. 8, no. 1, 2000.
- [10] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*, Ch 3, MIT Press, 2006.