Evaluation of Statistical POMDP-based Dialogue Systems in Noisy Environments

Steve Young, Catherine Breslin, Milica Gašić, Matthew Henderson, Dongho Kim, Martin Szummer, Blaise Thomson, Pirros Tsiakoulis, Eli Tzirkel Hancock

Abstract Compared to conventional hand-crafted rule-based dialogue management systems, statistical POMDP-based dialogue managers offer the promise of increased robustness, reduced development and maintenance costs, and scaleability to large open-domains. As a consequence, there has been considerable research activity in approaches to statistical spoken dialogue systems over recent years. However, building and deploying a real-time spoken dialogue system is expensive, and even when operational, it is hard to recruit sufficient users to get statistically significant results. Instead, researchers have tended to evaluate using user simulators or by reprocessing existing corpora, both of which are unconvincing predictors of actual real world performance. This paper describes the deployment of a real-world restaurant information system and its evaluation in a motor car using subjects recruited locally and by remote users recruited using Amazon Mechanical Turk. The paper explores three key questions: are statistical dialogue systems more robust than conventional hand-crafted systems; how does the performance of a system evaluated on a user simulator compare to performance with real users; and can performance of a system tested over the telephone network be used to predict performance in more hostile environments such as a motor car? The results show that the statistical approach is indeed more robust, but results from a simulator significantly over-estimate performance both absolute and relative. Finally, by matching WER rates, performance results obtained over the telephone can provide useful predictors of performance in noisier environments such as the motor car, but again they tend to over-estimate performance.

Steve Young
Cambridge University Engineering Department, UK e-mail: sjy@eng.cam.ac.uk

Eli Tzirkel Hancock
General Motors Advanced Technical Center Israel, e-mail: eli.tzirkel@gm.com

All other authors
Cambridge University Engineering Department, UK

1 Introduction

A spoken dialogue system (SDS) allows a user to access information and enact transactions using voice as the primary input-output medium. Unlike so-called voice search applications, the tasks undertaken by an SDS are typically too complex to be achieved by a single voice command. Instead they require a conversation to be held with the user consisting of a number of dialogue turns. Interpreting each user input and deciding how to respond lies at the core of effective SDS design.

In a traditional SDS as shown in Fig. 1, the symbolic components of Fig. 1 are implemented using rules and flowcharts. The semantic decoder uses rule-based surface parsing techniques to extract the most likely user dialogue act and estimate the most likely dialogue state. The choice of system action in response is then determined by if-then else rules applied to the dialogue state or by following a flowchart. These systems are tuned by trial deployment, inspection of performance and iterative refinement of the rules. They can work well in reasonably quiet operating environments when the user knows exactly what to say at each turn. However, they are not robust to speech recognition errors or user confusions, they are expensive to produce and maintain, and they do not scale well as task complexity increases. The latter will be particularly significant as technology moves from limited to open domain systems.

To mitigate against the deficiencies of hand-crafted rule-based systems, statistical approaches to dialogue management have received considerable attention over recent years[1, 2, 3]. The statistical approach is based on the framework of partially observable Markov decision processes (POMDPs)[4]. As shown in Fig. 2, in the statistical approach the dialogue manager is split into two components: a belief tracker which maintains a distribution over all possible dialogue states $b(s)$, and a policy which takes decisions based not on the most likely state but on the whole distribution. The semantic decoder is extended to output a distribution over all possible user dialogue acts and the belief tracker updates its estimate of b every turn using this distribution as evidence. The policy is optimised by defining a reward function for each dialogue turn and then using reinforcement learning to maximise the total (possibly discounted) cumulative reward.

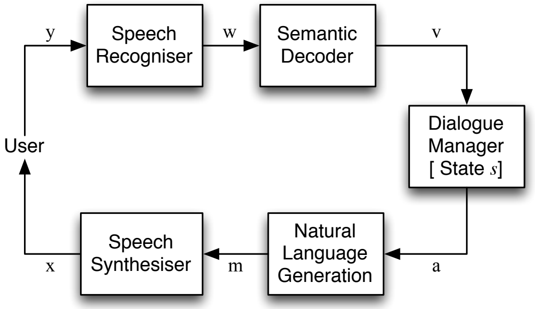

Fig. 1 Block diagram of a conventional SDS. Input speech y is mapped first into words w and then into a user dialogue act v. A dialogue manager tracks the state of the dialogue s and based on this generates a system action a which is converted to a text message m and then into speech x.

One of the difficulties that researchers face when developing an SDS is training and evaluation. Statistical SDS often require a large number of dialogues ($\sim 10^5$ to 10^6) to estimate the parameters of the models, and optimise the policy using reinforcement learning. As a consequence, user simulators are commonly used operating directly at the dialogue act level[5, 6, 7]. These simulators attempt to model real user behaviour. They also include an error model to simulate the effects of speech recognition and semantic decoding errors[8, 9]. A user simulator also provides a convenient tool for testing since it can be run many times and the error rate can be varied over a wide range to test robustness.

The use of simulators obviates the need to build a real system, thereby avoiding all of the engineering complexities involved in integrating telephony interfaces, voice activity detection, recognition and synthesis. However, evaluation using the same user simulator as for training constitutes training and testing under perfectly matched conditions and it is not clear how well this approach can predict system performance with real users.

Even when a real live spoken dialogue system is available for evaluation, there remains the significant problem of recruiting and managing subjects through the tests in sufficient numbers to obtain statistical significance. For example, previous experience (eg. [10]) has shown that direct testing in a motor car is a major undertaking. To provide statistically significant results, a system contrast may require 500 dialogues or more. Recruiting subjects and managing them through in-car tests is slow and expensive. Safety considerations prevent direct testing by the driver, hence testing can only be done by a passenger sitting next to the driver with the microphone system redirected accordingly. Typically, we have found that a team of three assistants plus a driver can process around 6 to 8 subjects per day with each subject completing around 12 to 20 dialogues. Adding the time taken in preparation to recruit and timetable subjects, means that each contrast will typically take about 10 man-days of resource. For large scale development and testing, this is barely practicable.

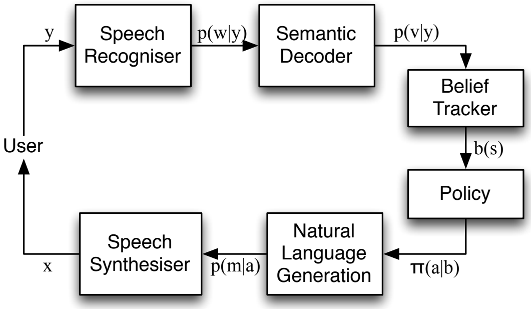

Fig. 2 Block diagram of a statistical SDS. The semantic decoder generates a distribution over possible user dialogue acts v given user input x. A dialogue manager tracks the probability of all possible dialogue states $b(s)$ using $p(v|y)$ as evidence. This distribution b is called the *belief state*. A policy maps b into a distribution over possible system actions a which is converted back into natural language and sampled to provide spoken response x.

Provided that the system is accessible via telephone, one route to mitigating this problem is to use crowd-sourcing web sites such as Amazon Mechanical Turk (MTurk)[11]. This allows subjects to be recruited in large numbers, and it also automates the process of distributing task scenarios and checking whether the dialogues were successful.

This paper describes an experimental study designed to explore these issues. The primary question addressed is whether or not a statistical SDS is more robust than a conventional hand-crafted SDS in a motor car and this was answered by the traditional route of recruiting subjects to perform tasks in a car whilst being driven around a busy town. However, in parallel a phone-based system was configured in which the recogniser's acoustic models were designed to give similar performance to that anticipated in the motor car. This parallel system was tested using MTurk subjects. The results were also compared with those obtained using a user simulator.

The remainder of this paper is organised as follows. Section 2 describes the Bayesian Update of Dialogue State (BUDS) POMDP-based restaurant information system used in the study and the conventional system used in the baseline. Section 3 then describes the experimental set-up in more detail and section 4 reports the results. Finally, section 5 offers conclusions.

2 The Dialogue Systems

Both the conventional baseline and the statistical dialogue system share a common architecture and a common set of understanding and generation components. The recogniser is a real-time implementation of the HTK system[12]. The front-end uses PLP features with energy, 1st, 2nd and 3rd order derivatives mapped into 39 dimensions using a heteroscedastic linear discriminant analysis (HLDA) transform. The acoustic models use conventional HTK tied-state Gaussians and the trigram language model was trained on previously collected dialogue transcriptions with attribute values such as food types, place names, etc. mapped to class names. The semantic decoder extracts n-grams from the confusion networks output by the recogniser and uses a bank of support vector machine (SVM) classifiers to construct a ranked list of dialogue act hypotheses where each dialogue act consists of an act type and a set of attribute value pairs[13, 14]. Some example dialogue acts are shown in Table 1 and a full description is given in [15].

Table 1 Example dialogue acts

Dialogue Act	Example user utterance
`inform(area=centre)`	I want something in the centre of town
`request(phone)`	What's the phone number?
`confirm(pricerange=cheap)`	And it is cheap isn't it?
`affirm(food=chinese)`	Yes, I want chinese food.

The statistical dialogue manager is derived from the BUDS system[16]. In this system the belief state is represented by a dynamic Bayesian network in which the goal, user input and history are factored into conditionally independent attributes (or *slots*) where each slot represents a property of a database entity. An example is shown in Fig 3 for the restaurant domain which shows slots for the type of food (French, Chinese, snacks, etc.), the price-range (cheap, moderate, expensive) and area (central, north, east, etc.). Each time step (ie turn), the observation is instantiated with the output of the semantic decoder, and the marginal probabilities of all of the hidden variables (unshaded nodes) are updated using a form of belief propagation called expectation propagation[17]. The complete set of marginal probabilities encoded in the network constitute the belief state b.

The initial parameters of the Bayesian network are estimated from annotated corpus data. Since expectation propagation can deal with continuous as well as discrete variables, it is also possible to extend the network to include the parameters of the multinomial distributions along with their conjugate Dirichlet priors. The network parameters can then be updated on-line during interaction with real users although that was not done in this trial [18].

The belief state b can be viewed as a vector with dimensionality equal to the cardinality of the state space i.e. $b \in R^{|S|}$ where $|S|$ is equal to the total number of discrete values distributed across all of the nodes in the network. Since this is large, it is compressed to form a set of features appropriate for each action, $\phi_a(b)$. A stochastic policy with parameters θ is then constructed using a softmax function:

$$\pi(a|b;\theta) = \frac{e^{\theta \cdot \phi_a(b)}}{\sum_{a'} e^{\theta \cdot \phi_{a'}(b)}} \tag{1}$$

which represents the probability of taking action a in belief state b. At the end of every turn, the probability of every possible action is sampled using (1), and the most probable action is selected.

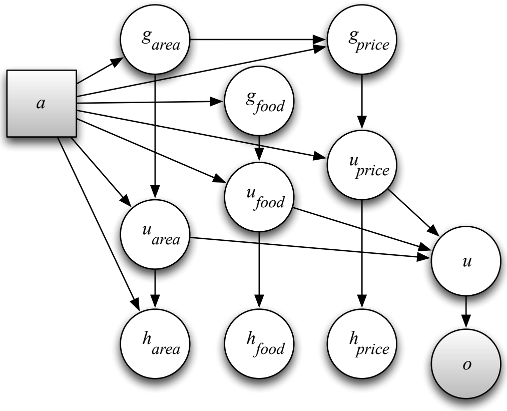

Fig. 3 Example BUDS Dynamic Bayesian Network Structure. Shaded variables are observed, all others are hidden. Each slot is represented by 3 random variables corresponding to the users goal (g), last user input (u) and history (h). The network shown represents just one time slice. All variable nodes are conditioned by the last action. Goal and history nodes are also conditioned on previous time slice.

Since the policy defined by (1) is smoothly differentiable in θ, gradient ascent can be used to adjust the parameter vector θ to maximise the reward[19]. This is done by letting the dialogue system interact with a user simulator[20]. Typically around 10^5 training dialogues are required to fully train the policy.

The baseline dialogue manager consists of a conventional state estimator which maintains a record for each possible slot consisting of the slot status (*filled* or *unfilled*), the slot value, and the confidence derived directly from the confidence of the most likely semantic decoder output. Based on the current state of the slots a set of if-then rules determine which of the possible actions to invoke at the end of each turn. The baseline was developed and tested over a long period and was itself subject to several rounds of iterative refinement using the same user simulator as was used to train the POMDP system.

The output of the dialogue manager in both systems is a system dialogue act following exactly the same schema as for the input. These system acts are converted first to text using a template matching scheme, and then into speech using a HTS-based HMM synthesiser[21]. A fully statistical method of text generation is also available but was not used in this trial to ensure consistency of output across systems[22].

3 Experimental Set-Up

As noted in the introduction, the aims of this evaluation were to firstly establish whether or not a fully statistical dialogue system is more robust in a noisy environment such as a motor car and to investigate the extent to which performance in a specific environment can be predicted by proxy environments which afford testing with higher throughput and lower cost.

The overall system architecture used for the in-car evaluation is shown in Fig. 4. The same system was used for the phone-based MTurk evaluation except that users spoke directly into the phone via a US Toll-free number, rather than via the On-Star Mirror.

3.1 In-car Evaluation

For the in-car evaluation, subjects were recruited using the Gumtree advertising service[1] to ensure variability in demographics. Each of the 12 participants was given 10 dialogue tasks to complete on each system. The systems were called in counter-balanced order across the participants. To elicit more complex dialogues some tasks had no solution in the database and in that case the participant was advised to ask

[1] www.gumtree.com

for something else, e.g. find an Italian restaurant instead of French. Also sometimes the user was asked to find more than one venue that matched the constraints.

To perform the test, each participant was seated in the front passenger seat of a saloon car fitted with the On-Star mirror system and a supervisor sat in the rear seat in order to instruct the subject and monitor the test. The On-Star mirror was affixed to the passenger seat visor to make it useable by the passenger rather than the driver. Power for this assembly was taken from the cars lighter socket. A digital recorder with an external microphone was used to provide a second recording.

The subject received only limited instructions consisting of a brief explanation of what the experiment involved and an example dialogue. For each dialogue the subject informed the supervisor if they thought the dialogue was successful. After the experiment the subjects were asked to fill in a questionnaire.

3.2 Proxy Phone-based Evaluation

By providing a toll-free access number to the system shown in Fig. 4, large numbers of subjects can be recruited quickly and cheaply using crowd sourcing services such as Amazon Mechanical Turk. In order to simulate the effect of a noisy environ-

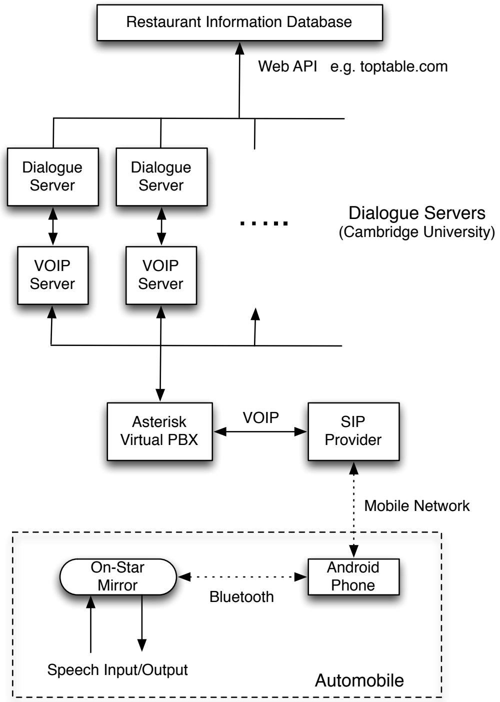

Fig. 4 Block diagram of the overall system architecture used for the in-car evaluation. The On-Star mirror[23] includes a microphone and signal-processing for far-field voice capture in a motor car. The speech input to the mirror is transported via Bluetooth to an Android phone and then over the mobile network to a commercial SIP server (IP-Comms). The signal is then channeled to an Asterisk virtual PABX in order to allow multiple channels to be supported. The PBX routes the call through to an available VOIP server which interfaces directly to the Spoken Dialogue System. At the backend, task related information (in this case restaurant information) is extracted from an on-line database and locally cached.

ment, the technique usually used for off-line speech recognition evaluation is to add randomly aligned segments of pre-recorded background noise to the clean acoustic source. However, in the architecture shown in Fig. 4, this is difficult to achieve for a variety of reasons including ensuring that the user hears an appropriate noise level, avoiding disrupting the voice/activity detection and compensating for the effects of the various non-linear signal processing stages buried in the user's phone, the pabx and the voip conversion. As an alternative, a simpler approach is to reduce the discrimination of the acoustic models in the recogniser so that the recognition performance over the phone was similar to that achieved in the car. This was achieved by reducing the number of Gaussian mixture components to 1 and controlling the decision tree clustering thresholds to fine tune the recogniser using development data from previous phone and in-car evaluations.

Given this change to the recogniser, the experimental protocol for the phone-based evaluation was identical to that used in the car except that the presentation of the tasks and the elicitation of feedback was done automatically using a web-based interface integrated with Amazon Mechanical Turk.

4 Experimental Results

The results of the evaluation are summarised in Table 2. The *in-car* results refer to the supervised tests in a real motor car travelling around the centre of Cambridge, UK, and the *phone proxy* results refer to the phone-based evaluation with MTurk subjects where the speech recogniser's acoustic models were detuned to give similar performance to that obtained in a motor car. Also, shown in this table for comparison are results for a regular phone-based MTurk evaluation using fully trained acoustic models. As can be seen, the average word error rate (WER) obtained in the car driving around town was around 30% compared to the 20% obtained over the telephone. The average WER for the proxy phone system is also around 30% showing that the detuned models performed as required.

Three metrics are reported for each test. Prior to each dialogue, each user was given a task consisting of a set of constraints and an information need such as find the phone number and address of a cheap restaurant selling Chinese food. The objective success rate measures the percentage of dialogues for which the system provided the subject with a restaurant matching the task constraints. If the system provided the correct restaurant and the required information needed such as phone number and address, then this is *full success*. If a valid restaurant was provided, but the user did not obtain the required information (perhaps because they forgot to ask for it), then a *partial success* is recorded. The users *perceived success* rate is measured by asking the subjects if they thought the system had given them all of the information they need. The partial success rate is always higher than the full success rate. Note that the tasks vary in complexity, in some cases the constraints were not immediately achievable in which case the subjects were instructed to relax one of them and try again.

Table 2 Summary of results for in-car and proxy-phone evaluation. Also shown is performance of phone-based system using fully trained acoustic models. Contrasts marked * are statistically significant ($p < 0.05$) using a Kruskal-Wallis rank sum test.

Test	System	Num Dialogs	Objective Success Rate		Perceived Success Rate	Average Turns	WER
			Partial	Full			
In-car	Baseline	118	78.8 ± 3.7 *	67.8 ± 4.3 *	77.1 ± 3.8 *	7.9 ± 3.1	29.7
	POMDP	120	85.0 ± 3.2	75.8 ± 3.9	83.3 ± 3.4	9.7 ± 3.7	26.9
Phone Proxy	Baseline	387	80.1 ± 2.0 *	75.2 ± 2.2 *	91.2 ± 1.4	6.9 ± 3.6	29.4
	POMDP	548	87.0 ± 1.4	81.2 ± 1.7	89.8 ± 1.3	9.3 ± 4.8	30.3
Phone	Baseline	589	88.8 ± 1.3	84.6 ± 1.5	94.4 ± 1.0	6.5 ± 2.9	21.4
	POMDP	578	91.0 ± 1.2	86.9 ± 1.4	94.5 ±1.0	8.3 ± 3.8	21.2

As can be seen in Table 2, the in-car performance of the statistical POMDP based dialogue manager was better than the conventional baseline on all three measures. The proxy phone test showed the same trend for the objective measures but not for the subjective measures. In fact, there is little correlation between the subjective measures and the objective measures in all the MTurk phone tests. A possible explanation is that the subjects in the in-car test were supervised throughout and were therefore more likely to give accurate assessments of the system's performance. The Turks used in the phone tests were not supervised and many might have felt it was safest to say they were satisfied just to make sure they were paid.

The objective proxy phone performance overestimated the actual in-car performance by around 2% on partial success and by around 10% on full success. This may be due to the fact that the subjects in the car found it harder to remember all of the venue details they were required to find. Nevertheless, the proxy phone test provides a reasonable indicator of in-car performance.

To gain more insight into the results, Fig. 5 shows regression plots of predicted full objective success rate as a function of WER computed by pooling all of the trial data. As can be seen, the statistical dialogue system (POMDP-trial) consistently outperforms the conventional baseline system (Baseline-trial). Fig. 5 also plots the success rate of both systems using the user simulator used to train the POMDP system (xxx-sim). It can be seen that the general trend is similar to the user trial data but the simulator success rates significantly overestimate performance, especially for the statistical system. This is probably due to a combination of two effects. Firstly, the user simulator presents perfectly matched data to both systems.[2] Secondly, the simulation of errors will differ to the errors encountered in the real system. In particular, the errors will be largely uncorrelated allowing the belief tracking to gain maximum advantage. When errors are correlated belief tracking is less accurate because it tends to over-estimate alternatives in the N-best list[24].

[2] As well as being used to train the POMDP-based system, the user simulator was used to tune the rules in the conventional hand-crafted system.

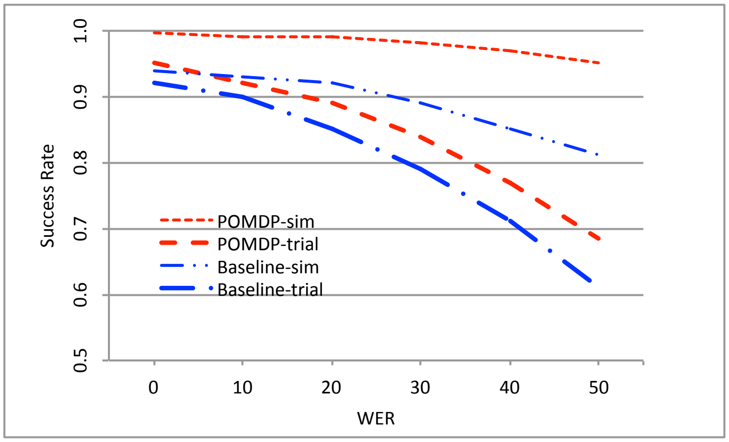

Fig. 5 Comparison of system performance obtained using a user simulator compared to the actual performance achieved in a trial

5 Conclusions

The widespread adoption of end-to-end statistical dialogue systems offers the potential to develop systems which are more robust to noise, and which can be automatically trained to adapt to new and extended domains[25]. However, testing such systems is problematic requiring considerable resource not only to build and deploy working real-time implementations but also to run the large scale experiments needed to properly evaluate them.

The results presented in this paper show that fully statistical systems are not only viable, they also outperform conventional systems especially in challenging environments. The results also suggest that by matching word error rate, crowd sourced phone-based testing can be a useful and economic surrogate for specific environments such as the motor car. This is in contrast to the use of user simulators acting at the dialogue level which grossly exaggerate expected performance. A corollary of this result is that using user simulators to train statistical dialogue systems is equally undesirable, and this observation is supported by recent results which show that when a statistical dialogue system is trained directly by real users, success rates further improve relative to conventional systems[26].

References

1. N. Roy, J. Pineau, and S. Thrun, “Spoken dialogue management using probabilistic reasoning,” in *Proceedings of ACL*, 2000.
2. S. Young, “Talking to Machines (Statistically Speaking),” in *Proceedings of ICSLP*, 2002.
3. J. Williams and S. Young, “Partially Observable Markov Decision Processes for Spoken Dialog Systems,” *Computer Speech and Language*, vol. 21, no. 2, pp. 393–422, 2007.
4. S. Young, M. Gasic, B. Thomson, and J. Williams, “POMDP-based Statistical Spoken Dialogue Systems: a Review,” *Proceedings IEEE*, vol. 101, no. 5, pp. 1160–1179, 2013.
5. K. Scheffler and S. Young, “Probabilistic simulation of human-machine dialogues,” in *ICASSP*, 2000.
6. O. Pietquin and T. Dutoit, “A probabilistic framework for dialog simulation and optimal strategy learning,” *IEEE Transactions on Speech and Audio Processing, Special Issue on Data Mining of Speech, Audio and Dialog*, vol. 14, no. 2, pp. 589–599, 2006.
7. J. Schatzmann, K. Weilhammer, M. Stuttle, and S. Young, “A Survey of Statistical User Simulation Techniques for Reinforcement-Learning of Dialogue Management Strategies,” *KER*, vol. 21, no. 2, pp. 97–126, 2006.
8. O. Pietquin and S. Renals, “ASR System Modelling for Automatic Evaluation and Optimisation of Dialogue Systems,” in *Int Conf Acoustics Speech and Signal Processing*, (Florida), 2002.
9. B. Thomson, M. Henderson, M. Gasic, P. Tsiakoulis, and S. Young, “N-Best error simulation for training spoken dialogue systems,” in *IEEE SLT 2012*, (Miami, FL), 2012.
10. P. Tsiakoulis, M. Gašić, M. Henderson, J. Planells-Lerma, J. Prombonas, B. Thomson, K. Yu, S. Young, and E. Tzirkel, “Statistical Methods for Building Robust Spoken Dialogue Systems in an Automobile,” in *Proceedings of the 4th Applied Human Factors and Ergonomics*, 2012.
11. F. Jurčíček, S. Keizer, M. Gašić, F. Mairesse, B. Thomson, K. Yu, and S. Young, “Real user evaluation of spoken dialogue systems using Amazon Mechanical Turk,” in *Proceedings of Interspeech*, 2011.
12. S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book Version 3.4*. Cambridge, England: Cambridge University, 2006.
13. F. Mairesse, M. Gašić, F. Jurčíček, S. Keizer, B. Thomson, K. Yu, and S. Young, “Spoken language understanding from unaligned data using discriminative classification models,” in *Proceedings of ICASSP*, 2009.
14. M. Henderson, M. Gasic, B. Thomson, P. Tsiakoulis, K. Yu, and S. Young, “Discriminative Spoken Language Understanding Using Word Confusion Networks,” in *IEEE SLT 2012*, (Miami, FL), 2012.
15. S. Young, “CUED Standard Dialogue Acts,” report, Cambridge University Engineering Department, 14th October 2007 2007.
16. B. Thomson and S. Young, “Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems,” *Computer Speech and Language*, vol. 24, no. 4, pp. 562–588, 2010.
17. T. Minka, “Expectation Propagation for Approximate Bayesian Inference,” in *Proc 17th Conf in Uncertainty in Artificial Intelligence*, (Seattle), pp. 362–369, Morgan-Kaufmann, 2001.
18. B. Thomson, F. Jurcicek, M. Gasic, S. Keizer, F. Mairesse, K. Yu, and S. Young, “Parameter learning for POMDP spoken dialogue models,” in *IEEE Workshop on Spoken Language Technology (SLT 2010)*, (Berkeley, CA), 2010.
19. F. Jurcicek, B. Thomson, and S. Young, “Natural Actor and Belief Critic: Reinforcement algorithm for learning parameters of dialogue systems modelled as POMDPs,” *ACM Transactions on Speech and Language Processing*, vol. 7, no. 3, 2011.
20. J. Schatzmann, B. Thomson, K. Weilhammer, H. Ye, and S. Young, “Agenda-Based User Simulation for Bootstrapping a POMDP Dialogue System,” in *Proceedings of HLT*, 2007.
21. K. Yu and S. Young, “Continuous F0 Modelling for HMM based Statistical Parametric Speech Synthesis,” *IEEE Audio, Speech and Language Processing*, vol. 19, no. 5, pp. 1071–1079, 2011.

22. F. Mairesse, M. Gašić, F. Jurčíček, S. Keizer, B. Thomson, K. Yu, and S. Young, "Phrase-based statistical language generation using graphical models and active learning," in *Proceedings of ACL*, 2010.
23. OnStar, "OnStar FMV Mirror," 2013. http://www.onstarconnections.com/.
24. J. Williams, "A critical analysis of two statistical spoken dialog systems in public use," in *Spoken Language Technology Workshop (SLT)*, (Miami, FL), 2012.
25. M. Gasic, C. Breslin, M. Henderson, D. Kim, M. Szummer, B. Thomson, P. Tsiakoulis, and S. Young, "POMDP-based dialogue manager adaptation to extended domains," in *SigDial 13*, (Metz, France), 2013.
26. M. Gasic, C. Breslin, M. Henderson, D. Kim, M. Szummer, B. Thomson, P. Tsiakoulis, and S. Young, "On-line Policy Optimisation of Bayesian Spoken Dialogue Systems via Human Interaction," in *ICASSP 2013*, (Vancouver, Canada), 2013.