Advances in Human Aspects of Road and Rail Transportation
CRC Press 2012, Pages 744–753, ISBN: 978-1-4398-7123-2

CHAPTER 75

Statistical Methods for Building Robust Spoken Dialogue Systems in an Automobile

Pirros Tsiakoulis, Milica Gasic, Matthew Henderson, Joaquin Plannels-Lerma, Jorge Prombonas, Blaise Thomson, Kai Yu, Steve Young

Cambridge University Engineering Department
{pt344, mg436, mh521, jp566, jlp54, brmt2, ky219, sjy}@cam.ac.uk

Eli Tzirkel

General Motors Advanced Technical Center – Israel
eli.tzirkel@gm.com

ABSTRACT

We investigate the potential of statistical techniques for spoken dialogue systems in an automotive environment. Specifically, we focus on partially observable Markov decision processes (POMDPs), which have recently been proposed as a statistical framework for building dialogue managers (DMs). These statistical DMs have explicit models of uncertainty, which allow alternative recognition hypotheses to be exploited, and dialogue management policies that can be optimised automatically using reinforcement learning. This paper presents a voice-based in-car system for providing information about local amenities (e.g. restaurants). A user trial is described which compares performance of a trained statistical dialogue manager with a conventional handcrafted system. The results demonstrate the differing behaviours of the two systems and the performance advantage obtained when using the statistical approach.

Keywords: spoken dialogue systems, statistical dialogue management, POMDP

1 INTRODUCTION

Spoken dialogue has many potential advantages over, or in combination with, other modalities in an automobile. This has led to the deployment of various in-car systems integrating spoken dialogue, with applications such as telephony, entertainment, and navigation.

However, the user experience of these systems is often extremely poor. Part of the problem is that the noisy conditions can cause a drastic deterioration of automatic speech recognition performance. This results in systems repeatedly asking the same question, which in turn causes users to become frustrated.

This paper discusses how a change to the dialogue manager can result in improved overall performance, even in the context of poor recognition accuracy. Current dialogue managers are typically designed as deterministic decision networks. Since system designers will usually include a confidence threshold before accepting a given piece of information, the system ends up asking the same question over and over. This decision network structure also makes it difficult for system designers to use the information in N-best lists of speech-recognition hypotheses. The result is typically a system that is not robust to understanding errors.

Recently, a statistical approach to building dialogue managers has been developed based on partially observable Markov decision processes (POMDPs). These statistical DMs have explicit models of uncertainty which allow alternative recognition hypotheses to be exploited, and dialogue management policies that can be optimised automatically using reinforcement learning. This paper will present a voice-based in-car system for providing information about local amenities (e.g. restaurants). A user trial is described which compares performance of a trained statistical dialogue manager with a conventional handcrafted system.

2 SYSTEM OVERVIEW

The system architecture is shown in figure 1. It consists of 5 distinct modules: automatic speech recognition (ASR), semantic decoding or spoken language understanding (SLU), dialogue management (DM), natural language generation (NLG) and speech synthesis or text-to-speech (TTS).

The user speaks to the system, and the acoustic signal y is converted by the speech recognizer into a set of sentence hypotheses w, which represents a probability distribution over all possible things that he might have said. In practice, *w* is represented by an N-best list. The sentence hypotheses are converted into an N-best list of dialogue acts *v* by a semantic decoder. The dialogue manager represents the dialogue state s in a factored form via the triple <*u,g,h*> (see also figure 2) where *u* is the actual (but unknown) user utterance, *g* is the assumed user goal and *h* represents the history (Williams and Young, 2007). Since this state cannot be directly observed the system maintains a probability distribution *b* over this state, which is called the belief state. The belief state is updated at every user turn using Bayesian inference treating the input *v* as evidence.

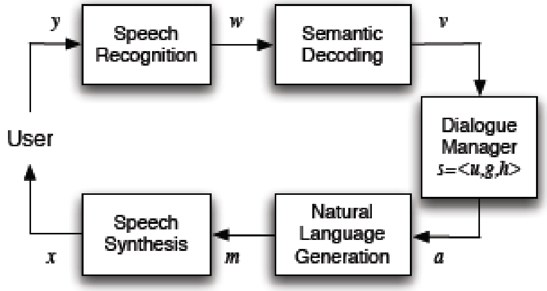

Figure 1 Architecture of the Spoken Dialogue System

Based on *b*, the optimal system response *a* is selected using a mapping $\pi{:}b\rightarrow a$ which is called a policy and which is trained automatically using reinforcement learning. The abstract system dialogue act *a* is converted to an appropriate utterance *m* by a natural language generator and then converted to speech *x* by a HMM-based speech synthesizer.

The spoken dialogue system is implemented using the ATK platform (http://mi.eng.cam.ac.uk/research/dialogue/atk_home.html). ATK is an API designed to facilitate building experimental applications for systems trained using the HTK speech recognition toolkit (see http://htk.eng.cam.ac.uk). It consists of a C++ layer sitting on top of the standard HTK libraries. ATK is multi-threaded and supports real-time speech recognition. ATK allows a variety of components to be connected together to implement different architectures and applications, and hence it provides a flexible platform for building spoken dialogue systems.

To illustrate the system's capability, two typical dialogues are shown in Tables 1 and 2. The first is an example of a straight forward query for which the system is able to provide recommendations immediately. The second is an example where there is no restaurant matching the user's requirements. In this case, the user relaxes his constraints and the system offers him alternatives.

Table 1 An example of a straightforward dialogue

Sys	Hello, welcome to the Cambridge restaurant system? You can ask for restaurants by area, price range or food type. How may I help you?
Usr	Hi, I'm looking for an Indian restaurant in the north.
Sys	There are 3 restaurants in the north of town serving Indian food! Would you like something in the cheap, moderate, or expensive price range?
Usr	I don't mind
Sys	Royal spice is a nice restaurant in the north of town serving Indian food.
Usr	What's the address?
Sys	Sure, It is on Victoria Avenue Chesterton.

Usr	Thanks, bye

Table 2 An example dialogue where the user relaxes their constraints

Sys	Hello, welcome to the Cambridge restaurant system? You can ask for restaurants by area, price range or food type. How may I help you?
Usr	Hi, I'm looking for an expensive Chinese restaurant in the west.
Sys	I am sorry, but there is no expensive restaurant that is in the west part of town and serves Chinese food.
Usr	Do you have anything in the moderate price range?
Sys	Sorry would you like something in the expensive price range or in the moderate price range?
Usr	Moderate.
Sys	I am sorry, but there is no moderately priced restaurant that serves Chinese food and in the west part of town.
Usr	How about the south?
Sys	Would you like Chinese food or you don't care?
Usr	I'd like a Chinese restaurant.
Sys	The Lucky star is a moderately priced Chinese restaurant in the south of town.
Usr	What's the address?
Sys	Sure, It is on Cambridge Leisure Park Clifton Way Cherry Hinton.
Usr	Thank you, goodbye

Next, we describe in more detail each of the basic components of the system.

The **Automatic Speech Recognition (ASR)** module uses the Cambridge HTK/ATK speech recogniser. It accepts live speech input and voice activity detection (VAD) is used to determine the start and end of each user utterance. A tri-gram language model and word-internal triphone HMMs are then applied in a two-pass decoding scheme. In the first pass, a bi-gram language model is used and the resulting lattice is rescored by a tri-gram language model in the second pass. The output is the 10 best hypotheses along with their confidence scores. The pruning thresholds and model size are optimised so that the decoder runs in real time. The acoustic model is trained on data from various sources, including desktop microphone (wide-band data), landline and mobile phone (narrow-band data) and VoIP recording (in-domain narrow-band data), about 130 hours in total. The front-end uses perceptual linear predictor (PLP) features with energy and first, second and third derivatives. A heteroscedastic linear discriminant analysis (HLDA) transform is used to project these features down to 39 dimensions. The language model (LM) is trained using a class-based language modeling technique. The language model training data is about 400K words. The basic idea is to train a generic statistical LM and then expand this generic LM to a specific LM using the information for the specific task in hand. The word list (dictionary) of the generic LM consists of common words used in dialogues (such as good-bye, hello, can ...) and abstract slot labels (such as SLOT_NAME, SLOT_AREA ...). This generic LM is trained on a corpus generated from the 400K-word corpus by substituting abstract slot labels for the actual slot values in our previously collected data. Hence, the generated LM is a special class-based LM consisting of both common words and abstract slot values. We then expand this LM to a specific word-based LM using a class-to-word mapping from the database.

The **Spoken Language Understanding (SLU)** module converts the N-Best ASR output to N-Best dialogue acts with confidence scores. A rule-based hand-crafted Phoenix semantic parser is used in the baseline system (Ward 1991). Various minor modifications of the original system were made to convert the output of this parser into the Cambridge dialogue act format. The Cambridge dialogue format structures the semantic representation as an act type followed by a sequence of items, where each item contains a concept and a value (both of which are optional). Each ASR hypothesis from the N-Best list is parsed into a distinct dialogue act. Identical dialogue acts are then merged and the confidence scores are added together. The resultant multiple dialogue act hypotheses with confidence scores are then fed into the dialogue manager. Table 3 shows the dialogue acts in the Cambridge dialogue semantic representation format for the example shown in Table 1.

Table 3 Example dialogue in the Cambridge dialogue act format

Sys	hello()
Usr	inform(food=indian, area=north, type=restaurant)
Sys	confreq(count="3", type=restaurant, food=indian, area=north, pricerange, option=cheap, option=moderate, option=expensive)
Usr	inform(=dontcare)
Sys	inform(name="Royal spice", type=restaurant, food=Indian, area=north)
Usr	request(addr)
Sys	inform(addr="Victoria Avenue Chesterton")
Usr	bye()

The **Dialogue Management (DM)** component uses the Bayesian Update of Dialogue State (BUDS) framework for updating belief states and a learned policy to select system action (Thomson and Young, 2010). The BUDS DM is an approach within the Partially Observable Markov Decision Process (POMDP) framework. In this approach, the DM maintains a distribution over the dialogue state space, which is referred to as the belief state. The belief state is updated according to the user goal, the observed SLU hypotheses and the dialogue history. The system action is then selected according to the updated belief state. In order to deal with the intractability of both updating belief states and action selection, two techniques of compressing the dialogue state space are employed. The first optimisation is to factorize the user goal and dialogue history further into a series of concepts with concept-level goals (called sub-goals) and concept-level histories.

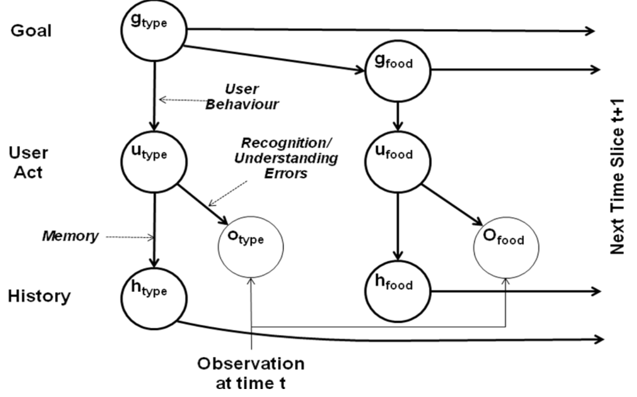

Figure 2 Dependencies in the BUDS dialogue system

Conditional independence assumptions are taken such that the concept-level goal is only dependent on the previous system action, the same concept-level goal of the previous turn and optionally the parent concept-level goal of the previous turn (see figure 2). With the factorization, tractable belief state update can be developed. In the case of the baseline restaurant system, the concepts are name, area, price range, food type, address, post code and signature dish. The second optimisation is used to simplify the action selection process. After updating the belief state, the full state space (also referred to as a master state space) is mapped into a much smaller summary space, in which action selection takes place, resulting in a summary action that gets mapped back into master space using information from the original belief state. The mapping from the (summarized) belief state to system action is referred to as a policy. In the baseline system, a handcrafted policy is used.

The **Natural Language Generation (NLG)** component reads in the dialogue act output by the DM component and generates natural language responses. In the baseline system, we use a template-based NLG. We have also added basic anaphoric references to the NLG to make replies more natural. For instance, when the system suggests a restaurant to the user, the name of that restaurant becomes the topic of the conversation. Instead of repeating the restaurant's name many times, the system refers simply to “it”. When the system operates in noisy conditions, it is often the case that the system has to repeat the same dialogue act. In such cases NLG adds variations in the output text in order to appear more natural to the user.

The **Text-To-Speech (TTS)** component gets text from the NLG components and synthesizes speech using a trainable HMM-based speech synthesizer. Text analysis is first performed to convert raw text into phone-level context labels. Context-dependent HMMs are then used to model spectrum, fundamental frequency and duration of the context-dependent phones. During synthesis, the duration, spectrum and fundamental frequency are generated from the HMMs and then converted to speech waveform. As it is a parametric model, HMM-TTS has more flexibility than traditional unit-selection approaches and is especially useful for producing expressive speech (Yu, Mairesse and Young 2010).

3 PROOF-OF-CONCEPT DEPLOYMENT

The Restaurant Information task was selected for the deployment and evaluation of the system described above in an automotive environment. Using a far field hands-free microphone, users can ask for information directly about a restaurant by name but more interestingly, they can search for a restaurant by expressing their preferences for food type, price range and area. The system will then suggest restaurants and when requested provide the user with details such as post-code, address and telephone number. Users can also change their mind or adjust their constraints if no venues match their initial query. It must be noted that the system uses an open microphone architecture, i.e. there is no push-to-talk. The system is always listening and the user can speak at any time.

In a real world scenario the user would be calling the system via mobile phone while driving, for example someone driving to a new town would like to find out a place to eat. The system would have to select the target town either using location based information (e.g. the selected destination in the GPS device), or if the user specifically requests a town. We have integrated our dialogue system with an online restaurant information service provider. However, we currently assume that the town is known in advance. The TopTable (http://www.toptable.com) web-service provides information on around 20,000 restaurants from around the world. Once the town is selected, the system extracts the following information for each venue: the venue's name, area, price range, food type, address, post code and signature dish. The system ontology is updated with a list of possible names and food types and the semantic decoder is also updated to be able to understand the user talking about the restaurants' names and food types. The ASR and TTS pronunciation dictionaries are automatically updated to include new unknown words, such as restaurant names, addresses, dishes etc. This enables our dialogue system to operate for any town, as long as it is available through the web-service. For the evaluation process the city of Cambridge was selected, and a database of about 150 venues was built.

A VoIP interface was implemented so that remote users can use telephones to communicate with the system. SIPGate (www.sipgate.co.uk) was used as the SIP (Session Initiation Protocol) service provider. This service converts incoming PSTN (Public Switched Telephone Network) calls to IP data and streams them to our server. An open source SIP client, PJSIP, is used to manage SIP calls from SIPGate and the low-level audio IO functionality required by the dialogue server.

4 EVALUATION

The evaluation process took place in two stages. In the first stage we used a handcrafted system to collect training data. We then used the collected data to train statistical models for the semantic decoder and the dialogue manager. In the second stage we evaluated the trained dialogue components in contrast to the handcrafted ones. The aim of the experiment was to test the robustness and performance of a spoken dialogue system in a moving car.

The data collection stage of the experiment was performed in a standing car. About 25 subjects were asked to perform a set of 20 randomly generated tasks. The subject was sitting in the front passenger seat while the operator was sitting in the driver's seat. A call was made using an android mobile phone paired via Bluetooth with an OnStar mirror (www.onstar.com). The mirror's built-in microphone and loudspeakers were used as the speech interface. Two recording conditions were tested; for the first one the car's fan was off, while for the second one the fan was set in its maximum speed. For each task, the subject read one of the tasks, while the instructor placed the cal to the system. After, the dialogue completion the subject was asked if the dialogue was successful or not.

A total of 511 dialogues were collected for both conditions. The dialogue success rate was 67.1% (79.6% with the fan off, and 55.5% with the fan on). The collected speech data was automatically transcribed using Amazon Mechanical Turk crowd sourcing (Jurčíček et al 2011). The total audio length of all the dialogues was 2.2 hours (user turns only). The resulting speech database was further split into a training set (~60%) and a test set (~40%). This was done at the speaker level; the word error rate (WER) per speaker was measured from the transcriptions, and then each speaker was assigned to either the train or the test set. The distributions of WERs per speaker, as well as the ratio of male to female speakers were kept the same in both train and test sets. The training set was used to adapt the ASR acoustic model. This resulted in an absolute WER reduction of about 9% (from 25.8% to 16.9%) for the in-car test data set. The LM was not updated since the available data was not sufficient for language model training.

In the second stage of the experiment we trained a POMDP dialogue system and evaluated its performance in comparison with a hand-coded one based on MDP and optimized for the restaurant info task. Henceforth, the two systems are termed as:

- *MDP,* hand-coded dialogue manager based on an MDP architecture
- *POMDP,* trained POMDP dialogue manager

Both systems used the same Phoenix-based semantic decoder and the same retrained speech recognition system.

The experimental setup was similar to the one used for data collection, while taking place in a car driven around the city of Cambridge by a professional driver. The subject sat in the front seat and the experiment instructor in the rear seat. Each subject was asked to complete 14 dialogue tasks (7 for each of the two systems). After the completion of each dialogue, the subject reported to the instructor if he or she considered the task successful or not. A total of 30 subjects took part in the experiment and a total of 399 dialogues were collected. Some of the subjects did not complete the whole set of tasks due to technical difficulties. Table 4 summarises the subjective performance for each of the dialogue systems. For each system, the table shows the total number of dialogues, the number of successful dialogues, the estimated subjective success rates, and the word error rate (WER). The standard error on the success rates was estimated under the assumption that the success rates follow a binomial distribution (Thomson and Young, 2010). The actual word error rates may be lower than the ones reported here since the crowd sourcing transcription process usually introduces additional word errors.

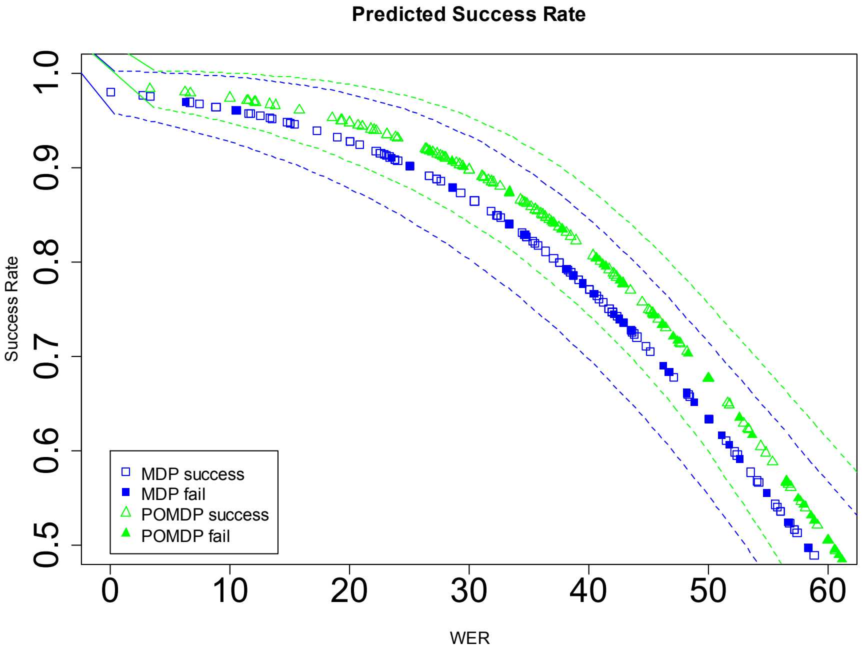

Figure 3 The effect of the WER factor on success rates of the MDP and POMDP systems.

Table 4 Subjective evaluation results

System	*# Dialogoues*	*# Successful*	*% Success Subjective*	*WER*
MDP	200	130	65.00 ± 3.37	46.83
POMDP	199	141	**70.85 ± 3.22**	43.93

We performed analysis of variance to test the statistical significance of the results. The test shows that the difference in the success rate of the two systems is not statistically significant at the 95% confidence level [p-value = 0.20]. We also tested the statistical significance of the WER factor, which also showed that there is no statistically significant difference between the word error rates of the two systems [p=0.44]. To have a more accurate depiction of the WER factor on the performance of the systems, figure 3 shows the logistic regression of the probability of success against the word error rate for a particular dialogue. The plotted points show the true WER of a given dialogue along with the predicted probability of success according to the logistic regression model. Unfilled markers depict successful dialogues, while filled ones depict unsuccessful dialogues. The dotted lines show one standard error on each side of the predicted probability of success.

The POMDP system has a higher success rate on average. Although this difference is not statistically significant, the results strongly suggest that the POMDP system can cope better with ASR errors. We can see, for example, that the MPD system has some failure points even when the speech recognition performs reasonably well (WER<15%), while, on the other hand, the POMDP system starts to break at WER above 25%. This suggests that the improvement in the performance of the POMDP system is not due to better speech recognition performance but rather due to a more robust dialogue manager.

5 CONCLUSIONS

We have investigated a statistical approach to dialogue management for in-car speech-based information systems with the aim of increasing robustness to recognition errors. A Restaurant Information task was selected for this pilot study. A statistical dialogue manager based on POMDPs was trained and evaluated in comparison with a non-statistical one optimized manually for the same task. The evaluation took place in a realistic scenario where human subjects were driven around in an urban environment while talking to the dialogue system in an open microphone manner. Moreover, the system was integrated with an online service providing up to date restaurant information. Although the corpus of 400 dialogues collected for the evaluation is too small to give statistical significance, the results did indicate that the statistical dialogue manager has a higher success rate and is more robust to speech recognition errors. This suggests that in noisy situations, such as the automotive environment, where speech recognition is prone to errors, the statistical approach to dialogue management can improve overall performance.

ACKNOWLEDGMENTS

This work was funded in part by General Motors in the framework of the GM project “Robust Dialogue for Infotainment” (GM Project Number: NV887).

REFERENCES

Jurčíček, F., Keizer, S., Gašić, M., Mairesse, F., Thomson, B., Yu, K., & Young, S. (2011) “Real user evaluation of spoken dialogue systems using Amazon Mechanical Turk”. Proc. Interspeech, Florence, Italy.

Mairesse, F., Gasic, M., Jurcicek, F., Keizer, S., Thomson, B., Yu, K., & Young, S. (2009). “Spoken language understanding from unaligned data using discriminative classification models”. Proc. ICASSP, Taipei, Taiwan.

Thomson, B. & Young, S. (2010)“Bayesian Update of Dialogue State: A POMDP framework for spoken dialogue systems.” Computer Speech and Language 24(4):562-588.

Ward, W.H., (1991), “The Phoenix system: Understanding spontaneous speech.” Proc. ICASSP, Toronto, Canada.

Williams, J. & Young, S. (2007). "Partially Observable Markov Decision Processes for Spoken Dialogue Systems." Computer Speech and Language 21(2): 393-422.

Yu, K., Mairesse, F. & Young, S. (2010). "Word-level Emphasis Modelling in HMM-based Speech Synthesis." Proc. ICASSP, Dallas, TX.