# Probabilistic Dialogue Modeling for Speech-Enabled Assistive Technology

*William Li, Jim Glass, Nicholas Roy, Seth Teller*

MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA 02139, U.S.A.
{wli, glass, nickroy, teller}@csail.mit.edu

## Abstract

People with motor disabilities often face substantial challenges using interfaces designed for manual interaction. Although such obstacles might be partially alleviated by automatic speech recognition, these individuals may also have co-occurring speech-language challenges that result in high recognition error rates. In this paper, we investigate how augmenting speech applications with dialogue interaction can improve system performance among such users. We construct an end-to-end spoken dialogue system for our target users, adult wheelchair users with multiple sclerosis and other progressive neurological conditions in a specialized-care residence, to access information and communication services through speech. We use boosting to discriminatively learn meaningful confidence scores and ask confirmation questions within a partially observable Markov decision process (POMDP) framework. Among our target users, the POMDP dialogue manager significantly increased the number of successfully completed dialogues (out of 20 dialogue tasks) compared to a baseline threshold-based strategy ($p = 0.02$). The reduction in dialogue completion times was more pronounced among speakers with higher error rates, illustrating the benefits of probabilistic dialogue modeling for our target population.

**Index Terms**: spoken dialogue systems, speech interfaces, POMDPs

## 1. Introduction

People with mobility or physical impairments may have difficulty with touch-based user interfaces. Automatic speech recognition (ASR) potentially offers an alternative, natural means of device access, but such systems can still be challenging to use for individuals who have speech impediments or disorders. For example, mismatches may exist between their speech and that of trained ASR systems. These technical challenges mean that ASR often fall short of its potential as an access equalizer for people with disabilities [1].

Current approaches to recognizing speakers with disabilities often use speaker adaptation techniques [2, 3]. Such training, however, may be costly, tiring, and difficult for the speaker. As well, in some real-world systems, it may not be possible to access or adapt the underlying acoustic models. Meanwhile, in many assistive technology applications, such as device control or information access, the success metric may not be the word error rate, but rather whether the system successfully understands the user's intent and ultimately responds correctly. Motivated by this abstraction, and faced with highly challenging speech, we seek to construct a system that optimizes performance at the user intent level.

The present paper describes the use of probabilistic dialogue modeling for a population of speakers with high recognition error rates. Specifically, we developed an assistive spoken dialogue system in a partially observable Markov decision process (POMDP) framework, in which the the dialogue system seeks to infer the user's intent and handles speech recognition uncertainty by asking confirmation questions. We learn models of: 1) how speech recognition hypotheses map to user intents and 2) meaningful confidence scores from ASR features so that our dialogue manager can make better response decisions. Our work draws on modeling techniques from work in spoken dialogue system POMDPs (e.g., [4, 5, 6]) and is inspired by other POMDP-based assistive technologies for handwashing (e.g., [7]) and intelligent wheelchair navigation (e.g., [8, 9, 10]), all of which model the user's intent as a hidden state to be inferred from observations.

Our work has two main contributions. First, we defined and modeled our problem as a spoken dialogue system POMDP by understanding our users and the design constraints. We collected data specifically for this application, trained the probabilistic models that are part of the dialogue system, and made design decisions appropriate to our application, all of which we describe in this paper. Second, we conducted experiments involving speakers with disabilities that demonstrated the effectiveness of the POMDP framework under high-error conditions. As illustrated in our results, handling uncertainty with the POMDP-based dialogue manager led to higher dialogue completion rates and shorter dialogue times, particularly for users with high speech recognition error rates.

This paper is structured as follows: We describe our assistive technology application domain and our target user population (Section 2), the formulation of our POMDP-based spoken dialogue system (SDS-POMDP) (Section 3), our model-building efforts (Section 4), and the experiments designed to test the effectiveness of our end-to-end system (Section 5). We conclude with insights on using dialogue interaction for assistive technology.

## 2. Problem Domain

Our target population is the residents at The Boston Home (TBH), a specialized-care residence in Boston, Massachusetts, USA for adults with multiple sclerosis (MS) and other progressive neurological conditions, and our goal is to develop speech-enabled assistive technology that can be bedside or wheelchair accessible. One example physical setup for a resident is shown in Figure 1. MS and other related neurological conditions are often associated with co-occurring speech pathologies, including rapid fatigue, voice weakness, very slow speaking style, or mild to severe dysarthria [11]. In addition, cognitive impairments associated with MS can also lead to language disorders [12], which could challenge conventional ASR language models.

Table 1 illustrates the performance of our ASR system on on 30 utterances for members of our target population. All of



*SLPAT 2013, 4th Workshop on Speech and Language Processing for Assistive Technologies*, pages 67–72,
Grenoble, France, 21–22 August, 2013. 

Figure 1: Example bedside setup of speech interface in resident room at TBH.

the utterances were processed by the MIT SUMMIT speech recognizer [13] using the same set of acoustic and language models. Our target users were seven adult residents at TBH (5 male, 2 female, ages 45 to 70), all of whom use wheelchairs and expressed an interest in using a speech recognition-based system. More precisely, our metric of interest is whether the speech recognition hypothesis maps to the user's intent — an utterance is labeled "correct" if its top hypothesis and its ground-truth label map to the same intent in our dialogue system. For example, if the utterance "what is monday's lunch menu" is hypothesized as "what is monday the lunch", this utterance would be marked as correct because both the hypothesis and the label correspond to the intent `(lunch monday)`.

Table 1 also shows the performance of the speech recognizer for a control group of seven students (6 male, 1 female, ages 21 to 32) without speech impairments of any kind. The target and control users are not paired in any way; our main reason for showing the system performance with these control users is to provide a quantitative sense of how the speech of our target users is handled conventional ASR systems. In addition, by evaluating our system with both target and control users in our dialogue system, as we show in Section 5, we can compare the value of using dialogue among high- and low-error speakers.

Table 1: Concept error rates (30 utterances) for target and control populations

| Speaker (Target) | Intent Error Rate | | Speaker (Control) | Intent Error Rate |
|---|---|---|---|---|
| target01 | 13.3% | | control01 | 3.3% |
| target02 | 3.3% | | control02 | 10.0% |
| target03 | 33.3% | | control03 | 6.7% |
| target04 | 56.7% | | control04 | 13.3% |
| target05 | 26.7% | | control05 | 3.3% |
| target06 | 9.4% | | control06 | 3.3% |
| target07 | 6.6% | | control07 | 0.0% |
| mean | 21.4% | | mean | 7.5% |
| std. dev. | 18.9% | | std. dev. | 4.3% |

Clearly, the target group of users has a much higher error rate, meaning that a system that simply parses the top hypothesis would be unusable for many target users. This research hypothesizes that dialog strategies that consider the uncertainty associated with user utterances can enable higher task completion rates, particularly for speakers with high speech recognition error rates. The system should handle ASR errors robustly, with the aim of deciphering the user's intent in order to respond appropriately.

## 3. Partially Observable Markov Decision Processes (POMDPs) for Spoken Dialog

Substantial research exists on modeling spoken dialogue as a partially observable Markov decision process (POMDP) [4, 5, 6]. Briefly, a POMDP is specified as a tuple $\{S, A, Z, T, \Omega, R, \gamma\}$ and is a sequential decision model that handles uncertainty in the environment in a principled way. A POMDP spoken dialogue system (SDS-POMDP) treats speech recognition results as noisy observations of the user's intent: it encodes the user's intent as a hidden state, $s \in S$; automatic speech recognition hypotheses as observations, $z \in Z$, of that state; and system responses as actions, $a \in A$. The transition model $T = P(s'|s, a)$ gives the probability that the user's intent will change to $s'$ given the previous intent $s$ and the system action $a$; the observation model $\Omega = P(z|s, a)$ describes the probability of ASR observation $z$ for a given intent $s$ and action $a$; and $R(s, a)$ specifies the immediate reward associated with each system action $a$ and user intent $s$. The discount factor $\gamma$ is a parameter ($0 \leq \gamma \leq 1$) that weighs the value of future rewards to immediate rewards.

Bayesian filtering is used to infer a distribution over the user's state at each time step t from the history of actions and observations, $p(s_t|a_{0:t}, z_{0:t})$ [14]. This distribution is usually referred to as the belief, $b$. The SDS-POMDP maintains the belief distribution, $b$, over the user's possible intents and chooses actions based on a policy, $\Pi(b)$, that maps every possible belief to an action, $a$, in order to maximize the expected discounted reward, $\sum_t \gamma^{-t} R(s, a)$. We describe the key elements of the SDS-POMDP in the context of the system that we developed for our experiments below.

### 3.1. SDS-POMDP System Implementation

**User Goals (States, $S$) and System Responses (Actions, $A$)**: When a user interacts with the dialogue manager, we assume that he or she has a goal, $s \in S$. The purpose of the dialogue manager is to choose an action, $a \in A$, that satisfies the user's goal. More precisely, the dialogue manager seeks to infer which goal the user is trying to achieve and take an appropriate action.

For our system, we identified the following areas of interest to residents at TBH:

- Time and date;
- Recreational activities schedules;
- Breakfast, lunch, and dinner menus;
- Making phone calls.

Our SDS-POMDP has 62 states, corresponding to each of the possible user goals. For example, `(weather today)` or `(make phone call)` are two different states.

The definition of the action space, $A$, follows from the set of states. For every state, there are two corresponding action: one that asks the user for confirmation, and the other "executes" that goal in the SDS-POMDP's user interface. For example, the state `(weather today)` has two corresponding actions: `(confirm, (weather today))`

and `(show, (weather today))`. In addition, the SDS-POMDP can greet the user or ask the user to repeat, for a set of 126 system actions.

**ASR Outputs (Observations, $Z$)**: The SDS-POMDP uses the aforementioned MIT SUMMIT speech recognizer [13]. Each spoken utterance is processed into a ten-best list of hypotheses with acoustic and language model scores. We then extract keywords to deterministically map the top hypothesis into one of 65 concepts: observations corresponding to each of the 62 goals (such as `( weather today )` and `(lunch monday)`), a `( yes )` and `( no )` command, and a `( null )` command if there is no successful parse. Meanwhile, the text of the ten hypotheses for each utterance, along with the acoustic and language scores for each utterance computed by the speech recognizer, are used as features to assign a confidence score to the hypothesis, as detailed in Section 4. An observation $z$ in the SDS-POMDP, therefore, consist of a discrete part, $z_d$ (one of 65 possible parses) and a continuous confidence score, $z_c$ (where $0 \leq z_c \leq 1$).

**Observation Model ($\Omega$)**: $\Omega = P(z|s,a)$ is our model, learned from data, of recognition hypotheses given the user's intent, $s$, and the system's response, $a$. As described above, our observations consist of a discrete ($z_d$) and a continuous ($z_c$) part, meaning that we need to learn the model $P(z_d, z_c|s,a)$. We factor the observation function into two parts as per Equation 1 using the chain rule:

$$\Omega = P(z_d, z_c|s,a) = P(z_d|s,a)P(z_c|s,a,z_d) \quad (1)$$

The first term, $P(z_d|s,a)$ is estimated from our labeled data using maximum likelihood; for each discrete observation $z_d^*$, the value $P(z_d^*|s,a)$ is computed as follows:

$$P(z_d^*|s,a) = \frac{c(z_d^*, s, a)}{\sum_{z_d} c(z_d, s, a)} \quad (2)$$

Meanwhile, for the term $P(z_c|s,a,z_d)$, data sparsity makes it challenging to directly learn the model of confidence score for every $(s,a,z_d)$-triple. To mitigate this issue, we use an approximation similar to the one used by [15], where we learn two models: 1) the distribution of confidence scores when the utterance hypothesis is correct ($P(z_c|$correct observation$)$), and 2) the distribution of confidence scores when there is an error ($P(z_c|$incorrect observation$)$). The motivation for this approach is that correctly recognized utterances should have a different distribution of confidence scores than incorrectly recognized utterances. In addition, an equivalent statement to the observation being correct is that that $z_d$ *corresponds to* $s$ (denoted below as $z_d \mapsto s$). As a result, for all possible user goals $s$ and discrete observations $z_d$, we can approximate $P(z_c|s,a,z_d)$ as follows:

$$P(z_c|s,a,z_d) = \begin{cases} P(z_c|\text{correct observation}) & \text{if } z_d \mapsto s \\ P(z_c|\text{incorrect observation}) & \text{otherwise} \end{cases} \quad (3)$$

We describe our efforts to learn the confidence score model from our data in Section 4. Figure 3 illustrates that, indeed, the distributions of $P(z_c|$correct observation$)$ and $P(z_c|$incorrect observation$)$ are different in our dataset. These two models capture the insight that the confidence score contains information about whether the utterance has been correctly or incorrectly recognized. By assuming that the distribution of confidence scores for correct and incorrect observations are the same for every concept, our approach helps overcome data sparsity issues.

**Transition Model, T**: For our prototype system, our transition function $T = P(s'|s,a)$ is simple: we assume that that the user's goal does not change over the course of a single dialog, meaning that the transition function equals 1 if $s_{n+1} = s_n$ and 0 otherwise.

**Reward Function, R**: The reward function specifies a positive or negative reward for each state-action pair in the SDS-POMDP; as a result, it is described by as $R(S,A)$. We handcrafted a reward function that has positive rewards for "correct" actions (e.g. showing the user the weather if the user's goal was to know the weather), large negative rewards for "incorrect actions" (e.g. making a phone call if the user's goal was to know the lunch menu), and small negative rewards for information-gathering confirmation questions. The reward for confirmation questions that do not correspond to the user's goal is slightly more negative than for the "correct" confirmation question.

**Belief Updates**: Over the course of a dialog, our SDS-POMDP updates the belief distribution, $b$, from the observed hypothesis, the observed confidence score, and the transition function, $T = P(s'|s,a)$. At time step $n+1$, the SDS-POMDP uses these models and the prior belief, $b_n$, to compute $b_{n+1}$:

$$b_{n+1}(s') \propto P(z_d|s',a)P(z_c|s',a,z_d) \sum_s P(s'|s,a)b_n(s) \quad (4)$$

During runtime, the SDS-POMDP does not have access to the ground-truth label of the user's utterance. For each state $s'$, the terms $P(z_d|s',a)$ and $P(z_c|s',a,z_d)$ are chosen from the appropriate conditional probability distribution in Equations 2 and 3, respectively.

**Computing the Policy, $\Pi$**: The policy, which maps beliefs to actions, is computed offline from the specified models in the SDS-POMDP. Given how we incorporate the continuous confidence score $z_c$ into the observation function $\Omega$, conventional methods of computing the POMDP policy are computationally expensive. We chose the QMDP approximation to compute the policy for the SDS-POMDP. While QMDP is a greedy heuristic, as opposed to an optimal POMDP solution, we hypothesized that it could produce an effective dialogue policy in our work. Specifically, the QMDP algorithm computes a function $Q$ for each state-action pair,

$$Q(s_i,a) = R(s_i,a) + \sum_{j=1}^{N} \hat{V}(s_i)P(s_j|s_i,a) \quad (5)$$

where $\hat{V}$ is the converged value function of the SDS-POMDP's underlying Markov decision process (MDP) [16]. Then, for a belief state $b = (p_1, p_2, ...p_N)$, where $p_i$ corresponds to the probability mass in state $i$, the policy is simply

$$\Pi(b) = \arg\max_a \sum_{i=1}^{N} p_i Q(s_i,a) \quad (6)$$

It is impractical to describe the policy's prescribed action for every possible $b$ in our system, but a few representative belief points and corresponding actions are:

1. if $b$ is uniform, then the dialogue system asks the user to repeat;
2. if $b$ has very high probability in one state, $s^*$, and the remainder of the probability mass is uniformly distributed in the other states, then the dialogue system takes the terminal action corresponding to that state;

3. between situations 1 and 2, i.e. if the probability mass in $s^*$ is not high enough for the system to perform the terminal action, then it will ask a confirmation question corresponding to $s^*$.

**User Interface**: Finally, the user interface for the SDS-POMDP is presented to the user on a netbook computer. In our current implementation, the speech recognizer is run locally. A screenshot of the interface is shown in Figure 2.

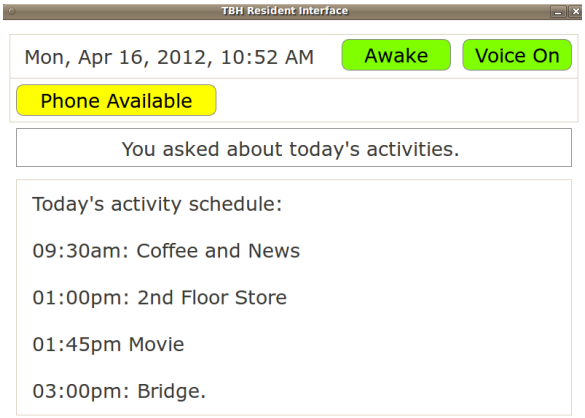


Figure 2: Graphical user interface of assistive spoken dialogue system, with indicators of time, system state ("Awake"), and speech synthesizer state ("Voice On").

## 4. Model Training and Confidence Scoring

**Data Collection**: A total of 2701 utterances were collected and manually transcribed from volunteers in our research lab and at TBH. Participants were prompted with possible goals and asked to speak a natural-language command corresponding to the goal, prefaced by an activation keyword like "chair" or "wheelchair." Because our target population has difficulty using buttons or other physical access devices, a speech-activity detector based on the measured spectral power of the audio signal was used instead of a push-to-talk activation method typical in many speech applications. This corpus of utterances was used to estimate the discrete and continuous parts of the observation model $\Omega$, as summarized in Equations 2 and 3.

**Learning the Confidence Score**: To learn the confidence score, $z_c$, each of the 2701 utterances was labeled as "correct" $(+1)$ if the parse of the top hypothesis matched the parse of the transcription and "incorrect" $(-1)$ otherwise. We then extracted features from each utterance's 10-best list and trained a classifier on 90% of the utterances using AdaBoost [17]. At each iteration, AdaBoost chooses a feature with the lowest weighted error, and re-weighs training data points by assigning more weight to misclassified examples; some of the features that it selected are shown in Table 2. Using this weighted set of features, the classification error rate on a held-out test set (10% of the utterances) was 6.9%.

Next, we fit a logistic regression curve to AdaBoost's weighted sum of features to interpret the AdaBoost classifier's result as a confidence score. The resulting distribution of confidence scores for correctly and incorrectly recognized utterances is shown in Figure 3. For a given confidence score $z_c$, we can compute the necessary quantities in Equation 3 from these two histograms. These two distributions reveal that the confidence score contains important information about whether the observed concept is correct or incorrect. During the belief update step of the SDS-POMDP, we draw from the "correct observation" distribution for the state corresponding to the observation concept and from the "incorrect observation" distribution for all other states. For example, the hypothesis `(lunch, today)` paired with a high confidence score could shift the belief distribution sharply toward the corresponding `(lunch, today)` state; in contrast, a low confidence score could actually cause the probability mass to shift away to other states.

Table 2: Features selected by AdaBoost classifier

| Feature Category | Examples |
|---|---|
| Concept-level | parse success; category of concept |
| ASR scores | acoustic, language, and total model scores; difference between top score and second-highest hypothesis score |
| Word-/ sentence-level | fraction of stop words; presence of multiple concepts; presence of highly misrecognized words or often merged/split word pairs |
| $n$-best list | concept entropy of $n$-best list; fraction of total acoustic or language model scores |

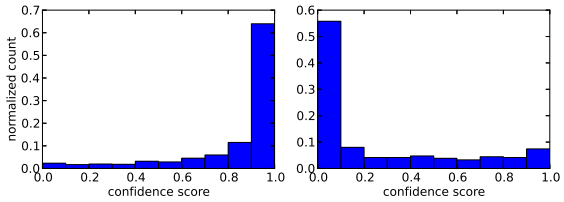


Figure 3: Distribution of confidence scores for correct $(P(z_c|\text{correct observation}))$ *(left)* and incorrect $(P(z_c|\text{incorrect observation}))$ *(right)* utterances.

## 5. SDS-POMDP Experiments

### 5.1. Experimental Design

We conducted a within-subjects study that compared our SDS-POMDP dialogue manager to a baseline threshold-based dialogue manager. In the SDS-POMDP, each dialogue began with a uniform distribution over states, the belief was updated according to Equation 4, and the system response was selected using the learned policy. In the threshold-based model, the confidence threshold was set at 0.75, where the system would ask the user to repeat if the threshold was not achieved.

The 14 individuals listed in Table 1 (seven "target" users and seven "control" users) participated in our experiments, which consisted of a single session for each user. In the session, the user was required to complete 40 dialogues. The 40 dialogues consisted of 20 goals, each presented once with the SDS-POMDP and once with the threshold-based dialogues manager. We randomized the ordering of the dialogues so that either the POMDP or baseline dialogue manager would be presented for a particular goal first. In addition, the same goal did not appear in consecutive dialogue tasks. Users were not told which dialogue

manager was in operation for a given task.

Although a threshold-based, memory-less baseline dialogue manager is simple, we chose it as our point of comparison because it represents the current approach used by many existing speech-enabled assistive technologies. Such a system could potentially have advantages over the SDS-POMDP; for instance, there is no risk that belief probability mass would accumulate in incorrect states and require the user to speak additional utterances to correct errors. Meanwhile, it might have been useful to try to learn a optimal threshold, conduct experiments with different threshold-based dialogue managers, or evaluate the POMDP-based system with dialogue management strategies. However, because 40 dialogues already took substantial effort for some of our target users to complete, we did not perform these additional points of comparison.

Each of the dialogue tasks was presented with a text prompt on our graphical user interface, similar to the one shown in Figure 2. Our evaluation metrics were 1) the total number of dialogues (out of 20) completed within 60 seconds and 2) the total duration of the dialog, from the start of the user's first utterance until the system executed the correct response.

### 5.2. Results

All seven control users were able to complete all 20 dialogues successfully within 60 seconds. In contrast, as shown in Table 3, the seven target users completed an average of 17.4 out of 20 dialogues successfully with the SDS-POMDP and 13.1 with the threshold-based dialogue manager. A one-way repeated-measures ANOVA indicates a significant effect of the SDS-POMDP on the number of dialogues completed within sixty seconds ($F(1,6)=10.23$, $p=.02$), compared to the threshold-based model.

Table 3: Number of completed dialogues by target population users by dialogue manager

| User | SDS-POMDP (/20) | Threshold (/20) |
|---|---|---|
| target01 | 18 | 13 |
| target02 | 17 | 16 |
| target03 | 20 | 20 |
| target04 | 19 | 18 |
| target05 | 13 | 5 |
| target06 | 18 | 10 |
| target07 | 17 | 10 |
| average | $17.4 \pm 0.9$ | $13.1 \pm 0.9$ |

In terms of dialogue completion times, the performance of the threshold-based and POMDP-based dialogue managers for all 14 participants is shown in Figure 4. In the case of unsuccessful dialogues, we assume that the total time elapsed was 60 seconds to compute the values in Figure 4.

## 6. Discussion

### 6.1. Analysis of Results

The results in Table 3 show that the target population users benefited considerably from the POMDP-based dialogue manager. In general, this improvement was due to users being able to achieve the dialogue goal after a few low-confidence utterances in the SDS-POMDP; in contrast, they were unable to generate a correct utterance above the confidence threshold in the required time.

Figure 4 illustrates that the largest improvements, in terms

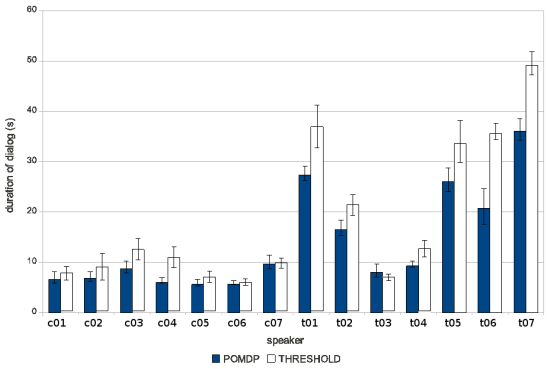


Figure 4: Dialog durations for POMDP- and threshold-based dialogue systems for control (c01-c07) and target (t01-t07) users. Error bars show standard error of the mean.

of time saved, were among users with the highest completion times with the baseline system. These users were able to complete dialogues in less time using the SDS-POMDP. This trend underscores the benefit of probabilistic dialog management in handling noisy speech recognition inputs: the SDS-POMDP performs just as well as simpler, threshold-based methods for speakers with low ASR error rates (*i.e.* the control participants), but as the uncertainty increases among users with more ASR errors, the SDS-POMDP becomes superior.

The key advantage of the SDS-POMDP over the baseline was that it acquired information about the user's intent from every utterance. The top recognition hypothesis and the confidence score updated the SDS-POMDP's belief. In cases where there was a speech recognition error, it was likely that some probability mass was allocated to the user's actual goal. As well, utterances with speech recognition errors were more likely to have lower confidence scores, resulting in less "peaked" updates to the belief. This behavior meant that probability mass was not incorrectly allocated to the goal corresponding to the incorrect hypothesis. For these reasons, over the course of multiple dialogues, the SDS-POMDP's belief update operation made it superior to the threshold-based dialogue manager.

## 7. Conclusion

This paper offers empirical evidence that probabilistic dialog modeling, particularly the use of confidence scoring and confirmation questions in a POMDP framework, could enhance the effectiveness of spoken dialogue systems among users with high ASR error rates. By asking confirmation questions, a system can become more confident about taking the right action or avoid taking incorrect actions. Such methods could be useful for deploying speech-enabled assistive technology among users with challenging speech characteristics or in other situations where error-prone speech recognition is expected.

## 8. Acknowledgments

We thank Don Fredette and Alexander Burnham for their advice and guidance at The Boston Home.

## 9. References

[1] V. Young and A. Mihailidis, "Difficulties in automatic speech recognition of dysarthric speakers and implications for speech-

based applications used by the elderly: A literature review," *Assistive Technology*, vol. 22, no. 2, pp. 99–112, 2010.

[2] F. Rudzicz, "Comparing speaker-dependent and speaker-adaptive acoustic models for recognizing dysarthric speech," in *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*. ACM, 2007, pp. 255–256.

[3] H. V. Sharma and M. Hasegawa-Johnson, "Acoustic model adaptation using in-domain background models for dysarthric speech recognition," *Computer Speech & Language*, 2012.

[4] N. Roy, J. Pineau, and S. Thrun, "Spoken dialogue management using probabilistic reasoning," in *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2000, pp. 93–100.

[5] 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.

[6] S. Young, M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu, "The hidden information state model: A practical framework for pomdp-based spoken dialogue management," *Computer Speech & Language*, vol. 24, no. 2, pp. 150–174, 2010.

[7] J. Hoey, A. Von Bertoldi, P. Poupart, and A. Mihailidis, "Assisting persons with dementia during handwashing using a partially observable markov decision process," in *Proc. Int. Conf. on Vision Systems*, vol. 65, 2007, p. 66.

[8] J. Pineau and A. Atrash, "Smartwheeler: A robotic wheelchair test-bed for investigating new models of human-robot interaction," in *AAAI spring symposium on multidisciplinary collaboration for socially assistive robotics*, 2007, pp. 59–64.

[9] T. Taha, J. V. Miró, and G. Dissanayake, "Pomdp-based long-term user intention prediction for wheelchair navigation," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE, 2008, pp. 3920–3925.

[10] P. Viswanathan, J. J. Little, A. K. Mackworth, and A. Mihailidis, "Navigation and obstacle avoidance help (noah) for older adults with cognitive impairment: a pilot study," in *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*. ACM, 2011, pp. 43–50.

[11] L. Hartelius, B. r. Runmarker, and O. Andersen, "Prevalence and characteristics of dysarthria in a multiple-sclerosis incidence cohort: relation to neurological data," *Folia phoniatrica et logopaedica*, vol. 52, no. 4, pp. 160–177, 2000.

[12] G. Arrondo, J. Sepulcre, B. Duque, J. Toledo, and P. Villoslada, "Narrative speech is impaired in multiple sclerosis," *European Neurological Journal*, vol. 2, no. 1, pp. 11–8, 2010.

[13] J. Glass, "A probabilistic framework for segment-based speech recognition," *Computer Speech and Language*, vol. 17, no. 2-3, pp. 137–152, 2003.

[14] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[15] J. Williams, "Partially observable markov decision processes with continuous observations for dialogue management," in *Computer Speech and Language*, 2005, pp. 393–422.

[16] M. Littman, A. Cassandra, and L. Kaelbling, "Learning policies for partially observable environments: Scaling up," *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 362–370, 1995.

[17] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Proceedings of the Second European Conference on Computational Learning Theory (EuroCOLT)*. London, UK: Springer-Verlag, 1995, pp. 23–37.