Finite-to-Infinite N-Best POMDP for Spoken Dialogue Management

Guohua Wu, Caixia Yuan, Bing Leng, and Xiaojie Wang

School of Computer,
Beijing University of Posts and Telecommunications, Beijing, China
{wugh,yuancx,lb19900314,xjwang}@bupt.edu.cn

Abstract. Partially Observable Markov Decision Process (POMDP) has been widely used as dialogue management in slot-filling Spoken Dialogue System (SDS). But there are still lots of open problems. The contribution of this paper lies in two aspects. Firstly, the observation probability of POMDP is estimated from the N-Best list of Automatic Speech Recognition (ASR) rather than the top one. This modification gives SDS a chance to address the uncertainty of ASR. Secondly, a dynamic binding technique is proposed for slots with infinite values so as to deal with uncertainty of talking object. The proposed methods have been implemented on a teach-and-learn spoken dialogue system. Experimental results show that performance of system improves significantly by introducing the proposed methods.

Keywords: Partially Observable Markov Decision Process (POMDP); Spoken Dialogue System (SDS); Dynamic Binding; N-Best

1 Introduction

Spoken dialogue system (SDS) is to provide an interface for human to access and manage information. Since the 1990s, many SDSs have been developed. For example, flight reservation system (Mercury) [8], weather information system (JUPITER) [14]. More recently developed systems are used as virtual assistants. For example, SDS for interactive search (PARLANCE) [2], Apple's intelligent personal assistant and knowledge navigator (Siri) and in-car voice assistant of Nuance (Dragon Drive).

Among all these dialogue systems, there is a kind of dialogue systems designed for retrieving specific important information. For instance, flight reservation system is such kind of dialogue system whose objective is to obtain all the information essential for flight-ticket ordering such as date, flight number and destination. Generally such kind of essential information is called slot and such kind of dialogue system is known as slots-filling based dialogue system.

Generally spoken dialog systems have a common logical architecture of three modules: spoken language understanding (SLU), dialogue management (DM) and natural language generation (NLG). Of all three modules, dialogue management is the core module, which controls the dialog flow.

According to the method employed, DMs can be divided into finite state-based, frame-based, agent-based and statistical based. In finite state-based DMs [3, 7], the user is taken through a dialogue consisting of a sequence of predetermined steps or states. This approach is suitable for well-structured tasks, but it can't deal with the uncertainty introduced by SLU. In frame-based DMs [3, 7], user can communicate with dialogue system in a more flexible way and uncertainty introduced by SLU can be disposed to some extent at the cost of more elaborate control in algorithm. In agent-based DMs [3, 7], communication is viewed as interaction between two agents, each of which is capable of reasoning about its own actions and beliefs, and sometimes also about the actions and beliefs of the other agent. Construction of such system requires lots of expert knowledge. Statistical DMs describe a dialogue as a probability decision procedure of which POMDP [4, 9] is one of the most representative models. POMDP-based DMs [6, 11, 10, 13, 12] model dialogue as a decision procedure under uncertainty due to errors in speech recognition and SLU. Parameters and policy of POMDP-based DMs is estimated from data. Thus, it can reduce the development cycle and improve domain portability.

POMDP-based DMs suffer from two principal problems. On the one hand, due to errors of speech recognition and semantic understanding, the observation attained by dialogue management is unreliable. Thus, how we estimate probability distribution of observation in POMDP will severely affect POMDP's performance. Williams et al. [11, 10] proposed SDS-POMDP model factorized hidden state of dialog into tree quantities, which include user's goal, user action and dialogue history. However, SDS-POMDP doesn't make use of the entire N-Best output of the automatic speech recognition (ASR).

On the other hand, POMDP-based DMs have the limitation that the number of slot value should be finite. For example, it can be applied to the ticket-ordering task where the possible values for date, flight number and destination are all finite. However, many tasks in real world can't meet this limitation. Taking the teach-and-learn task for example, even if objects mentioned in this task have fix-sized slots, the possible values of slots are infinite because of infinity of objects.

In this paper, we propose dynamic object binding method which can be applied to tasks whose fixed-size slots have infinite possible values. In addition, a method to estimate observation probability of POMDP by making use of entire N-Best list of ASR is presented. We conduct a series of experiments on a teach-and-learn dialogue system. Results of experiments indicate that the proposed model can help dialogue system accomplish teach-and-learn tasks with infinite objects with better performance.

The paper is organized as follows. In Section 2, we discuss how to make use of N-Best list of ASR to improve POMDP's observation function and propose a dynamic object binding method to solve certain case of infinite slot values problem. Furthermore, we introduce our improved POMDP-based DM model with its algorithm based on improvements mentioned above. Section 3 presents experimental setup and results. Section 4 concludes.

2 Model

The proposed model is based on SDS-POMDP, thus, we will first introduce original POMDP model and its application in DM known as SDS-POMDP model before introducing our model.

2.1 SDS-POMDP Model

A POMDP can be formally defined as a tuple $\{S, A, T, R, O, Z\}$. S is a set of world states. A is a set of actions the agent may take. T is a state transfer function $P(s'|s, a)$, which indicates the probability that the agent takes action a in state s the world ends in state s'. R is a reward function $R(s, a) \in \mathbb{R}$, which is the reward that the agent obtains when it takes action a in state s. O is a set of observation the agent may obtain from outer world. Z is an observation function $P(o'|s', a)$ which indicates the probability of obtaining observation o when the agent takes action a and lands in state s'.

The POMDP operates as follows. At each time-step, the world is at some state $s \in S$. Since s is not known exactly, the agent is maintaining a belief state b which is a distribution over states, with initial belief state b_0. $b(s)$ indicates the probability of being in a particular state s. Based on b and POMDP policy π (policy indicates a map from belief state space to action space $\pi(b) \in A$), the agent selects an action a, receives a reward $R(s, a)$, and makes the world transmit to state s'. The agent then receives an observation o' depending on s' and a, and updates its belief state b as follows:

$$b'(s') = p(s'|o', a, b) = \frac{p(o'|s', a, b)p(s'|a, b)}{p(o'|a, b)} = k \cdot p(o'|s', a) \sum_{s \in S} p(s'|a, s)b(s) \quad (1)$$

Williams et al. [11, 10] cast spoken dialog as a POMDP. The system's dialogue act is formulated as the POMDP's action a and the output of ASR/SLU is regarded as POMDP observation o. Hidden state of dialogue is referred to information that system can't observe directly and can be cast as hidden POMDP state s. In the SDS-POMDP model, the hidden state contains three quantities $s = (s_u, a_u, s_d)$. s_u indicates user's long-term goal in dialogue. a_u indicates true, unobserved user action. s_d indicates history of dialogue which records aspects of hidden state which the dialog designer considers important.

We substitute $s = (s_u, a_u, s_d)$ into the POMDP belief update in Equation (1) and make some reasonable independence assumptions in [11], the SDS-POMDP belief update equation becomes

$$\begin{aligned} b'(s'_u, s'_d, a'_u) = k \cdot & \underbrace{p(o'|a'_u)}_{\text{observation model}} \cdot \underbrace{p(a'_u|s'_u, a_m)}_{\text{user action model}} \cdot \sum_{s_u \in S_u} \underbrace{p(s'_u|s_u, a_m)}_{\text{user goal model}} \\ & \cdot \sum_{s_d \in S_d} \underbrace{p(s'_d|a'_u, s'_u, s_d, a_m)}_{\text{dialog history model}} \cdot \sum_{a_u \in A_u} b(s_u, s_d, a_u) \end{aligned} \quad (2)$$

Equation (2) can be composited into observation model, user action model, user goal model and dialog history model.

2.2 The Improved SDS-POMDP Model

In this section, we will describe our improvement to SDS-POMDP. The improved model is superior to SDS-POMDP in two aspects. First, the observation probability of POMDP is estimated from entire N-Best list of ASR rather than the top one. Secondly, the improved model can be applied to slots with infinite values so as to deal with uncertainty of object. In the following sections, we will first discuss these two improvements and then outline a principled framework for model solution.

Observation Model Based on ASR N-Best List Considering that 1-best output of speech recognition isn't accurate enough, N-Best result of ASR is employed by DM instead. In this way, DM obtains more information with which it can guide the dialogue in a more appropriate direction when 1-best result of speech recognition is inaccurate. This method is somewhat similar to the way people make ambiguous information explicit as dialogue proceeds.

Assume that the N-Best list of ASR is:

$$[< hyp_1, conf_1 >, < hyp_2, conf_2 >, \cdots, < hyp_N, conf_N >]$$

Where hyp_i is i^{th} entry of N-Best list and $conf_i$ is corresponding confidence. A probability distribution over possible user actions is generated by SLU as follows:

$$\left[< \tilde{a}_u^1, p_1 >, < \tilde{a}_u^2, p_2 >, \ldots, < \tilde{a}_u^M, p_M >\right]$$

Where $p_m = P(a_u = \tilde{a}_u^m | hyp_i)$. Every one of these probabilities corresponds to a certain entry in N-Best list. The weighted sum of these N probability distributions is calculated according to corresponding confidence:

$$P(a_u = \tilde{a}_u^m | o) = \sum_{i \in N} P(a_u = \tilde{a}_u^m | hyp_i) \cdot conf_i$$

After normalization of the sum of N distributions, observation of DM is finally obtained:

$$o = \left[< \tilde{a}_u^1, p_1 >, < \tilde{a}_u^2, p_2 >, \ldots, < \tilde{a}_u^M, p_M >\right] \tag{3}$$

Here p_i is $P(a_u = \tilde{a}_u^i | o)$, but it's $P(o|a_u)$ required in belief update equation (2). Thus, probability provided in this list can't be applied to updating the belief state directly. A transformation based on Bayes' rule is applied to these to probabilities [10] as follows:

$$P(o|a_u) = \frac{P(a_u|o)P(o)}{P(a_u)} = k_1 \cdot \frac{P(a_u|o)}{P(a_u)} \approx k_2 \cdot P(a_u|o) \tag{4}$$

During belief update, $P(o)$ is constant, which means it can be absorbed into the normalization constant k_1. In addition, we assume $P(a_u)$ is uniform over all user actions, which make it can also be absorbed into the normalization constant k_2 in (4).

Dynamic Object Binding In slot-filling tasks based on SDS-POMDP, set of user goals S_u is Cartesian product of all slots, which needs to be defined in advance. Supposing that we need to fill N slots in a dialogue,$SLOT = \{slot_1, slot_2, \ldots, slot_n\}$ represents the finite set of slots, where the set of value for $slot_i$ is $V_i = \{v_{i,1}, v_{i,2}, \ldots, v_{i,n_i}\}$,which is a finite set for $i = 1, 2, \cdots n$. In the case mentioned above, S_u is a finite set, thus this dialogue can be well modeled by SDS-POMDP. However, SDS-POMDP can't be employed to tasks where $SLOT$ or any V_i is infinite. Therefore, some improvement is made to SDS-POMDP to extend it to tasks where $SLOT$ is finite and value set of V_j is infinite for some exact j. A typical example for this situation is the teach-and-learn task. For instance, in a dialogue concerning teach-and-learn , our goal is to teach the robot some object's class, color and shape. At this point, the goal of our task is to fill four slots of what (object), class, color and shape. We can assume that slots of class, color and shape are with finite possible values. For example, the slot of class may take four possible values from fruit, vegetable, meat and other. However, slot of what may have infinite values because the goal of the task is to teach these three attributes of any object to robot. What deserves attention is that here we can't avoid the demand for listing all the objects even if we have a simple assumption that slots have a value of other, because it's ridiculous that an object known as "other" appears in dialogue with attributes of class, color and shape. However, it's reasonable to assign value of "other" to the other slots, such as the class of the desk is "other".

First of all, a generalized assumption is made that if j^{th} slot with infinite values is given a certain value, values of other slots can be obtained with a similar POMDP policy. Take the teach-and-learn task for example. If slot of "what" has been obtained, system may obtain values for other three slots using a similar policy. In this way, system can learn other three slots for different teaching objects like apple, watermelon, tomato, basketball or pencil-box. Let set of values for slots with infinite values be $V_j = \{target, other\}$. Here *target* is a variable to be bind to some value during dialogue when values of other slots remain unchanged. Take the Cartesian product of all possible values of four attributes, we'll get the whole user goal space.

Secondly, when the DM finds out user's current action is to tell system the value of V_j in a dialogue, system needs to confirm value of V_j extracted from ASR results with user. After value of V_j is confirmed, we bind this value to variable *target*. Then an observation is constructed to update belief state. The rest of slots are obtained completely by similar POMDP policy.

We take the teach-and-learn task as an example to explain dynamic object binding. System performs language understanding when user inputs an utterance, which includes intention recognition and slot value extraction. Intention recognition is the process of being aware of the intentions of user. In the teach-and-learn task, intentions of user include teaching slot of what, slot of class, slot of color, slot of shape and other (refer to them as *teach-what*, *teach-class*, *teach-color*, *teach-shape* and *other*). Slot value extraction is the process of extracting teaching object, class, color and shape. For example, when user inputs

"这是一个柿子 (this is a persimmon.)", this utterance is processed by intention recognition and a distribution over intentions is obtained as follows:

$$[< \textit{teach-what}, 0.74 >, < \textit{teach-class}, 0.12 >, \ldots, < \textit{other}, 0.01 >]$$

From the distribution above, we can find out user's current action is to tell system the value of what, and system will extract slot values from user's utterance. In this example, slot value extraction gives the value of what is "柿子(persimmon)". Due to current value of *target* is empty, the probability of *teach-what* will be assigned to *teach-what-other*, we finally get the observation for DM:

$$o = [< \textit{teach-what-other}, 0.74 >, < \textit{teach-what-target}, 0.00 >, \ldots, < \textit{other}, 0.01 >]$$

Because *target* is empty, DM will confirm "柿子 (persimmon)" as value of "what" with user. After value of what is confirmed by user, we bind this value to variable *target*, and then an observation is constructed to update belief state:

$$o = [< \text{teach-what-target}, 0.9 >, < \text{teach-what-other}, 0.01 >, \cdots, < \text{other}, 0.01 >] \quad (5)$$

The rest of three slots is obtained completely by POMDP policy.

Algorithm In Algorithm 1, we give out algorithm for the improved SDS-POMDP, which is consistent to algorithm for SDS-POMDP. First, we need to initialize the belief state. Then system figures out what action should take at present according to POMDP policy π. After that, DM obtains observation from SLU. At last, dialogue system updates belief state according to current belief state and observation, and figures out what action should be taken in next round. Dialogue system will be executing above procedures until it takes the *submit* action.

Our modification to observation model is mainly reflected in line 6 and line 17 to 20. Line 6 mainly reflects how we get observation probability according to (3). Line 17 to 20 is mainly about how to update belief state using observation probability according to (2).

The procedure of dynamic binding is shown between lines 7 to 16. When POMDP-based DM observes *teach-what-other* with the highest probability in all the user actions, it'll extract the object that user's talking about and bind this object's name to the *target* variable. At the same time, POMDP will confirm whether the object user talking about in this round is exactly the value of target variable. If user gives the answer of yes, algorithm will construct observation in (5) and use it to update belief state. Or algorithm will empty variable of *target* and go into the next round without updating belief state.

Combining these two improvements, our algorithm for the improved SDS-POMDP is obtained. The improved model can make full use of information of N-Best list of ASR to produce more accurate belief state updates, and expands the application range of SDS-POMDP to situation where $SLOT$ is finite and value set of V_j is infinite for some exact j.

Algorithm 1 Algorithm for the improved SDS-POMDP

```
Require: Solving POMDP to find policy $\pi$
Initialize belief state $b$
$target \leftarrow None$    ▷ Doesn't know what slot currently
$a \leftarrow \pi(b)$
while $a \neq submit$ do
    Execute action $a$
    Observe $o' \leftarrow [< \tilde{a}_u^1, p_1 >, < \tilde{a}_u^2, p_2 >, \cdots, < \tilde{a}_u^M, p_M >]$
        where $p_m = P(a_u = \tilde{a}_u^m | o')$ and $p_1 \geq p_2 \cdots \geq p_M$
        from SLU module
    if $(\tilde{a}_u^1 = teach\text{-}what\text{-}other)$ and $(target = None)$ then
        ▷ Bind object
        $target \leftarrow$ teaching object extracted from user utterance
        $result \leftarrow$ confirm target value with user
        if $result = yes$ then    ▷ Positive confirm
            $o' \leftarrow [< \text{teach-what-target}, 0.9 >, <$
            $\text{teach-what-other}, 0.01 >, \cdots, < \text{other}, 0.01 >]$
        else
            $target \leftarrow None$    ▷ Reset target to unknown
            continue    ▷ Skip follow statements, jump to next loop
        end if
    end if
    for all $(s'_u, s'_d, a'_u)$ in $S_u \times S_d \times A_u$ do    ▷ Belief update
        ▷ Adopt approximation $P(o|a_u) \approx k_2 \cdot P(a_u|o)$
        Update $b'(s'_u, s'_d, a'_u)$ according to equation (2).
    end for
    $b \leftarrow b'$
    $a \leftarrow \pi(b)$
end while
```

3 Experiments and Results

3.1 Experimental Setup

In this paper, a teach-and-learn dialogue system based on the improved model is implemented. Where the ASR and text to speech (TTS) are implemented by calling Google API, NLG is implemented by simple template filling. The SLU module consists of intention recognition based on Maximum Entropy Model and slot value extraction based on Conditional random field.

In experiments, user attempts to teach a robot some object's type, color and shape. As shown in Table 1, let set of objects be *WhatSet*, set of class be *ClassSet*, set of color be *ColorSet* and set of shape be *ShapeSet*. The user's goal is given as $s_u = (x, y, z, w)$, where $x \in$ *WhatSet*, $y \in$ *ClassSet*, $z \in$ *ColorSet* and $w \in$ *ShapeSet*. The total number of user goals $|S_u|$, which is size of Cartesian product of four sets shown in Table 1, is 128. And the initial distribution over user goals is uniform. The user's action a_u is drawn from the set *teach-what-x*, *teach-class-y*,

Table 1. Set of value for each slot

What collection	Class collection	Color collection	Shape collection
target	fruit	red	circular
	vegetable	green	square
other	stationery	yellow	triangle
	other	other	other

teach-color-z, *teach-shape-w* and *other*, where in all cases $x \in \mathit{WhatSet}$, $y \in \mathit{ClassSet}$,$z \in \mathit{ColorSet}$ and $w \in \mathit{ShapeSet}$. The dialogue history $s_d = (x) : x \in \{0, 1\}$ indicates whether the current turn is the first turn (1) or not (0). These state components yield a total of 3841 states with an absorbed state according to $s = (s_u, a_u, s_d)$. The POMDP agent has 134 actions available, including *greet*, *ask-what*, *ask-class*, *ask-color*, *submit-x-y-z-w* and *fail*, where in all cases $x \in \mathit{WhatSet}$, $y \in \mathit{ClassSet}$,$z \in \mathit{ColorSet}$ and $w \in \mathit{ShapeSet}$.

It is assumed that the user's goal is fixed throughout the dialogue (i.e. $p(s'_u|s_u, a_m) = 1$ when $s'_u = s_u$). User action model indicates the probability of different responses user may take to some system's action under new user's goal. A portion of user action model is given in Table 2, the probabilities were handcrafted based on experience of slot-filling dialogue system. Probabilities in Table 2 describe the probability of every response user may take to different system's actions under the premise of that s'_u is *(targe,fruit,red,circular)*.

Table 2. Portion of user action model employed in experiments

a_m	s'_u	a'_u	$p(a'_u|s'_u, a_m)$
greet	*(targe,fruit,red,circular)*	*teach-what-target*	0.3
		teach-class-fruit	0.2
		teach-color-red	0.2
		teach-shape-circular	0.2
		other	0.2
ask-what	*(targe,fruit,red,circular)*	*teach-what-target*	0.6
		teach-class-fruit	0.15
		teach-color-red	0.09
		teach-shape-circular	0.09
		other	0.07

We define the observation function as

$$p(o'|s', a) = p(\tilde{a}'_u|a'_u) = \begin{cases} 1 - p_{err} & \text{if } \tilde{a}'_u = a'_u, \\ \frac{p_{err}}{|A_u|-1} & \text{if } \tilde{a}'_u \neq a'_u. \end{cases} \tag{6}$$

When solving POMDP policy we set p_{err} to 0.3. Observation received from SLU module will be utilized directly to update belief state in the improved SDS-POMDP when we are interacting with user.

Because of s_d contains only information about whether the current turn is the first turn (1) or not (0), the dialog history model $p(s'_d|a'_u, s'_u, s_d, a_m) = 1$ only when $s'_d = 0$ (i.e. next turn of dialogue must not be the first turn).

The reward measure reflects both task completion and dialog "appropriateness". The reward assigns -1 or -100 for taking the *greet* at beginning of the dialog or not, respectively; -10 for taking the *fail* action; +20 or -20 for taking the *submit-x-y-z-w* action when the user's goal is (x, y, z, w) or not, respectively; and -1 otherwise.

POMDP optimization was performed with an efficient point-based value iteration algorithm called *SARSOP* [5].

The MDP-based dialogue system is constructed to compare the improved SDS-POMDP in their ability to make decision under uncertainty of ASR/SLU. We define 81 MDP states as

$$\begin{aligned} &S = \{\mathit{what\text{-}class\text{-}color\text{-}shape}, \mathit{start}, \mathit{end}\}\ ,\ \mathit{where} \\ &\qquad \mathit{what} \in \{u, o, c\}\ , \mathit{class} \in \{u, o, c\}\ , \mathit{color} \in \{u, o, c\}\ , \mathit{shape} \in \{u, o, c\} \end{aligned} \tag{7}$$

The four components of *what-class-color-shape* refer to the *what* slot, *class* slot, *color* slot and *shape* slot respectively. u indicates unknown; o indicates observed but not confirmed; c indicates confirmed.

Action set of MDP-based DM is similar to that of POMDP-based DM. They both have *greet*, *ask-what*, *ask-class*, *ask-color*, *ask-shape* and *submit*. But dialogue system based on MDP needs four extra actions of *confirm-what*, *confirm-class*, *confirm-color* and *confirm-shape*. They are used to confirm teaching object, class of object, color of object and shape of object. In MDP, states are assumed to be observed completely. However, the value of attribute MDP obtains from SLU may be wrong, so every slot need to be confirmed with user once. Differently, POMDP uses observation function to model uncertainty of observation. When POMDP believe some slot is not certain enough, it will take the responding action of ask. For instance, when POMDP think color is not believable enough, it will continue taking action of *ask-color*. Thus, spoken dialogue system based on POMDP doesn't need action of confirm. The MDP is optimized using Q-Learning [1].

3.2 Results

In experiments, user will teach a robot four attributes of some objects includes object's name, class, color and shape. We have two measurements to evaluate our system. One is the length of dialogue denoted by D in which user has accomplished teaching an object. The other is Knowledge Acquisition Rate denoted by K, which is used to measure the level at which knowledge learned by system can meet user's requirement. The Knowledge Acquisition Rate is defined as

$$K = \frac{\mathit{Number\ of\ slots\ obtained\ correctly}}{\mathit{Number\ of\ slots\ obtained}} \tag{8}$$

Due to the characteristic of teaching task, we're paying more attention to knowledge acquisition rate. In the following experiments, three users are guided to teach these 12 groups of objects separately, and average knowledge acquisition rate $\bar{K}$ and average length of dialogue $\bar{D}$ are computed based on the data collected.

Table 3. The impact of adapting dynamic object binding

Dynamic object binding is employed?	$\bar{K}$	$\bar{D}$	$\bar{U}$
No	0.75	5.75	0.0
Yes	0.993	8.42	0.972

To evaluate whether the method of dynamic object binding can deal with uncertainty of teaching object, we have carried out a series of comparative experiments. Knowledge usability rate denoted by U is defined to show the necessity of introducing the method of dynamic object binding.

$$U = \frac{\textit{Number of objects whose all slots are taught correctly}}{\textit{Number of objects obtained}} \tag{9}$$

Average of U is denoted by $\bar{U}$. For instance, when user teaches system cherry, robot learns that name of teaching object is *other*, that class is fruit, that color is red and that shape is circular. Although type, color and shape in this case are all learned correctly, knowledge obtained from this dialogue is still useless for teaching object is not obtained correctly. The result of our experiment is show in Table 3. The value of N for N-Best in our experiment is 5. We can see that adopting dynamic object binding method makes the length of dialogue increase by 2.67 rounds as well as knowledge acquisition increase to 0.993 from 0.75. System without dynamic object binding can only achieve a knowledge acquisition rate of 0.75 because it can't obtain the slot of teaching object. Moreover, knowledge usability rate of system without dynamic object binding is poorly 0, because none of objects is taught completely correctly. That is, knowledge obtained by that kind of system can't be assigned to an appropriate object, which means it's of little use. Thus, we come to the conclusion that dynamic object binding can deal with the problem of infinite teaching objects.

Table 4. System performance with different size of N-Best in noise environment

N	$\bar{K}$	$\bar{D}$
1-Best	0.979	10.89
3-Best	0.993	9.28
5-Best	0.993	8.47

In Table 4, we give out how average knowledge acquisition rate and average length of dialogue change as N for N-best changes. As N increases, system's average knowledge acquisition rate increases and the length of dialogue decreases.

It can be concluded that a proper larger N can make use of more information, improve the accuracy of belief update and reliability of system and reduce the length of dialogue.

Table 5. The robustness of MDP-based and POMDP-based SDS in environment with and without noise

Dialogue system	In noisy environment or not	$\bar{K}$	$\bar{D}$
MDP	No	0.987	8.67
	Yes	0.882	10.17
POMDP	No	0.993	7.81
	Yes	0.979	10.89

The robustness of MDP-based and POMDP-based SDS to ASR error are compared in Table 5. Because of the limitation of MDP-based SDS, Only the top 1 hypothesis of N-Best is employed in both these systems for fairness. With the introduce of noise, the average knowledge acquisition rate of POMDP decreases by 0.014, but the average knowledge acquisition rate of MDP decreases by 0.105. It can be concluded that POMDP-based system is more capable of overcoming ASR errors. However, the average dialog length of POMDP increases by 3 rounds while that of MDP increases by 1.5 rounds. Length of POMDP increases because action “other” obtains a fairly high probability when noise exists and it won’t help to determine user’s goal. The reason why MDP increases less than POMDP is that MDP frequently confirm an incorrect value by a double error when noise exists. For example, an input of “它是个篮球 (It’s a basketball)” may be recognized as “他是 一个男囚 (He is male prisoner)”. “男囚 (male prisoner)” is extracted as the value of what. Then it’s confirmed with user. Unfortunately, user’s response of “不是 (No)” is mistakenly recognized as “是 (Yes)”. These two mistakes made by MDP may have itself regard the incorrect value as the correct. This kind of double error doesn’t increase dialog length but do decrease knowledge acquisition rate.

4 Conclusion

The improved SDS-POMDP model proposed in this paper can be employed to situation where the number of possible objects is infinite and every object has finite slots. Also, we come up with an approach to improve reliability of observation probability by making full use of N-Best list of ASR. We implement a teach-and-learn SDS based on the improved model and conduct a series of experiments on this system. Results of experiments indicate that our method can effectively overcome the infinite teaching objects problem in the teach-and-learn task. Moreover, estimating observation based on N-Best list of ASR can improve system robustness and the overall performance especially for dialogues with high ASR/SLU uncertainties. We solve the certain case of infinite slot values in this paper, and we will seek to model more general case of infinite slot values in future work.

Acknowledgments. This work was partially supported by National Natural Science Foundation of China (No.61273365, NO.61202248), discipline building plan in 111 base (No.B08004) and Engineering Research Center of Information Networks, Ministry of Education.

References

1. Barto, A.G.: Reinforcement learning: An introduction. MIT press (1998)
2. Hastie, H., Aufaure, M.a., Alexopoulos, P., Cuayáhuitl, H., Dethlefs, N., Gasic, M., Henderson, J., Lemon, O., Liu, X., Mika, P., Mustapha, N.B., Rieser, V., Thomson, B., Tsiakoulis, P., Vanrompay, Y., Villazon-terrazas, B., Young, S.: Demonstration of the Parlance system : a data-driven , incremental , spoken dialogue system for interactive search. In: Proceedings of the SIGDIAL 2013 Conference. pp. 154–156 (2013), http://www.aclweb.org/anthology/W/W13/W13-4026
3. Jokinen, K., McTear, M.: Spoken Dialogue Systems. Synthesis Lectures on Human Language Technologies 2(1), 1–151 (2009)
4. Kaelbling, L.P., Littman, M.L., Cassandra, A.R.: Planning and acting in partially observable stochastic domains. Artificial Intelligence 101(1), 99–134 (1998)
5. Kurniawati, H., Hsu, D., Lee, W.S.: SARSOP: Efficient Point-Based POMDP Planning by Approximating Optimally Reachable Belief Spaces. In: Robotics: Science and Systems (2008)
6. Levin, E., Pieraccini, R., Eckert, W.: A stochastic model of human-machine interaction for learning dialog strategies. Speech and Audio Processing, IEEE Transactions on 8(1), 11–23 (2000)
7. McTear, M.: Spoken dialogue technology: enabling the conversational user interface. ACM Computing Surveys (CSUR) 34(1), 90–169 (2002)
8. Seneff, S., Polifroni, J.: Dialogue management in the Mercury flight reservation system. In: Proceedings of the 2000 ANLP/NAACL Workshop on Conversational Systems - Volume 3. pp. 11–16. Association for Computational Linguistics (2000)
9. Shani, G., Pineau, J., Kaplow, R.: A survey of point-based POMDP solvers. Autonomous Agents and Multi-Agent Systems 27(1), 1–51 (Jun 2012), http://link.springer.com/10.1007/s10458-012-9200-2
10. Williams, J.D.: A case study of applying decision theory in the real world : POMDPs and spoken dialog systems. Decision Theory Models for Applications in Artificial Intelligence: Concepts and Solutions pp. 315–342 (2010)
11. Williams, J.D., Young, S.: Partially observable Markov decision processes for spoken dialog systems. Computer Speech & Language 21(2), 393–422 (Apr 2007)
12. Young, S., Gasic, M., Thomson, B., Williams, J.D.: Pomdp-based statistical spoken dialog systems: A review. Proceedings of the IEEE 101(5), 1160–1179 (2013)
13. Young, S., Gašić, M., Keizer, S., Mairesse, F., Schatzmann, J., Thomson, B., Yu, K.: The Hidden Information State model: A practical framework for POMDP-based spoken dialogue management. Computer Speech & Language 24(2), 150–174 (Apr 2010), http://linkinghub.elsevier.com/retrieve/pii/S0885230809000230
14. Zue, V., Seneff, S., Glass, J.R., Polifroni, J., Pao, C., Hazen, T.J., Hetherington, L.: JUPITER: A telephone-based conversational inferface for weather information. IEEE Transactions on Speech and Audio Processing 8(1), 85–96 (2000)