Trigram dialogue control using POMDPs

Yasuhiro Minami‡, Ryuichiro Higashinaka†, Kohji Dohsaka‡, Toyomi Meguro‡, and Eisaku Maeda‡
‡NTT Communication Science Laboratories, NTT Corporation
† NTT Cyber Space Laboratories, NTT Corporation

ABSTRACT

This paper proposes hybrid dialogue control of both trigram and POMDP dialogue controls by extending our proposed method that uses two approaches: automatically acquiring POMDP structures and rewards for target dialogues through Dynamic Bayesian Networks (DBNs) with a large amount of dialogue data and reflecting action predictive probabilities into the POMDP structures. In this extension, we modify the action predictive probabilities to treat trigram dialogue controls. Experimental results show that the proposed method can treat a trigram dialogue control with robustness for erroneous conditions and can simultaneously maximize trigram probability and the dialogue evaluations obtained from users.

Index Terms—Stochastic processes, Stochastic approximation, Cooperative systems, Stochastic automata, Intelligent robots

1. INTRODUCTION

Our research goal is to automatically acquire a conversation system's action control strategy for spoken dialogues. Here, we assume that the dialogue structure is unknown. Under this situation, the system must create and establish action control strategies based on a large amount of data for system-to-human or human-to-human communications. Several statistical models have been proposed for treating this situation [1-6]. POMDPs play an effective role in making decisions about selecting the most statistically reliable and available actions by observing speech with uncertainty. Dialogue controls using POMDPs exist for buying train tickets [7-8], for weather information dialogues [9], for digital subscriber line troubleshooting dialogues [10], and for the action control of robots by human speech and gestures [11]. Since these systems are based on task-oriented dialogue control and we know how the system should work, setting rewards and calculating transition probabilities are easy. However, if we do not know how the system should work, as in person-to-person communication, we have to estimate this with a large amount of data. The problem is how to create the POMDP structure from such a large amount of data. Although Fujita solved this problem with dynamic Bayesian networks (DBNs) to model a POMDP structure with a great deal of data, their task was simple and task-oriented [12].

Our proposed method automatically obtains the emission and observation probabilities of hidden states with a dynamic Bayesian network (DBN) based on expectation-maximization (EM) from a large amount of data [13]. Then it sets rewards for the POMDPs and performs value iteration to train a policy. In addition, our method introduces extra hidden states that match actions with one-to-one correspondence and sets the POMDP rewards to maximize the predictive probabilities of the hidden states using value iteration. With this procedure, dialogue control can generate an action sequence by reflecting the statistical characteristics of the training data. The proposed method is a hybrid method of an ordinary POMDP-based method and a probability-based method. Although Henderson has proposed a hybrid method of a reinforcement training method and a probability-based method [14], it only treats MDP conditions not POMDP conditions.

Dialogue action controllers using trigram models have also been proposed [15-16]. They select the action that maximizes the predictive probabilities for future action and observation sequences. Although the frameworks of POMDPs and trigrams seem similar, no studies have combined the two methods. In this paper, we extend our method so that it can cope with trigram dialogue statistics. Since one of the merits of POMDPs is input error robustness, we investigate our method in terms of this point. Our method is compared with the dialogue controllers using trigram models with actual dialogue data. In addition, we investigate whether our system can simultaneously maximize trigram probability and dialogue evaluation measurement obtained from users.

A general POMDP is presented in Section 2, our dialogue control using a POMDP and trigram models using our POMDPs are described in Section 3 and 4, the dialogue data used in this paper are shown in Section 5, and the evaluations of our action control algorithm are provided in Section 6. Finally, a discussion, future work, and a conclusion are given.

2. PARTIALLY OBSERVABLE MARKOV DECISION PROCESS

A POMDP is defined as $\{\mathrm{S}, \mathrm{O}, \mathrm{A}, \mathrm{T}, \mathrm{Z}, \mathrm{R}, \gamma,$ and $b_0\}$. S is a set of states described by $s \in \mathrm{S}$. O is set of observations o described by $o \in \mathrm{O}$. A is a set of actions a described by $a \in \mathrm{A}$. T is a set of state transition probabilities from s to s', given a, $\Pr(s' \mid s, a)$. Z is a set of emission probabilities of o' at state s', given a, $\Pr(o' \mid s', a)$. R is a set of expected rewards when the system performs action a at state s, $r(s,a)$.. The basic structure employed is shown in Fig. 1. The rhomboids show the fixed values, the dotted circles show the hidden variables, the solid circles show the observed variables, and the solid squares show the system actions.

In POMDPs, since states can't be directly observed, as in HMMs, we can only discuss their distribution. Here, suppose that the distribution of states $b_{t-1}(s)$ is known. Using the transition and emission probabilities, the distribution update is performed by

$$b_t(s') = \eta \cdot \Pr(o' \mid s', a) \sum_s \Pr(s' \mid s, a) b_{t-1}(s), \quad (1)$$

where η is a factor so that the distribution summation is one.

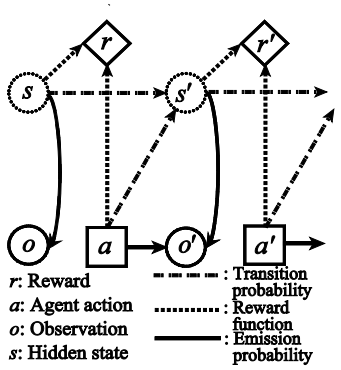

Fig. 1 POMDP structure

If the initial value of b is set as b_0, can be obtained iteratively using a recursive equation. With this distribution, the average discounted reward at time *t*, which is the objective function, can be obtained as

$$V_t^{\pi} = E_{\pi}\left[\sum_{\tau=0}^{\infty}\gamma^{\tau}\sum_{s} b_{\tau+t}(s)r(s,a_{\tau+t})\right], \quad (2)$$

where γ is a discount factor. POMDP obtains the optimal policy, which is a function from $b_t(s)$ to action a, by maximizing the average discounted reward in respect to π. In infinite time, the optimal value function tends to reach the equilibrium point in an iterative manner called the value iteration [17-19]. Although this value iteration obtains the optimal policy, it is time consuming. PBVI is comprised of one approximate solution technique [20-21].

3. HYBRID DIALOGUE CONTROL USING POMDPS

We previously proposed hybrid dialogue control of ordinary POMDP-based and probability-based dialogue controls. This section briefly explains this dialogue control.

3.1 Dialogue control goal

Our dialogue control assumes two conditions. One is that the statistics of the data are unknown and have to be estimated with training data. The other is that the data contain a set of dialogues we would like the system to achieve. Using them, we hope to achieve two purposes: having the system perform the target action sequences, and having the system perform the action sequences that reflect the data's statistical characteristics. We proposed a hybrid dialogue control [13] obtained from four methods to resolve the above issues:

(1) Automatically acquiring POMDP parameters
(2) Obtaining rewards that generate target dialogues
(3) Reflecting action predictive probabilities into POMDP structures
(4) Obtaining rewards for hybrid dialogue control and its policy

For (1) we use the DBN structure described in 3.2. For (2) and (3), we use two rewards: r_1 and r_2. In 3.3 and 3.4 we describe these rewards settings.

3.2 Automatically acquiring POMDP parameters and obtaining a policy for target dialogues

We proposed the POMDP training procedure shown in Fig. 2 [13]. POMDPs are required for training the transition probabilities, the emission probabilities, and the rewards described in the previous section. Ordinary dialogue systems assume that the probabilities and the rewards are given. In this paper, these parameters are automatically trained from the data. The corresponding DBN (Fig. 3) is used to train the probabilities in the POMDP. The DBN structure is constructed and trained by the EM algorithm and converted into the POMDP structure. After this, POMDP rewards are obtained. Finally, value iterations are performed to obtain a policy:

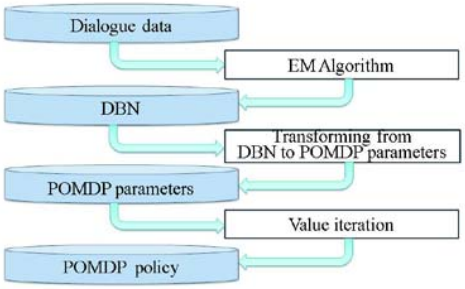

Fig. 2 Flow of training POMDP parameters and policy

3. 3 Obtaining rewards that generate target dialogues

The probabilities used in a POMDP can be obtained from the DBN structure. However the problem of calculating the rewards remains; this can be solved as follows. After the dialogue, the user evaluates whether the dialogue satisfied the given questionnaires by looking at its sequence. For example, our questionnaires asked whether the dialogue has closeness (familiarity). Based on this result, the user scores it. These scores are converted into rewards using variable d.

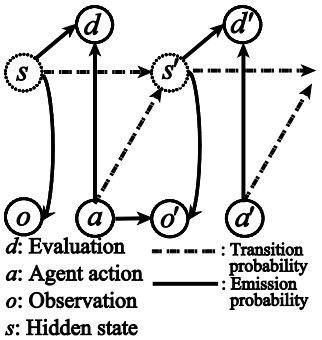

Fig. 3 DBN structure corresponding to POMDP

The following processes were used to make rewards and a POMDP for the target dialogue data:

(1) The positive evaluation score is set to the target data as variable d.
(2) A DBN is trained. d is also treated as a random variable.
(3) The DBN is converted to a POMDP, where we convert d

values and d 's probabilities into POMDP fixed rewards by

$$r_1(s,a) = \sum_{d=0}^{1} d \times \Pr(d \mid s,a). \quad (3)$$

(4) We set the reward into the POMDP structure.

If a state generates target dialogue data at a higher probability, the state should obtain higher rewards.

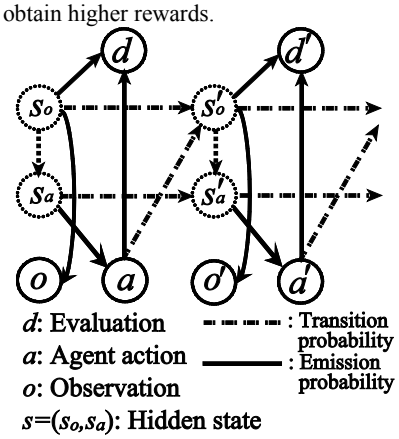

Fig. 4 DBN that reflects action predictive probabilities in action control

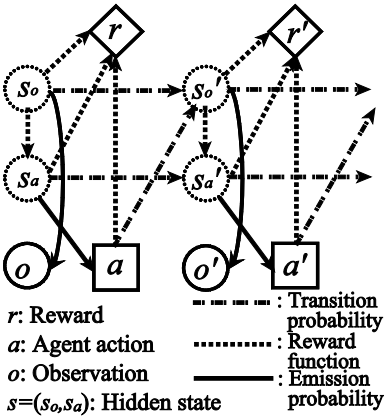

Fig. 5 POMDO that reflects action predictive probabilities in action control

3.4 Reflecting action predictive probabilities in POMDP structures

In dialogue control, naturalness is critical. We assume that if the system generates actions based on the statistics of the data, the actions are natural. To reflect action predictive probabilities, we introduce extra hidden DBN and POMDP states to those in Figs. 1 and 3 as $s = (s_o, s_a)$ (Figs. 4 and 5). s_o is identical as the previous state in Figs. 1 and 3. s_a is introduced for estimating the predictive probability of action a and for selecting a to maximize the predictive probability. This selection is performed by following the reward settings. This method is an extension of the one described in the previous section. Due to increased parameters, the following probability approximations are used (Fig. 4):

$$\Pr(s' \mid s,a) \approx \Pr(s'_a \mid s_a, s'_o) \Pr(s'_o \mid a, s_o), \text{ and} \quad (4)$$

$$\Pr(o' \mid s',a) \approx \Pr(o' \mid s'_o). \quad (5)$$

With these approximations, $b_t(s')$ can be written as

$$b_t(s') = b_t(s'_o, s'_a)$$
$$= \eta \Pr(o' \mid s'_o) \sum_s \Pr(s'_a \mid s_a, s'_o) \Pr(s'_o \mid s_o, a) \Pr(a \mid s_a) b_t(s_o, s_a). \quad (6)$$

Note that although the original formulation described in [13] ignores factor $\Pr(a \mid s_a)$, it is important because after the policy selects an action, it is fixed; it is no longer probabilistic. In this paper, we apply this information to the hidden states for actions.

The corresponding objective function can be written as

$$V_t^{\pi} = E^{\pi}\left[\sum_{\tau=0}^{\infty} \gamma^{\tau} \sum_s b_{\tau+t}(s_o, s_a) r((s_o, s_a), a_{\tau+t})\right]. \quad (7)$$

We introduce rewards into (7) so that if $b_{\tau+t}(s_a)$ is high, the POMDP may obtain a higher reward. If $a = s_a$,we set $\Pr(a \mid s_a) = 1$ in the DBN (Fig. 4) so that s_a corresponds one-on-one with a . Based on this, if $a_t = s_a$ is given, we obtain

$$\Pr(a_t \mid o_1, a_1, \ldots, a_{t-1}, o_t)$$
$$= \sum_{s'_a} \Pr(a_t \mid s'_a) \Pr(s'_a \mid o_1, a_1, \ldots, a_{t-1}, o_t) \quad (8)$$

$$= \Pr(s_a \mid o_1, a_1, \ldots, o_{t-1}, a_{t-1}, o_t) = b_t(s_a), \quad (9)$$

where

$$b_t(s_a) = \Pr(s_a \mid o_1, a_1, \ldots, o_{t-1}, a_{t-1}, o_t) = \sum_{s_o} b_t(s). \quad (10)$$

These are for propagating the predictive probabilities of the actions into the probabilities of hidden states ($b_t(s_a)$). Our objective here is to select a_t so that the probability of a_t is maximized when $o_1, a_1, \ldots, o_{t-1}, a_{t-1}, o_t$ are given. The rewards must be set to satisfy this result because they must be set by maximizing (10). To do this, we set $r_2(s = (*, s_a), a) = 1$ when $s_a = a$, where * is arbitrary s_o . Otherwise, $r_2(s = (*, s_a), a) = 0$.

3.5 Obtaining rewards for hybrid dialogue control and its policy

To obtain final rewards for hybrid dialogue control we replace r in (7) into $r_1 + r_2$ as

$$r(s,a) = r_1((s_o, *), a) + w \cdot r_2((*, s_a), a); \quad (11)$$

we obtain new objective function V_t^{π} with the POMDP structure shown in Fig. 5 and modify reward definition r_1 described in (3) using * so that we can handle extra hidden states s_a . The POMDP policy is then trained by value iteration. Using this formulation, the POMDP can select the action that simultaneously gives higher predictive probability of the action and obeys the target dialogue sequence.

4. EXTENTION TO TRIGRAM DIALOGUE MODEL

In this section we describe how to model trigram dialogue control in POMDPs.

4.1 General trigram model for action control

From past research [14-15], this formulation

$$a_t = \arg\max_{a_t} \left[\max_{a_t, o_t} \{\Pr(o_t \mid o_{t-1}, a_{t-1}) \Pr(a_t \mid a_{t-1}, o_t)\} \right] \quad (12)$$

is used for estimating the next action; here suppose that O_t is a category. We can modify this formulation. Assuming that current O_t is directly observed, we obtain

$$a_t = \arg\max_{a_t} \{\Pr(a_t \mid a_{t-1}, o_t)\}. \quad (13)$$

Considering one-step future predictive probabilities and using summation with respect to O_t instead of maximization, we obtain

$$a_t = \arg\max_{a_t} [\max_{a_t, a_{t+1}} \{\Pr(a_t \mid a_{t-1}, o_t) \cdot \sum_{o_{t+1}} \Pr(o_{t+1} \mid o_t, a_t) \Pr(a_{t+1} \mid a_t, o_{t+1})\}]. \quad (14)$$

In this paper, we target this formulation.

4.2 Treating trigram dialogue model in POMDP

Here we set $|o| = |s_o|$. If $o = s_o$, we set $\Pr(o \mid s_o) = 1$ in the DBN (Fig. 4) so that s_o corresponds one-on-one with O. Based on this, if $o_t = s_o$ and $a_t = s_o$ is given, we obtain

$$\Pr(s' \mid s, a) = \Pr(s'_O \mid s_O, a) \Pr(s'_a \mid s'_O, s_a) = \Pr(o' \mid o, a) \Pr(a' \mid a, o'). \quad (15)$$

This is the same formulation as the trigram predictive probability calculation process in one interaction described in the previous section. Using this formulation with $r_1((s_o, *), a) = 0$, we obtain a target function that sums up the future trigram statistics for actions. The difference with the target function of the trigram model is that our model calculates the summation of the probabilities with appropriate future action decisions, and the trigram model calculates the best path with short future candidates. In addition, we introduce $\Pr(a \mid s_a)$ into the state update equation (6). This part is also different.

4.3 Treating uncertain input

One merit of POMDP is treating the uncertainty of such input as speech recognition and dialogue type recognition results. To treat such uncertain data, we set recognition accuracy to $\Pr(o \mid s_o)$. Using this formulation, O is the observed symbol with recognition error, and s_o is the true value of O.

5. DIALOGUE DATA

We used text dialogue data to evaluate our methods [22]. In this dialogue, the user and system discuss likes and dislikes about animals (fireflies in this case), disclosing preferences and agreeing with each other. An example dialogue is shown in Fig. 6.

S1 Hello. (GREETING)
U1 Hello. (GREETING)
S2 Let's talk about what kind of animals and insects we like. (OPEN-DIALOGUE)
U2 Do you like insects? (Q-DISC-P)
S3 Do you like insects? (Q-DISC-P)
U3 I like fireflies. (DISC-P)
S4 You like fireflies. (REPEAT)
U4 I like fireflies because they glow. (DISC-P-R)
S5 I also like them very much. (DISC-AGREE-P)
U5 Why do you like them? (Q-DISR-R)
S6 I like them too. (DISC-AGREE-P)
They are fleeting. (DISC-R)
U6 I like them because they are beautiful. (DISC-P-R)
S7 I see. (ACK),
I understand. (EMP)
I also think they are beautiful. (DISC-AGREE-R)
Can you tell me more about why you like fireflies? (Q-DISC-R-OTHER)
U7 I like them because they glow at night. (DISC-P-R)

Fig. 6 Dialogue excerpt from our experiment. S stands for system utterance, and U for user utterance. Dialogue acts corresponding to utterances are shown in parentheses.

We defined 29 dialogue acts, which were used for action and observation labels. We list their major 22 dialogue acts below.

Self-disclosure: DISC-P, DISC-R, DISC-P-R, DISC-R-OTHER, and RES. DISC-P is used to disclose proposition P. A proposition is either like (X,Y) or dislike (X,Y), meaning that conversational participant X likes animal Y or vice versa. DISC-R discloses reason R for some aforementioned proposition P. DISC-P-R simultaneously discloses P and R, and DISC-R-OTHER discloses R in addition to the already mentioned reasons (e.g., "another reason is that. . . "). We consider RES, which is a response to a yes-no question (i.e., Q-DISC-P), to be self-disclosure.

Agreement: DISC-AGREE-P, DISC-AGREE-R, EMP, and REPEAT. DISC-AGREE-P and DISC-AGREE-R show agreement to the propositions or reasons mentioned by the partner. EMP denotes an explicit empathic action (e.g., "I understand"), and REPEAT means the repetition of the partner's previous self-disclosure to show understanding.

Disagreement: DISC-DISAGREE-P and DISC-DISAGREER. They show disagreement to the propositions or reasons mentioned by the partner; e.g., saying "I don't like cats" to a partner who has already disclosed that he/she likes cats.

Dialogue-control: GREETING, GOODBYE, OPENDIALOGUE, and Q-OPEN-DIALOGUE and CLOSE-DIALOGUE as dialogue-initiating/ending acts. SHIFT-TOPIC introduces a new topic (animal) into the dialogue.

Question: Four questioning acts, Q-DISC-P, Q-DISC-POPEN (an open question such as "how about cats?"), Q-DISCR, and Q-DISC-R-OTHER, ask for propositions or reasons from the partner.

Acknowledgment: ACK acknowledges the partner's utterance using back-channels.

In the actual dialogues, participants expressed several utterances during the same turn. POMDPs and trigram models, however, cannot handle multiple utterances. To avoid this situation, we insert label "eps" between the utterances of the same participant. After each dialogue, an annotator (not a participant) filled out a questionnaire (five-point Likert scale) that asked for subjective evaluations of the dialogue. The questionnaires asked about closeness.

To simulate the input errors for checking the robustness of the methods, we used errors from the following dialogue act recognition. User utterances were first separated into word tokens using a Japanese morphological analyzer and converted into dialogue acts by understanding the grammar realized as a weighted finite state transducer (WFST) in a manner similar to [23]. We defined the sequences of words that formed dialogue acts and from them compiled a WFST that maps a sequence of words into a scored list of dialogue acts augmented with attribute-value pairs. In all, our grammar has a vocabulary of 2,276 words, including 1,005 adjectives taken from the evaluative expression dictionary [24].

We annotated the user utterances with correct dialogue acts. A single annotator (not one of the authors) annotated each dialogue. The system's dialogue act recognition accuracy (excluding DISC-OTHER and OTHER) was 50%.

6. EXPERIMENTAL RESULTS

We evaluated our system under both error free and error conditions.

6.1 Experimental setting

We used 90 dialogue sequences for training the trigrams and the DBNs. The average turn length in a dialogue sequence was 38 turns. The trigrams and the DBNs were trained separately. We confirmed that both sets of statistics are completely identical. We didn't use a smoothing technique for estimating both sets of statistics to focus on the basic performances of both methods. The user actions were simulated by randomly selecting them based on the trigram statistics obtained from the training data. Using conventional models and the proposed methods, 1000 simulated dialogues were generated for evaluation.

6.2 Evaluation measure

We prepared two measures to evaluate the methods. One is the average trigram probability of the generated actions defined by

$$E_1 = \frac{1}{N}\sum_{i}^{N}\frac{\sum_{t} P\,(a^{i}_{t+1} \mid a^{i}_{t}, o^{i}_{t+1})}{L_i}, \qquad (16)$$

where N is the number of dialogues, L_i, is the length of each one, and $P_i\,(a^{i}_{t+1} \mid a^{i}_{t}, o^{i}_{t+1})$ is the training data trigram probability of a^{i}_{t+1} given a^{i}_{t}, o^{i}_{t+1}. This measure checks how well the generated actions maximize the training data statistics. The other is the average user evaluation scores defined by

$$E_2 = \frac{1}{N}\sum_{i}^{N}\frac{\sum_{t}\overline{d}\,(a_t, o_t)}{L_i}, \qquad (17)$$

where $\overline{d}\,(a,o)$ is the average user evaluation score for the observation and action pairs. Although user evaluations are affected by the dialogue histories, we assumed that the user evaluations for a system are strongly affected by the actions responding to the last user observation.

6.3 Experimental result under error free condition (without introducing closeness rewards)

We prepared the following three conventional methods:

(1) Random
Action is randomly generated by the following equation:

$$a_t \sim P(a_t \mid a_{t-1}, o_t). \qquad (18)$$

(2) Trigram-0
Action is selected by the current trigram model such as

$$a_t = \arg\max_{a_t}\{P(a_t \mid a_{t-1}, o_t)\}. \qquad (19)$$

(3) Trigram-1
Action is selected by a one-step future trigram model such as

$$a_t = \arg[\max_{a_t}\ \max_{a_{t+1}}\{P(a_t \mid a_{t-1}, o_t) \cdot \sum_{o_{t+1}} P(o_{t+1} \mid o_t, a_t) P(a_{t+1} \mid a_t, o_{t+1})\}]. \qquad (20)$$

We prepared the following two proposed methods:

(4) POMDP ($\gamma = 0.0$)

Although this is almost the same as Trigram-0, the parameters are automatically trained by value iteration.

(5) POMDP ($\gamma = 0.7$)

Using this setting, POMDPs consider more future probabilities than Trigram-1.

Table 1 shows the E_1 evaluation result for the conventional and proposed methods. Trigram-1 outperforms Trigram-0. This means that future information is important. The performances of the proposed methods are almost the same as those of the trigram methods. Since POMDPs can reduce the calculation time, they can be used instead of the trigram methods. POMDP's result with $\gamma = 0.0$ is worse because POMDPs use point-based value iteration, which is not an exact value iteration but an approximation.

Table 1 Experimental result for error free condition

	Random	Trigram-0	Trigram-1	POMDP ($\gamma = 0.0$)	POMDP ($\gamma = 0.7$)
E_1	0.380	0.462	0.466	0.460	0.467

6.4 Experimental result under an erroneous condition (without closeness rewards)

To simulate the erroneous condition, we used the confusion matrix obtained by the data collection process described in Section 4. In this condition, POMDPs outperform the trigram methods (Table 2). This is reasonable since POMDPs know the error statistics and can adapt to the error.

Table 2 Experimental result for noisy condition

	Random	Trigram-0	Trigram-1	POMDP ($\gamma = 0.0$)	POMDP ($\gamma = 0.7$)
E_1	0.329	0.433	0.429	0.439	0.440

6.5 Experimental result under an erroneous condition (adding closeness rewards)

Our POMDP models generated actions that obtained higher evaluation scores. In this experiment, we generated closeness rewards using Eq. (3) and set *w* in Eq. (11) to 10.0. Table 3 shows the result of E_1 and E_2. POMDPs increased the E_2 values without significant degradation of E_1

Table 3 Experimental result for erroneous condition (adding closeness rewards)

	Random	Trigram-0	Trigram-1	POMDP ($\gamma = 0.0$) +closeness rewards	POMDP ($\gamma = 0.7$) +closeness rewards
E_1	0.329	0.433	0.429	0.436	0.428
E_2	2.66	2.65	2.66	2.75	2.76

7. CONCLUSIONS

We extended our POMDP models so that they treat trigram dialogue control models and investigated POMDP robustness for erroneous conditions using trigram dialogue control. Our experimental results confirmed that our method also simultaneously maximized trigram probability and the closeness obtained from user evaluations. These results confirm that our proposed POMDP frameworks can be used for a variety of dialogue control strategies.

8. ACKNOWLEDGMENTS

This work was supported by a Grant-in-Aid for Scientific Research on Innovative Areas, "Formation of robot communication strategies" (21118004), from the Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan.

9. REFERENCES

[1] N. Roy, J. Pineau, and S. Thrun, "Spoken Dialogue Management using Probabilistic Reasoning," *Proc. ACL*, pp. 93-100, 2000.

[2] S. J. Young, "Probabilistic Methods in Spoken Dialogue Systems," Philosophical Trans of the Royal Society (Series A), 1769 (358), pp. 1389-1402, 2000.

[3] E. Levin and R. Pieraccini, "A Stochastic Model of Computer-Human Inter-Action for Learning Dialog Strategies," *Proc. Eurospeech*, pp. 1833-1886, 1997.

[4] E. Levin, R. Pieraccini, and W. Eckert, "Using Markov Decision Process for Learning Dialogue Strategies," *Proc. ICASSP*, pp. 201-204, 1998.

[5] R. Higashinaka, M. Nakano, and K. Aikawa, "Corpus-based Discourse Understanding in Spoken Dialogue Systems," *Proc. ACL*, pp. 240–247, 2003.

[6] M. Denecke, K. Dohsaka, and M. Nakano, "Learning Dialogue Policies Using State Aggregation in Reinforcement Learning," *Proc. ICSLP*, pp. 325–328, 2004.

[7] J. Williams, P. Poupart, and S. Young, "Partially Observable Markov Decision Processes with Continuous Observations for Dialogue Management," *Proc. SIGdial*, pp. 25-34, 2005.

[8] J. Williams, P. Poupart, and S. Young, "Factored Partially Observable Markov Decision Processes for Dialogue Management," *Proc. IJCAI*, pp. 75-82, 2005.

[9] K. Kim, C. Lee, S. Jung, and G. G. Lee, "A Frame-Based Probabilistic Framework for Spoken Dialog Management Using Dialog Examples," *Proc. SIGdial*, pp. 120-127, 2008.

[10] J. Williams, "Using Particle Filters to Track Dialogue State," *Proc. ASRU*, pp. 502-507, 2007.

[11] S. R. Schmidt-Rohr, R. Jakel, M. Losch, and R. Dillmann, "Compiling POMDP Models for A Multimodal Service Robot From Background Knowledge," *European Robotics Symposium*, pp. 53-62, 2008.

[12] H. Fujita, "Learning and Decision-planning in Partially Observable Environments," *Ph.D. dissertation*, Nara Institute of Science and Technology, 2007.

[13] Y. Minami, A. Mori, T. Meguro, R. Higashinaka, K. Dohsaka, and E. Maeda, "Dialogue Control Algorithm for Ambient Intelligence based on Partially Observable Markov Decision Processes," *Proc. ISCA IWSDS*, pp. 254-263. 2009.

[14] J. Henderson, O. Lemon, and K. Georgila, "Hybrid Reinforcement/supervised Learning of Dialogue Policies from Fixed Data Sets," Computational Linguistics, vol. 34, pp. 487-511, 2008.

[15] C. Hori, K. Ohtake, T. Misu, H. Kashioka, and S. Nakamura, "Weighted Finite State Transducer based Statistical Dialog Management," *Proc. ASRU*, pp. 490-495, 2009.

[16] C. Hori, K. Ohtake, T. Misu, H. Kashioka, and S. Nakamura, "Recent Advances in WFST-based Dialog System," *Proc.* Interspeech, pp. 268-271, 2009.

[17] R. S. Sutton and A. G. Barto, "Introduction to Reinforcement Learning," The MIT Press, 1998.

[18] S. Russell and P. Norvig, "Artificial Intelligence: a Modern Approach Second Edition," Prentice Hall, 2003.

[19] P. Poupart, "Exploiting Structure to Efficiently Solve Large Scale Partially Observable Markov Decision Processes," *Ph.D. dissertation*, University of Toronto, 2005.

[20] T. Smith and R. G. Simmons, "Point-Based POMDP Algorithms: Improved Analysis and Implementation," *Proc. UAI*, pp. 542-547, 2005.

[21] J. Pineau, G. Gordon, and S. Thrun, "Point-Based Value Iteration: an anytime algorithm for POMDPs," *Proc. IJCAI*, pp. 1025-1032, 2003.

[22] Ryuichiro Higashinaka, Kohji Dohsaka, Hideki Isozaki, "Effects of Self-Disclosure and Empathy in Human-Computer Dialogue," *Proc. SLT* , pp.109-1012, 2008.

[23] A. Potamianos and H.-K. J. Kuo, "Statistical recursive finite state machine parsing for speech understanding," *Proc. ICSLP*, vol. 3, pp. 510–513, 2000.

[24] H. Takamura, T. Inui, and M. Okumura, "Extracting semantic orientations of words using spin model," *Proc. 43rd ACL*, pp. 133–140, 2005.