Decision of Response Timing for Incremental Speech Recognition with Reinforcement Learning

Di Lu #1, Takuya Nishimoto *2, Nobuaki Minematsu #3

Graduate School of Information Science and Technology, The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656 Japan
[1] lu@gavo.t.u-tokyo.ac.jp
[3] mine@gavo.t.u-tokyo.ac.jp

* *Olarbee Japan*
2-13-7 Hataka, Akiku, Hiroshima, 736-0088 Japan
[2] nishimotz@olarbee.com

***Abstract*—In spoken dialog systems, it is important to reduce the delay in generating a response to a user's utterance. We investigate the use of incremental recognition results which can be obtained from a speech recognition engine before the input utterance ends. To enable the system to respond correctly before the end of the utterance, it is desired to utilize the incremental results effectively, although they are not reliable enough. We formulate this problem as a decision making task, in which the system makes choices iteratively either to answer based on previous observations, or to wait until the next observation. The reinforcement learning can be applied to the problem. As the results of experiments, the users highly evaluate the proposed method which estimate completion time of a user's utterance by using the results of speech recognition based on mora units.**

I. Introduction

With the significant improvement of the speech recognition and speech synthesis technology, the speed of speech processing becomes faster and the accuracy of speech recognition is also greatly improved even in the noisy environment. As a result, various kinds of spoken dialog systems based on such technology are implemented and made into use. However, the spoken dialog systems are not so popular today as expected previously. Sagayama tried to explain why the spoken dialog systems are not widely used as expected [1]. The following assumptions were proposed: (1)The appearance of the machine is not human-like. (2)The conversation between the machine and a user is not human-like (3)The machine is not intelligent enough.

To solve first problem, Galatea project [2] was carried out. In this project, Galatea toolkits, including speech recognition model Julius [6], speech synthesis model Gtalk and face synthesis model FSM, are developed and distributed as an open source. As a result, the first problem mentioned above is thought to be solved by developing an anthropomorphic spoken dialog agent.

This paper mainly describes an approach to solve the second problem, and find an efficient way to control the response timing and make the communication between a human and a machine more human-like.

In human-human communication, they can response for the partner's speech very fast even before the partner's speech ends. This phenomenon is called barge-in which commonly happens in human-human communication. For example, in a two-men conversation, whilea speaker is explaining something to the listener, the listener may suddenly interrupt the speaker and ask "Pardon?" as soon as the listener doesn't understand what the speaker just said at any time.

In this paper, we propose a method of using response timing control for human-machine conversation. The preliminary experimental result shows the effectiveness of this method.

II. Dialog Management for Smooth Conversation

A. Target

For most of the existing spoken dialog systems, a certain length of delay in generating a response to a user is very difficult to avoid [3].

Traditional continuous speech recognition engines, such as Julius of Galatea Toolkit, they commonly use a certain silence threshold, such as power threshold and zero-crossing threshold, to determine the end of the user's speech. Therefore a speech recognition result cannot be determined before the user's speech ends. As a result, the delay whose length equals to the silence threshold cannot be avoided, in addition to the delay of speech recognition processing. Also, since most of the spoken dialog systems is build with multi-process(thread), technology for transition of the sharing data, such as mutual exclusion or socket communication, is widely used, which results in a delay of the communication between modules. What's more, the determination of the response according to the dialog policy by the dialog manager also costs time.

In human-human communication, when the speaker is making an utterance, the listener can make use of the partial information of the utterance to guess what the speaker is going to saying and decide what to response instantly even there is risk of misunderstanding. As long as the risk is small enough, it is meaningful to realize the smooth conversation.

B. Traditional Methods

One approach is to get the incremental speech recognition result before the user's speech ends. Imai [4] proposed that the speech recognition result can be determined when the optimal

TABLE I
INCREMENTAL RECOGNITION RESULTS FOR "MO-I-K-KA-I-O-NE-GA-I-SHI-MA-SU"(IN JAPANESE)

Time(ms)	Incremental recognition results
300	U-n
600	Ha-i
900	A-ri-ga-to-go-za-i-ma-su
1200	Mo-i-k-ka-i-o-ne-ga-i-shi-ma-su
1500	Mo-i-k-ka-i-o-ne-ga-i-shi-ma-su
Final	Mo-i-k-ka-i-o-ne-ga-i-shi-ma-su

speech recognition result lasts a certain number of frames before the input utterance ends. This technology is used to add the subtile to the broadcast news. Skantze[5] proposed an incremental speech production model with a incremental speech recognition module, which can read out the speech recognition result incrementally by decreasing the silence threshold. Incremental spoken dialog systems make it possible to response before the user's utterance ends. Nishimura [6] proposed to add prosody analysis to the incremental speech recognition. Also DUG-1[7] of NTT and Robisuke [8] of Waseda Univ. proposed how to realize a barge-in function for speech input. However, these methods rely only on a specific task or vocabulary and are not robust against recognition errors.

III. PROPOSED METHOD

A. Overview

According to the problems of the traditional methods, the ollowing three problems are needed to be solved.

1) Obtain the reliable speech recognition result before the end of the user's speech.
2) Predict the user's speech as fast and correctly as possible.
3) Start the response at the proper timing for the user.

The solutions are described separately as follow.

B. Sub-word Speech Recognition

Whether a reliable speech recognition result can be obtained before an input utterance ends relies partly on the vocabulary. For example, speech recognition engine Julian has a function of displaying incremental speech recognition results generated by the first pass decoding. An incremental search engine for a spoken query was implemented [9]. To narrow a search space according to the user's speech, partial words should be included in the vocabulary. For example, for a key word "me-n-ti-ka-tu", its partial key words "me-n-ti" should be included in the vocabulary.

Table I shows incremental speech recognition results for "Mo-i-k-ka-i-o-ne-ga-i-shi-ma-su"(in Japanese) using Julius. "Final" is the second pass recognition result. As the passage of time, the incremental recognition result is becoming more similar to the second pass recognition result. But at the early stage, it is hard to get the correct recognition result because the processed segment is not long enough.

To make it easy to recognize the user's speech with short speech input entry, we divide the recognition candidate words in the vocabulary into sub-word such as phoneme, mora and syllable. For example, a Japanese phrase "ko-n-ni-ti-wa"(meaning "hello" in English) can be divided into five morae: "ko", "n", "ni", "ti", "wa", where each of the mora is represented as one of the hirakana in Japanese. Then the sub-words "ko", "ko-n", "ko-n-ni", "ko-n-ni-ti" can be added to the vocabulary. If the vocabulary is limited to "ko-n-ni-ti-wa" and "ko-n-ba-n-wa", we can determine the speech "ko-n-ni-ti-wa" with the initial three morae "ko-n-ni".

It is not necessary to divide the words in the vocabulary into mora units. Phoneme and syllable can be also used in the sub-word vocabulary. Since Japanese is a language using mora as the basis of the sound system and mora is a unit of sound with a certain length, we mainly focus on the mora unit in this paper.

This is similar with a Japanese (New Year's) card game Hyakunin Isshu that can be modeled as cohort model, in which the card can be confirmed only by several initial alphabets as long as the card set is limited.

C. Prediction of a User's Speech

1) Reinforcement Learning: As mentioned above, at each time step the user's speech can be predicted according to the incremental speech recognition result (observation), Therefore the prediction can be formulated as iterative choices between determining the user's speech using previous observations, and waiting until next observation. Since the prediction should be as fast and correct as possible, it is expected for a agent to choose a correct action at an early stage. As the passage of time, the incremental result becomes reliable, but waiting for the next incremental recognition result may result in the delay of the response. So there is a trade-off between the accuracy and the timing of response. Since it is difficult to find a proper trade-off using a rule-based model, machine learning is expected to be helpful to solve the problem. Ishiguro [10] proposed to use boosting for early classification. However, it is also difficult for automatic determination for each incremental speech recognition result with supervised learning because of the lack of the training data. If we treat the process of one determination of the user's speech as one episode, the accuracy and the timing of the response is easier to be evaluated for each episode. That is to say, at each episode, the system is highly evaluated by the earlier and correct determination, and vice versa.

In human-human conversation, the listener can determine the speaker's speech before the speech ends in spite of the risk of misunderstanding. They can make the trade-off between the accuracy and the timing of the determination, which is thought to be learnt for the experience.

In machine learning, such a problem can be solved by reinforcement learning [11]. Reinforcement learning is different with supervised learning in that correct labeled training data never exists. An agent learns which actions should be taken in a given environment so that reward can be maximized (Fig 1). In this paper, the state is an incremental speech recognition result and the action is either determining the user's utterance or waiting for the next incremental recognition result.

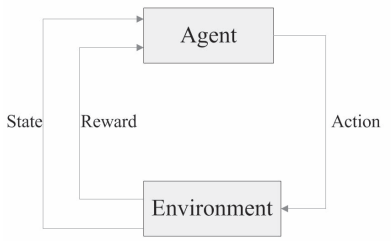

Fig. 1. Overview of Reinforcement Learning

Reinforcement learning has been proven useful for the dialog management of a spoken dialog system [12][13][14]. The difference between the existing methods and our proposal is that the action is determined at each incremental speech recognition result instead of at each turn unit.

2) Reinforcement Learning with a Simulator: For the learning of the dialog policy, it is too much cost to perform the reinforcement learning between a human and an agent through many trials, especially for a large-scaled task. That is because at the initial period of learning, the agent doesn't know any dialog policy (yet don't have any knowledge) and it has to choose the action randomly, which will take too much time. As a result, before the reinforcement learning between a human and an agent is conducted, reinforcement learning is usually done between an agent and a user simulator that can perform the user action by simulation. The user simulator in this paper is designed according to handcrafted rules or probabilities which were obtained from real-world dialog history statistically. Thus, the agent can learn some basic dialog policy through trial errors by an easy rule-based simulator. Since the dialog examples can be generated by a computer, it will cost much less time than learning between a human and an agent. Also the condition of learning and the change of reward can be easily evaluated.

The simulator that can output incremental speech recognition results can be designed as follows.

1) Determine a user's utterance, the speaking rate (morae/s)and the error rate of speech recognition.
2) Generate the sub-word speech in mora units according to the speaking rate and the time, while a part of the speech utterance may be substituted by a wrong word according to the error rate.
3) One episode ends when the system determines the user's speech and generates the response. Positive reward is given by the simulator when the timing of determining the speech is late. Negative reward is given by the simulator when the speech is determined correctly.

3) Belief Update by POMDP: Commonly reinforcement learning is modeled as Markov Decision Process(MDP) and Partial Observable Markov Decision Process(POMDP), which is an extended version of MDP.

POMDP is modeled as $\{S, A, T, R, O, Z\}$, where S is a set of states s which describe the environment with $s \in S$, A is a set of actions a that the agent may take with $a \in A$, T stands for a transition probability $p(s'|s, a)$ which shows the probability of the transition from s to s' while the agent's action is a, and R defines the expected real-valued reward $r(s, a) \in R$. What makes POMDP differs from MDP is O and Z, where O is a set of observations and Z defines an observation probability $p(o|s, a)$. In the POMDP framework, belief b is represented by distributions over states and updated from observations by Bayesian inference. Since the state s is not known exactly, the belief $b(s)$ stands for the probability of the state being state s, which can be inferred from the observation o. The formulation is shown as follows:

$$b(s) = \eta \cdot p(o|s, a) \sum_{s} p(s'|s, a) b(s) \tag{1}$$

In our approach, the dialog state S stands for all possible utterances that the user may speak. The observation O is the incremental speech recognition result produced by speech recognition engine so that o is every possible partial words of the user's utterances. The agent's action A is defined as ether to wait for the next observation or to determine the user's utterance and start to response. At each timestep, the agent observes incremental recognition result o and updates $b(s)$ which stands for the probability of user's utterance being s.

An example shows the update process of brief monitor with a simple spoken dialog domain in Figure 2. It is assumed that only three utterances are contained in the vocabulary, "a-ka-mo-n-wa-do-ko-de-su-ka", "a-ri-ga-to-go-za-i-ma-su" and "mo-u-i-k-ka-i". While the user is speaking the utterance "a-ka-mo-n-wa-do-ko-de-su-ka", the incremental speech recognition result is output at each time step. Note that a speech recognition error is made and is handled by the POMDP framework well, whereas the POMDP belief state is more robust.

D. Estimation of End Timing of User's Speech

The proper timing of the system's response for the user's speech relies on the task or the user's speech. Assume that the system is successful to determine what the user intends to say before the speech ends, it is possible for the system to start the response (1) as soon as possible, (2) or as soon as the user's speech ends.

To realize (2), the end timing of the use's speech has to be estimated.

At the timing when the agent determined the user's intention, let L_{IU} be the length of the incremental unit. Since the timing when the user's utterance starts can be obtained by speech recognition module, we can get the duration$t_{duration}$ of the user's incremental speech. Thus we can estimate the user's speaking rate s_{speech} by

$$s_{speech} = \frac{L_{IU}}{t_{duration}} \tag{2}$$

Then the user's end timing Δt, which means the user's speech is thought to finish Δt later, can be calculated with the total length of the speech L_{speech} that the system determines as

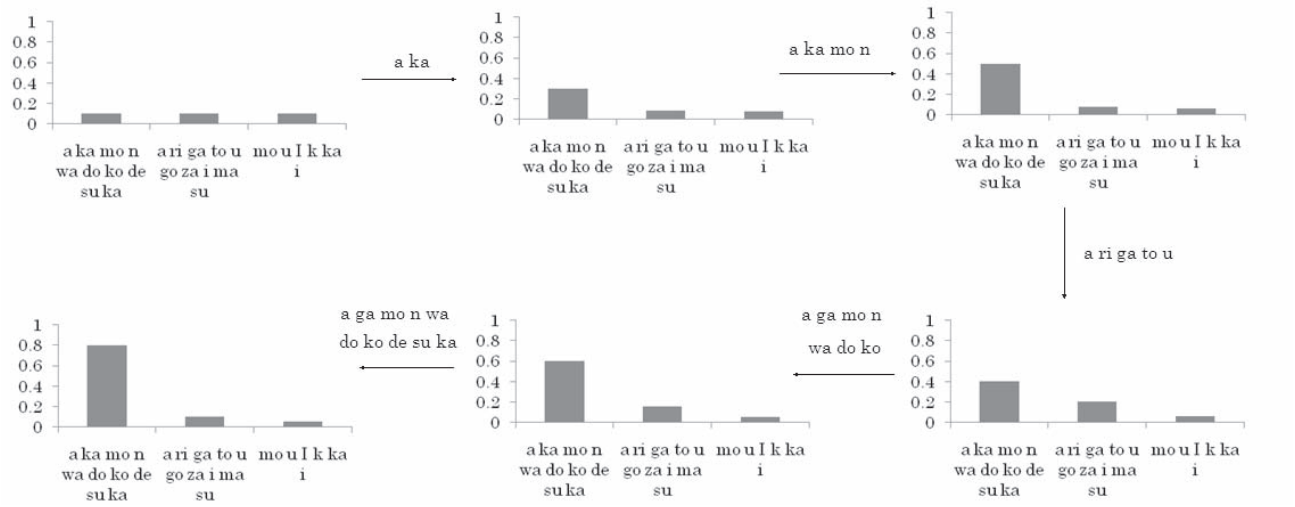

Fig. 2. Example of belief update from incremental speech recognition result

$$\Delta t = \frac{L_{speech} - L_{IU}}{s_{speech}} \tag{3}$$

At each time-step, if early determination is made, Δt will be calculated. As long as Δt is less than the delay of inside speech processing, the system starts to generate according to the determined speech. From the user's point of view, the system's response is just in time.

IV. Experimental Evaluation

A. Dialog Task

A testbed simulated dialog management problem is used to confirm the performance of the proposed method. In this domain, a total of 50 speeches can be decomposed into a total of 399 incremental units. The system has 51 actions, including a wait action and 50 determing actions. The incremental speech recognition result is obtained every 300(msec).

B. Early Rate and Correct Rate

Early rate is defined to evaluate how early the system can determine the user's utterance. At the timing when the system determines the user's speech, let L_{IU} be the length of the incremental unit (for example mora) and L_{speech} be the length of the speech which the agent estimates. Early rate can be calculated as:

$$P_{early} = \frac{L_{IU}}{L_{speech}} \tag{4}$$

Correct rate is also used to check whether dialog policy the system learned is correct. Let $T_{correct}$ be the times that the response is correct and T_{total} be the total times of evaluation. Thus, the correct rate can calculated as:

$$P_{correct} = \frac{T_{correct}}{T_{total}} \tag{5}$$

C. Experiment with a Simulator

The goal of this experiment is to confirm whether the proposed POMDP framework works well and to check the possibility of early determination with the incremental speech recognition results.

A user simulator that can output the incremental speech recognition result is used in the POMDP framework, and Q-learning algorithm is used for reinforcement learning. The error rate of speech recognition is selected among 0, 10, 20, 30, 40 and 50 (%), and the speaking rate is selected among 4, 5, 6 and 7 (morae/second). In the vocabulary there are a total of 50 kinds of phrases. 1000 simulated episodes were operated for optimizing the speech determination and 100 simulated episodes were used for evaluation.

Results are shown in Figure 3, where the x-axis is the error rate of speech recognition p_{err}, the solid line shows the correct rate of the decision and the doted line shows the early rate of the decision.

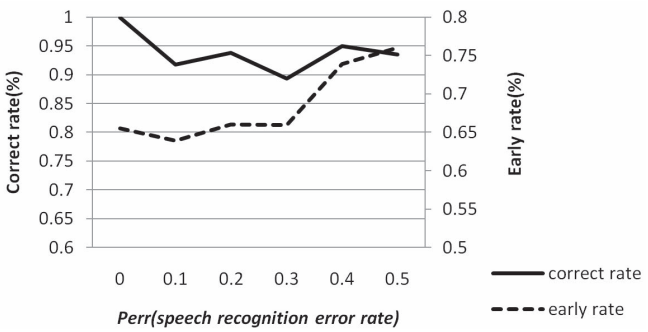

Fig. 3. Error rate of the speech recognition vs. early rate of the decision and the correct rate of the decision.

As the error rate of the speech recognition increases, the early rate increases from 0.65 to 0.75, which means the determination timing becomes later. Note that as the error rate increases from 0.1 to 0.5, the correct rate remains broadly constant between 0.89 and 0.96, but the early rate remains increasing. In another word, as error rate increases, the system

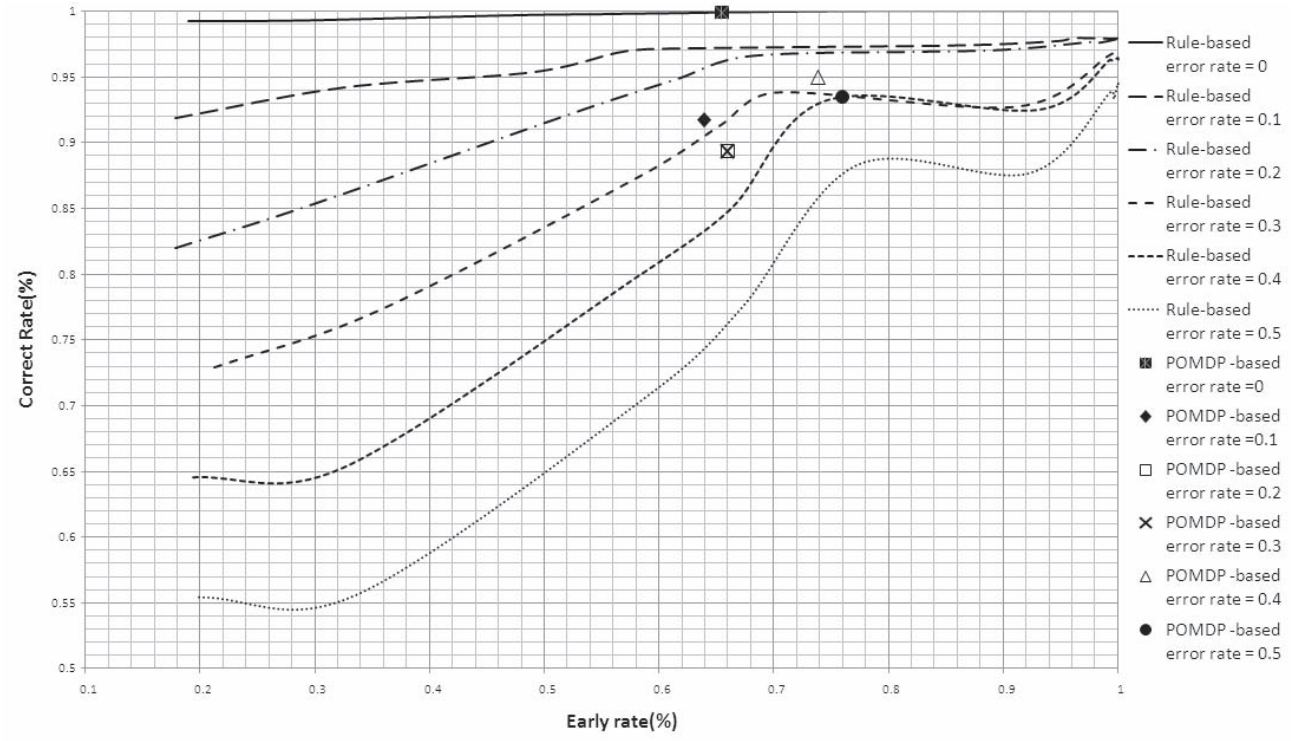

Fig. 4. Comparison between early rate and correct rate for rule-based model and POMDP-based model.

made the trade-off between the early rate and the correct rate. It is thought to be the reward function that affect the results because a correct determination can gain more reward than a fast determination and a wrong determination gets more penalty than a delayed determination.

To test the performance of POMDP framework, we created a rule-based dialog model as a baseline, which uses an early rate threshold to determine the timing. Figure 4 shows the relation between the early rate and the correct rate of the response for rule-based model and POMDP-based model. For both of two models, as the early rate increases, which means the determination timing becomes later, the correct rate increases. But as the error rate increases, the correct rate decreases. The result shows that the early determination policy learnt by POMDP-based model is similar to the rule-based model. Note that the POMDP-based model outperforms the baseline for p_{err} = 0.3, 0.4 and 0.5.

D. Implementation on Spoken Dialog System

A spoken dialog system which uses the incremental speech recognition result to determine the response timing is implemented based on Galatea toolkits (Figure 5), including the speech recognition engine Julius (Julian 3.5), a speech synthesis modal GalateaTalk and a face synthesis modal GalateaFSM. The system is implemented on Windows and the dialog control is implemented by python. The learning result in the simulation mentioned above is used as the parameter of the POMDP.

For comparison, we prepared three different dialog management models for the system.

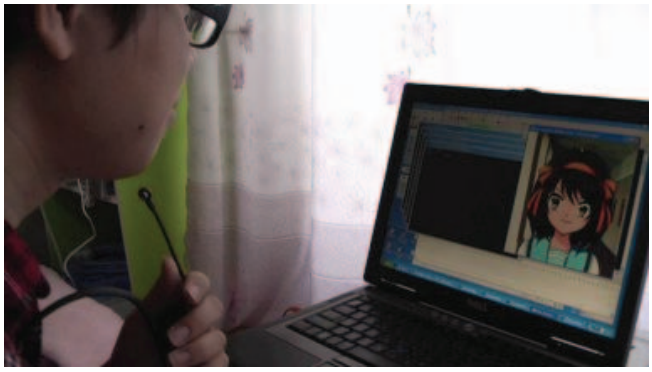

Fig. 5. Spoken dialog system based on Galatea toolkit

As a baseline, a traditional rule-based dialog model that only uses the final speech recognition result(starting recognition after the user's speech is over), which is called System A in the experiment. Both of the other two models are based on the proposed real-time POMDP model, but they are different in the response timing. System B will estimate the end timing of the user's speech from the incremental speech recognition result with the method introduced in the following passage, while System C generates the response as long as the user's utterance is early determined.

- System A: Use the final speech recognition result and start the response after user's speech ends.
- System B: Use the incremental speech recognition result and generate a response as soon as the user's utterance ends.
- System C: Use the incremental speech recognition result and generate a response as soon as the user's utterance

is determined.

In this experiment, there are a total of 10 participants, 8 male and 2 female. They were asked to talk to three different systems and they have to speak at least four sentences to each system. After that, the participants were asked to give a score ranging from 0(very bad) to 10(very good) for each system about the performance to find which one of the three systems was more human-like, which one performs more efficient dialog and which one is more attractive.

The average response delay and correct rate is shown in Figure 6. Since System A uses the final speech recognition result, the response timing is latest but the correct rate is highest. System C can response fastest but the correct rate is lowest.

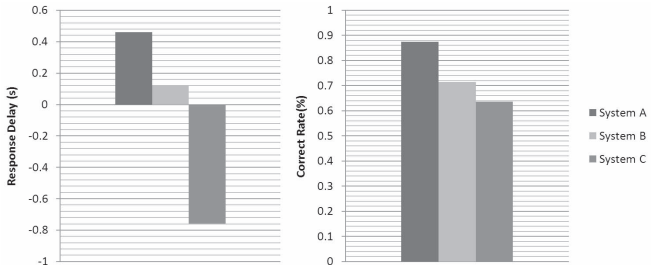

Fig. 6. Average response delay and correct rate for three systems

From the result, we can see that the proposed model outperformed the traditional model in all three aspects. System B is thought to be the most human-like and attractive system, because of the proper response timing. But system C can response very fast, so it is considered as the most efficient system.

It is difficult to judge the trade-off the system has made is good or bad, so we refer to Figure 7 for detail. From the result by analysis of variance, we found that there is no significant difference in human-likeness. About the efficiency, system A is significantly more efficient than system B. About the attractiveness, system B is significantly more attractive than system A and system C. Notice that system B is highly evaluated because of the end timing estimation.

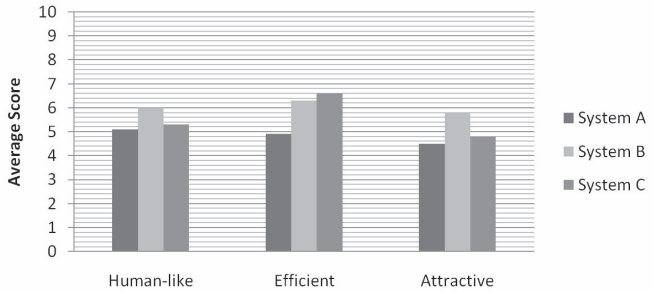

Fig. 7. Result of Questionnaire

V. Conclusion

In this paper, we proposed a method to use of incremental speech recognition based on POMDP framework. To deal with the error recognition result occurring in incremental processing, we use the sub-word vocabulary and reinforcement learning to enable a system to learn the determination timing. The simulation experiment shows the possibility of learning the trade-off between the timing and the accuracy of the determination. By implementing a spoken dialog system, we found that the system which can estimate the end timing of the user's speech is highly evaluated.

For future work, it is required to evaluate the proposed method in a large-scaled task domain. If the vocabulary is divided into mora unit, the recognition speed may be effected if the vocabulary is too large. This problem can be solved by using tree architecture or beam search algorithm. Further more, reinforcement learning is only carried out between a system and a simulator. The human-system learning is still needed to confirm the performance of the proposed method from now on.

Acknowledgment

I would like to thank my supervisor Prof. Shigeki Sagayama whose gentleness and thoroughness has provided me with great support over the years, for giving me such an interesting research theme. I would further like to thank all the students in Sagayama lab who had attended my experiment and provided me great suggestions in the seminar.

References

[1] S. Sagayama, "Why speech recognition is not used and how to make it into use(in japanese)," *Information Processing Society of Japan Report*, vol. 94, no. 40, pp. 23–30, 1994.

[2] S. Sagayama, T. Nishimoto, and M. Nakazawa, "Anthropomorphic spoken dialogue agent(in japanese)," *Information Processing Society of Japan*, vol. 45, no. 10, pp. 1044–1049, 2004.

[3] T. Nishimoto, M. Nakazawa, and S. Sagayama, "Nonverbal behavior modeling for spoken dialogue systems with anthropomorphic agents (in japanese)," *Pro. JSAI*, vol. 2C2, no. 01, 2004.

[4] T. Imai, H. Tanaka, A. Ando, and H. Isono, "Progressice early decision of speech recognition results by comparing most likely word sequences (in japanese)," *The Institute of Electronics, Information and Communication Engineers*, vol. J84, no. 9, pp. 1942–1949, 2001.

[5] G. Skantze and A. Hjalmarsson, "Towards incremental speech generation in dialogue systems," *In Proceedings of SIGdial*, 2010.

[6] R. Nishimura, N. Kitaoka, and S. Nakagawa, "A spoken dialog system for chat-like conversations considering response timing (in japanese)," *SIG-SLUD*, vol. 46, pp. 21–26, 2006.

[7] M. Nakano, K. Dohsaka, N. Miyazaki, J. Hirasawa, M. Tamoto, M. Kawamori, A. Sugiyama, and T. Kawabata, "Handling rich turn-taking in spoken dialogue systems," *Proc. of Eurospeech-99*, pp. 1167–1170, 1999.

[8] S. Fujie, K. Fukushima, R. Miyake, and T. Kobayashi, "Conversation system with back-channel feedback recognition and generation function (in japanese)," *SIG-SLUD*, vol. 45, pp. 41–46, 2005.

[9] T. Nishimoto, E. Iwata, M. Sakurai, and H. Hirose, "A voice interface system for exploratory search (in japanese)," *HCI-SLUD-127(2)*, pp. 9–14, 2008.

[10] K. Ishiguro, H. Sawada, and H. Sakano, "Multiclass boosting for sequence early classification and its applications," *MIRU2010*, 2010.

[11] A. G. Richard S., *Reinforcement Learning(in Japaanese)*. Morishita, 2000.

[12] N. Roy, J. Pineau, and S. Thrun, "Spoken dialog management for robots," *Proc. ACI*, 2000.

[13] E. Levin and R. Pieraccini:, "A stochastic model of computer-human interaction for learning dialog strategies," *Proc. Eurospeech*, 1997.

[14] J. D.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.