Full paper

Situated Spoken Dialogue with Robots Using Active Learning

Komei Sugiura *, **Naoto Iwahashi**, **Hisashi Kawai and Satoshi Nakamura**

National Institute of Information and Communications Technology, 3-5 Hikaridai, Seika-cho,
Soraku-gun, Kyoto 619-0289, Japan

Received 2 May 2011; accepted 30 May 2011

Abstract

In a human–robot spoken dialogue, the robot may misunderstand an ambiguous command from the user, such as ‘Place the cup down (on the table)’, thus running the risk of an accident. Although asking confirmation questions before the execution of any motion will decrease the risk of such failure, the user will find it more convenient if confirmation questions are not used in trivial situations. This paper proposes a method for estimating ambiguity in commands by introducing an active learning scheme with Bayesian logistic regression to human–robot spoken dialogue. We conduct physical experiments in which a user and a manipulator-based robot communicate using spoken language to manipulate objects.

Keywords

Active learning, confidence, robot language acquisition, object manipulation dialogue

1. Introduction

For practical reasons, most dialogue management mechanisms adopted for robots process verbal and nonverbal information separately (e.g., Ref. [1]). In these mechanisms, neither the situation nor previous experiences are taken into account when a robot processes an utterance, so there is a possibility that it will execute motions that the user had not imagined. In this study, we define the term ‘motion failure’ as an event occurring when a robot has executed an undesirable motion due to a recognition error. An earlier version of this work was presented in Ref. [2].

The goal of this study is to decrease the risk of motion failures. A simple solution to decrease this risk is to require confirmation utterances before motion execution, such as ‘You said ‘Bring me the cup’. Is this correct?’ However, there are two

* To whom correspondence should be addressed. E-mail: komei.sugiura@nict.go.jp

DOI:0.1163/016918611X595044

main hurdles to generating confirmation utterances: whether to confirm and how to confirm.

The problem of whether to confirm is a decision-making problem of whether a confirmation utterance should be made or not. Although making confirmation utterances before all motion executions would be simple and effective, this would seriously disrupt the dialogue. Specifically, the user will find it more convenient if confirmation questions were not made under trivial situations. In the field of spoken dialogue systems (SDS), the whether-to-confirm problem has received considerable attention in the context of error handling [3, 4].

The problem of how to confirm is the problem of paraphrasing the user's commands. The sentence 'Bring me a cup' is ambiguous when there are multiple cups and asking a confirmation question such as 'Do you mean the blue cup?' can disambiguate the sentence. Moreover, when direct and/or indirect objects are omitted (object ellipsis) in the user's utterance, such as 'Place the cup down (on the table)', it would be preferable to generate an appropriate description of the objects. The how-to-confirm problem deals with the mapping between language and physical/virtual objects and it has been widely explored in natural language generation (NLG) studies (e.g., Refs [5–7]). Lison *et al.* presented a model for priming speech recognition using visual and contextual information [8].

The robotics community has recently been paying greater attention to the mapping between language and real-world information, mainly focusing on motion [9–11]. Ogata *et al.* presented an application of recurrent neural networks to the problem of handling many-to-many relationships between motion sequences and linguistic sequences [12]. In the work of Takano *et al.*, a linguistic model based on the symbolization of motion patterns was proposed [13]. Moreover, we have proposed a robot language acquisition framework, LCore, that integrates multimodal information such as speech, motion and visual information [14].

In this study, we extend LCore with a dialogue management method based upon an adaptive confidence measure called the integrated confidence measure (ICM) function. Our key contributions are:

(i) A probabilistic model corresponding to each modality is used for speech understanding and speech generation. This enables us to introduce an active learning framework in human–robot dialogue. The model is explained in Section 3.

(ii) Bayesian logistic regression (BLR) [15] is used for learning the ICM function so that we can obtain the expected utility needed for generating confirmation utterances. The proposed method enables human–robot spoken interactions, while LCore [14] dealt with one-way interactions only from utterances to motions. The details are explained in Sections 4 and 6.

(iii) Active learning is used for selecting the optimal utterances, which effectively train the ICM function. The introduction of active learning is evaluated using likelihood criteria in Section 7.

2. Task Environment

In this section, we first explain the task environment, the task's three different phases and the hardware used for the experiments.

2.1. Object Manipulation Dialogue Task

Figure 1 shows the task environment used in this study. A user sits in front of a robot and commands the robot by speech to manipulate objects on the table located between the robot and the user. The robot is also able to command the user by speech to manipulate the objects. The objects used in the experiments are shown in Fig. 2a.

We assume that linguistic knowledge (e.g., phoneme sequence) and non-linguistic knowledge (e.g., motion and visual information) are learned by using LCore [14]. This knowledge is not given by the designer, but is learned through interaction with users. Specifically, the phoneme sequences of words such as 'red' and 'box' are learned. The visual concepts that those words represent are learned as well. Manipulation trajectories are learned by the method explained in Ref. [16]. Knowledge representation in LCore is explained in Section 3 in detail.

An object manipulation dialogue task runs as follows. First, the user commands the robot by speech to manipulate objects. Then, the user and the robot disambiguate the command by spoken dialogue. The task is accomplished if the robot executes the correct motion that is intended by the user. For example, in Fig. 1, suppose the user said '*Place-on* Barbabright red box' with the user intention shown as the trajectory. In this case, generating a confirmation utterance such as 'I'm placing

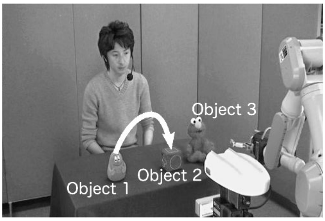
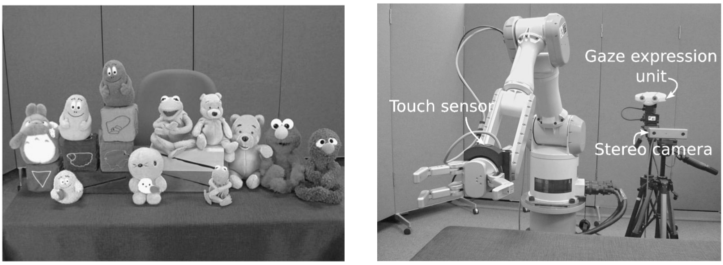


Figure 1. Example of an object manipulation dialogue task. In the original image, Objects 2 and 3 are red, and Object 1 is blue.



(a) (b)

Figure 2. (a) Objects used in experiments. (b) Robotic platform used for the experiments.

(it) on the small red box, is this OK?' would be appropriate. Finally, the task is accomplished if the robot moves Object 1 along the trajectory.

The robot may execute motions without confirmation, thus risking motion failure. If 'box' in the user utterance was misrecognized as 'Elmo', the robot might 'place Object 1 (Barbabright) on Object 3 (red Elmo)'. Although the objects shown in Fig. 1 are not likely to be damaged, it could be dangerous to execute unexpected motions with, for example, tableware.

The user utterances are supposed to have one verb and zero or more nouns/adjectives. Function words (e.g., 'on', 'with', 'the') are not supposed to be used in the commands. This is due to the fact that LCore is not yet able to learn them.

The main focus in this paper is on the many-to-many mapping between linguistic expressions (or symbols) and their referents in real-world situations. Therefore, the referent cannot always be disambiguated, even if speech recognition is perfect. Specifically, Object 2 shown in Fig. 1 can be referred to as 'box' or 'small red stuff', while, on the other hand, we can refer to Objects 2 and 3 as 'red'.

Based on this focus, we assume that neither pointing nor a graphical user interface (GUI) is allowed for disambiguation. One of the advantages of speech over text interfaces is that hands-free interaction is possible. In contrast, a text-based system using a keyboard interface could not allow the user to make commands if the user were holding objects in both hands.

2.2. Introduction of the Active Learning Phase

In the proposed method, a learning phase is introduced before the execution phase of the object manipulation task. The objective of the 'passive learning (PL)' phase is to reduce motion failures in the execution phase of the object manipulation task. In the PL phase, the user commands the robot to manipulate objects. A confidence measure is learned through interactions and the confidence is used for selecting the optimal confirmation utterance in the execution phase.

Although motion failures can be reduced in the execution phase, the introduction of the PL phase incurs motion failures in the PL phase itself. Therefore, we propose introducing another learning phase to reduce the motion failures in the PL phase. We call this phase the 'active learning (AL)' phase. In the AL phase, the robot commands the user to manipulate objects. The confidence measure is learned from the user's manipulation. The learned parameters of the confidence measure are used as the initial values in the PL phase so that the algorithm does not have to start learning from scratch. Figure 3 presents a schematic of the relationship of the three phases.

The advantage of introducing the AL phase is that motions by the robot are not required in the phase. Although there is a small chance that objects can be damaged by the user's manipulation, this possibility is negligible compared with possible damage by the robot's manipulation. Consequently, the proposed method provides a reasonable solution for reducing motion failures in the PL phase, as well as the execution phase.

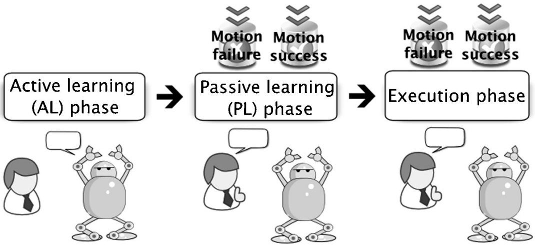


Figure 3. Three phases of the task.

2.3. Robotic Platform

Figure 2b shows the robot used in this study. The robot consists of a Mitsubishi PA-10 manipulator with 7 d.o.f., a 4-d.o.f. multi-fingered grasper by Barrett Technology, a microphone/speaker, a Point Grey Research Bumblebee 2 stereo vision camera, a MESA Swiss Ranger time-of-flight camera for three-dimensional distance measurement, and a head unit for gaze expression. Teaching signals can be provided by hitting a touch sensor on the grasper.

The visual features and positions of objects were extracted from image streams obtained from the stereo vision camera. The extraction and tracking of objects are done based on their color. The visual features have six dimensions: three for color (L*a*b* color space) and three for shape. The shape features, object area f_{area}, squareness f_{sq} and width–height ratio f_{whr} are defined as $f_{\text{area}} = wh$ and $f_{\text{sq}} = N_{\text{obj}}/wh$, where h, w and N_{obj} denote the object's height, width and number of pixels, respectively. For motion learning/recognition, the trajectories of objects' centers of gravity are used.

3. LCore Framework

The proposed method uses the LCore [14] framework as a base module. The functions of LCore are explained below.

3.1. LCore Overview

The proposed method uses the LCore [14] framework as a base module. When a user's utterance is input, LCore selects the optimal action based on an integrated probabilistic model trained by multimodal information. The integrated model Ψ consists of five modules: (i) speech, (ii) motion, (iii) vision, (iv) motion–object relationship and (v) behavioral context.

3.2. Motion Learning and Generation

A motion learning/generation method [16, 17] is used in LCore. In the method, the relative trajectory between the trajector (moved object) and the reference object is modeled with a hidden Markov model (HMM). The reference object can be the trajector itself or a landmark characterizing the trajectory of the trajector. In the case shown in Fig. 1, the trajector, reference object and reference point are Object 1, Object 2 and Object 2's center of gravity, respectively.

3.3. Speech Understanding in LCore

The word sequence of speech s is interpreted as the following conceptual structure:

$$z = [(\alpha_1, W_{\alpha_1}), (\alpha_2, W_{\alpha_2}), (\alpha_3, W_{\alpha_3})]$$
$$\alpha_i \in \{T, L, M\}, \quad i = 1, 2, 3,$$

where α_i represents the attribute of a phrase and is set as either trajector (T), landmark (L) or motion (M). The phrases W_{T}, W_{L} and W_{M} represent trajector, landmark and motion, respectively. For example, the user's utterance, '*Place-on Barbabright red box*', is interpreted as follows:

$$[(T, [\text{Barbabright}]), (L, [\text{red, box}]), (M, [\textit{place-on}])].$$

For motion concepts that do not require a landmark object, $z = [(T, W_{\mathrm{T}}), (M, W_{\mathrm{M}})]$.

The grammar G is a statistical language model that is represented by a set of occurrence probabilities of attribute α_i and words within a phrase. Here, word/attribute orders are learned using bigrams/trigrams.

Suppose that an utterance s is given under a scene O. O represents the visual features and positions of all objects in the scene. The set of possible actions A under O is defined as:

$$\begin{aligned} A &= \{(i_{\mathrm{t}}, i_{\mathrm{r}}, C_{\mathrm{M}}^{\langle j \rangle}) \mid i_{\mathrm{t}} = 1, \ldots, N_{\mathrm{O}}, i_{\mathrm{r}} = 1, \ldots, N_{\mathrm{R}}, j = 1, \ldots, N_{\mathrm{M}}\} \\ &\triangleq \{a_k \mid k = 1, 2, \ldots, |A|\}, \end{aligned} \tag{1}$$

where i_{t} denotes the index of a trajector, i_{r} denotes the index of a reference object, N_{O} denotes the number of objects in O, N_{R} denotes the number of possible reference objects for the verb $C_{\mathrm{M}}^{\langle j \rangle}$, and N_{M} denotes the total number of C_{M} in the lexicon.

Each module is defined as:

- Speech B_{S}. B_{S} is represented as the log-probability of speech s conditioned by z, under grammar G.

- Vision B_{I}. B_{I} is represented as the log-likelihood of a probabilistic model given Object i's visual features $\mathbf{x}_{\mathrm{I}}^{\langle i \rangle}$, where Object i is either the trajector i_{t} or the landmark i_{r}. Gaussian distributions are used for the probabilistic models.

- Motion B_{M}. B_{M} is defined as the log-likelihood of a probabilistic model given the maximum likelihood trajectory $\hat{\mathcal{Y}}_k$ for a_k conditioned by the trajector's position $\mathbf{x}_{\mathrm{p}}^{\langle i_{\mathrm{t}} \rangle}$, where i_{t} represents the trajector. HMMs are used for the probabilistic models of motions. The motion model λ is obtained from $\mathbf{x}_{\mathrm{p}}^{\langle i_{\mathrm{t}} \rangle}$, the landmark's position $\mathbf{x}_{\mathrm{p}}^{\langle i_{\mathrm{r}} \rangle}$, and motion index $C_{\mathrm{M}}^{\langle j \rangle}$ by a previously proposed imitation learning method [16, 17].

- Motion-object relationship B_{R}. Similar to B_{I}, B_{R} is represented as the log-likelihood of a probabilistic model given the visual features of Objects i_{t} and i_{r}.
- Behavioral context B_{H}. B_{H} represents the adequateness of Object i as the referent under the context $\mathbf{q}^{\langle i\rangle} = (q_1^{\langle i\rangle}, q_2^{\langle i\rangle})$. By using the parameter h_{c}, B_{H} is defined as:

$$B_{\mathrm{H}}(i, \mathbf{q}^{\langle i\rangle}; h_{\mathrm{c}}) = \begin{cases} 10 & (q_1^{\langle i\rangle} = 1) \\ h_{\mathrm{c}} & (\mathbf{q}^{\langle i\rangle} = (0,1)) \\ 0 & (\mathbf{q}^{\langle i\rangle} = (0,0)), \end{cases} \tag{2}$$

  where $q_1^{\langle i\rangle}$ and $q_2^{\langle i\rangle}$ stand for truth values representing the statements 'Object i is being grasped' and 'Object i was manipulated most recently', respectively. Minimum classification error (MCE) learning [18] is used to estimate h_{c}.

The integrated model Ψ is defined as the weighted sum of each module:

$$\begin{aligned}
\Psi(s, a_k, O, \mathbf{q}^{\langle i_{\mathrm{t}}\rangle}) = \max_z \{ & \gamma_1 \log P(s|z) P(z; G) && [B_{\mathrm{S}}] \\
& + \gamma_2 (\log P(\mathbf{x}_{\mathrm{I}}^{\langle i_{\mathrm{t}}\rangle} | W_{\mathrm{T}}) + \log P(\mathbf{x}_{\mathrm{I}}^{\langle i_{\mathrm{r}}\rangle} | W_{\mathrm{L}})) && [B_{\mathrm{I}}] \\
& + \gamma_3 \log P(\hat{\mathcal{Y}}_k | \mathbf{x}_{\mathrm{p}}^{\langle i_{\mathrm{t}}\rangle}, \mathbf{x}_{\mathrm{p}}^{\langle i_{\mathrm{r}}\rangle}, C_{\mathrm{M}}^{\langle j\rangle}) && [B_{\mathrm{M}}] \\
& + \gamma_4 \log P(\mathbf{x}_{\mathrm{I}}^{\langle i_{\mathrm{t}}\rangle}, \mathbf{x}_{\mathrm{I}}^{\langle i_{\mathrm{r}}\rangle} | C_{\mathrm{M}}^{\langle j\rangle}) && [B_{\mathrm{R}}] \\
& + \gamma_5 (B_{\mathrm{H}}(i_{\mathrm{t}}, \mathbf{q}^{\langle i_{\mathrm{t}}\rangle}) + B_{\mathrm{H}}(i_{\mathrm{r}}, \mathbf{q}^{\langle i_{\mathrm{r}}\rangle})) \}, && [B_{\mathrm{H}}]
\end{aligned} \tag{3}$$

where $\mathbf{x}_{\mathrm{p}}^{\langle i_{\mathrm{r}}\rangle}$ denotes the position of Object i and $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_5)$ denotes the weights of the modules. MCE learning [18] is used for the learning of $\boldsymbol{\gamma}$.

Inappropriate speech recognition results are re-ranked lower by using Ψ. There are several methods for re-ranking an utterance hypothesis (e.g., Ref. [19]). In contrast, information on physical properties such as vision and motion is used in Ψ, since object manipulation requires physical interaction.

4. Ambiguity Model Using ICM

4.1. ICM Function

The proposed method quantifies ambiguities in a user's utterances. In this subsection, we first explain the ambiguity criterion used in this study. Ambiguity is measured by the margin function, which represents the score difference between the first and second recognition candidates.

Given a context $\mathbf{q}$, a scene O and an utterance s, the optimal action $\hat{a}$ is obtained by maximizing Ψ:

$$\hat{a} = \operatorname*{argmax}_{a_k \in A} \Psi(s, a_k, O, \mathbf{q}). \tag{4}$$

We define the margin function d for the action $a_k \in A$ as the difference in the Ψ values between a_k and the action maximizing Ψ, $a_j (j \neq k)$:

$$d(s, a_k, O, \mathbf{q}) = \Psi(s, a_k, O, \mathbf{q}) - \max_{j \neq k} \Psi(s, a_j, O, \mathbf{q}). \tag{5}$$

Let a_l be an action that gives the second maximum Ψ value. When the margin for the optimal action $\hat{a}$ is nearly zero, the Ψ values of $\hat{a}$ and a_l are nearly equal; this means that the utterance s is a likely expression for both $\hat{a}$ and a_l. In contrast, a large margin means that s is an unambiguous expression for $\hat{a}$. Therefore, the margin function can be used as a measure of the utterance's ambiguity.

Now we define the ICM function by using a sigmoid function:

$$f(d; \mathbf{w}) = \frac{1}{1 + \exp^{-(w_1 d + w_0)}}, \tag{6}$$

where d is the value of the margin function for an action and $\mathbf{w} = (w_0, w_1)$ is the parameter vector. The ICM function is used for modeling the probability of success.

4.2. Learning the ICM Function

We now consider the problem of estimating the parameters $\mathbf{w}$ of the ICM function based on logistic regression. The ith training sample is given as a pair consisting of the margin d_i and teaching signal u_i. Thus, the training set $\mathbb{T}^{\langle N \rangle}$ contains N samples:

$$\mathbb{T}^{\langle N \rangle} = \{(d_i, u_i) \mid i = 1, \ldots, N\}, \tag{7}$$

where u_i is 0 (failure) or 1 (success). In experiments, u_i can be obtained through speech ('Yes/No') or a touch sensor.

Here, we assume that $f(d_i)$ estimates the probability where $u_i = 1$ given d_i. This assumption simplifies decision making explained in the Appendix. The parameters $\mathbf{w}$ can be estimated given the training set $\mathbb{T}^{\langle N \rangle}$ within the framework of logistic regression [20]. BLR [15] is used for obtaining the MAP estimate of $\mathbf{w}$. A univariate Gaussian prior with mean $m_i = 0$ and variance τ_i $(i = 0, 1)$ on each parameter w_i is used:

$$P(w_i | m_i, \tau_i) = \mathcal{N}(w_i; m_i, \tau_i) = \frac{1}{\sqrt{2\pi\tau_i}} \exp \frac{-w_i^2}{2\tau_i}. \tag{8}$$

5. Active Learning of the ICM Function

5.1. Utterance Generation Using the Active Learning Scheme

In the AL phase, the robot commands the user by speech to manipulate the objects. The ICM function is learned from the execution of the manipulation by the user. In this section we consider the problem of generating an utterance that is effective for learning.

The proposed method uses an active learning scheme to generate utterances in the AL phase. Applying active learning to dialogue management is the main originality of this paper and this has several advantages:

- The proposed method selects an utterance that reduces generalization error.
- As well as each module explained in Section 3, the active learning scheme can be integrated into a probabilistic framework.

Although some robotic studies use the term 'active learning' ambiguously, we adopt the term's definition that is common in machine learning. In this definition, active learning is a form of supervised learning in which inputs can be selected by the algorithm. The goal of active learning is to select the inputs that minimize generalization error. Generally speaking, an active learning scheme has a smaller generalization error than random sampling with the same number of training samples.

A schematic of the proposed method is illustrated in Fig. 4. Each function in Fig. 4 is explained below. The utterance candidate generator generates all possible word sequences for each action a_k and calculates their margin. Given a_k, the number of candidate word sequences, $N_\mathrm{S}(a_k)$, is calculated as:

$$N_\mathrm{S}(a_k) = \sum_{i=0}^{L_\mathrm{T}} N_\mathrm{V}^i \sum_{j=0}^{L_\mathrm{L}} N_\mathrm{V}^j, \tag{9}$$

where N_V denotes the number of words regarding appearances, and $(L_\mathrm{T}, L_\mathrm{L})$ denote the maximum lengths of $(W_\mathrm{T}, W_\mathrm{L})$. The above method is based on the brute-force search and it is not computationally efficient. Although we do not go into detail since the scope of the paper is an efficient algorithm, computational efficiency is of importance when the proposed method is applied to a larger set of words. After the search, the utterance candidate generator outputs the word sequences which have a positive margin, that is, $d > 0$.

The set of margins is input to the scoring function based on expected log loss reduction (ELLR) [21] and BLR [15]. Next, the optimal utterance is selected using

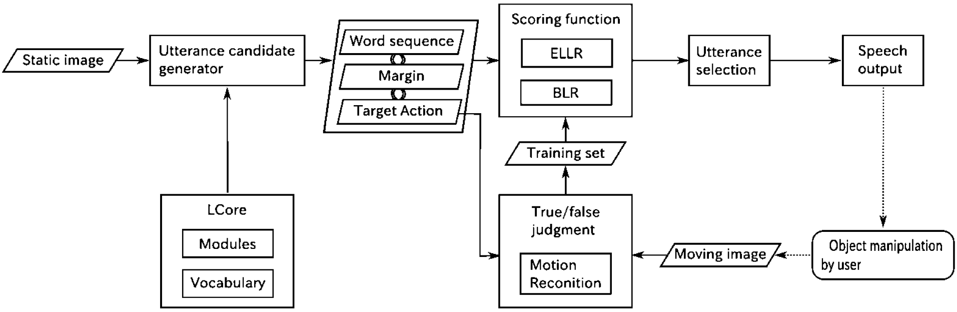
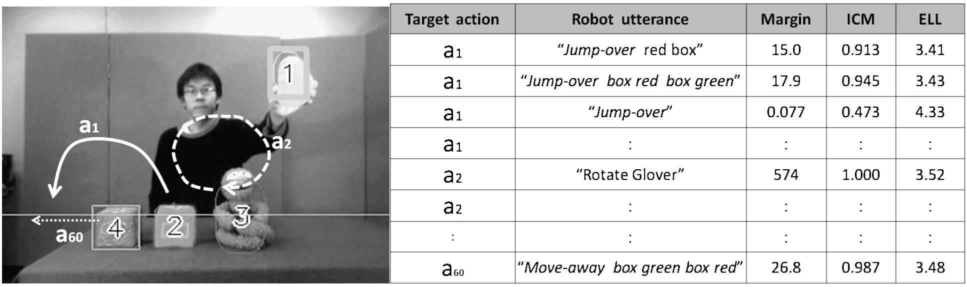


Figure 4. Schematic of the proposed method.



(a)

Target action	Robot utterance	Margin	ICM	ELL
a_1	*"Jump-over* red box"	15.0	0.913	3.41
a_1	*"Jump-over box red box green"*	17.9	0.945	3.43
a_1	*"Jump-over"*	0.077	0.473	4.33
a_1	:	:	:	:
a_2	"Rotate Glover"	574	1.000	3.52
a_2	:	:	:	:
:	:	:	:	:
a_{60}	*"Move-away box green box red"*	26.8	0.987	3.48

(b)

Figure 5. ELLR-based utterance generation.

ELLR and the utterance is output as a speech command to the user. The true/false judgment module recognizes the motion performed by the user and judges the result as 0 (false) or 1 (true). Here, the reference-point-dependent HMM [16] is used for motion recognition. The result is input to the training set and used by the scoring function.

Figure 5 illustrates an example of utterance generation by the proposed method. In Fig. 5a, three example target actions are illustrated by trajectories. In Fig. 5a, Object 4 was manipulated most recently. The first column of the table in Fig. 5b represents the target actions. The third, fourth and fifth columns of the table represent the margin d, ICM value $f(d; w)$ and expected log loss defined by (11), respectively.

In this case, 182 candidate utterances are generated by the utterance candidate generator. Among the candidates, the minimum ELL is 3.41, and so the utterance '*Jump-over* red box' is selected. The positive sample is given to the training set if the user action is recognized as the target action by the 'true/false judgment module'.

Basically, a training sample for learning the ICM function is obtained when a robot has executed a motion. Note that we assume the module for each modality and Ψ are used for speech understanding and speech generation. Based on this assumption, we can train the ICM function by using training data obtained when the robot commands the user by speech to manipulate an object.

5.2. *ELLR*

The proposed method selects the utterance that is most effective for learning the ICM function based on ELLR [21]. Among many criteria, uncertainty sampling [22] is the most basic method in active learning; however, it selects the sample with the most entropic prediction. In contrast, ELLR asks for labels on examples that, once incorporated in the training, will result in the lowest expected error on the test set [21].

Now, let $\hat{f}^{\langle N\rangle}(\cdot)$ denote the ICM function trained by the data set $\mathbb{T}^{\langle N\rangle}$. The log loss $L(\mathbb{T}^{\langle N\rangle})$ is defined as:

$$L(\mathbb{T}^{\langle N\rangle}) = \sum_{i=1}^{N} \{\hat{f}^{\langle N\rangle}(d_i) \log \hat{f}^{\langle N\rangle}(d_i) \\ + (1 - \hat{f}^{\langle N\rangle}(d_i)) \log(1 - \hat{f}^{\langle N\rangle}(d_i))\}. \tag{10}$$

In this case, $L(\mathbb{T}^{\langle N\rangle})$ can be regarded as the sum of entropy.

Let $S = \{s_j \mid j = 1, \ldots, |S|\}$ denote the utterance candidates in the scene O and d_j denote the margin linked with s_j. Here, S means the possible combinations of a word sequence that consists of learned words. We make S a finite set by limiting the length of a sequence. The proposed method selects the utterance that minimizes the expected log loss $E(\mathbb{T}^{\langle N\rangle}, d_j)$. $E(\mathbb{T}^{\langle N\rangle}, d_j)$ is defined as:

$$\begin{aligned} E(\mathbb{T}^{\langle N\rangle}, d_j) &= \hat{f}^{\langle N\rangle}(d_j) L(\mathbb{T}_{+}^{\langle N+1\rangle}) + (1 - \hat{f}^{\langle N\rangle}(d_j)) L(\mathbb{T}_{-}^{\langle N+1\rangle}) \\ \mathbb{T}_{+}^{\langle N+1\rangle} &\triangleq \mathbb{T}^{\langle N\rangle} \cup (d_j, 1) \\ \mathbb{T}_{-}^{\langle N+1\rangle} &\triangleq \mathbb{T}^{\langle N\rangle} \cup (d_j, 0). \end{aligned} \tag{11}$$

Accordingly, (11) takes into account the effect of a not-yet-obtained sample. In ELLR, $\hat{f}^{\langle N+1\rangle}(d_j)$ is trained in advance of obtaining the $(N + 1)$th sample. On the other hand, uncertainty sampling [22] does not take into account the effect of selecting the $(N + 1)$th sample.

6. Generation of Motions and Utterances

In Section 1, we explained the two problems we are trying to solve: whether to confirm and how to confirm. The problem of whether to confirm is a decision-making problem of whether a confirmation utterance should be made or not and the problem of how to confirm is the problem of paraphrasing the user's commands. A solution to the former problem is explained in the Appendix and that to the latter problem is explained below.

6.1. How to Confirm: Generation of Confirmation Utterances

The proposed method paraphrases object descriptions to make them more appropriate for the user and situation. Therefore, a confirmation utterance by the proposed method is not a mere speech recognition result. To paraphrase the user's utterances, words are inserted in the phrases W_{T} and/or W_{L}. The words are selected from the lexicon L based on the maximization of the margin d as follows.

Let $\psi(s, a_k, O, \mathbf{q}^{\langle i_{\mathrm{t}}\rangle}, z)$ be the weighted sum of modules. The differences between ψ and Ψ are such that ψ does not contain acoustic likelihood and ψ is not maximized with respect to z (*cf.*, (3)). We define d_z as the margin given z:

$$d_z(s, a_j, O, \mathbf{q}, z) = \psi(s, a_j, O, \mathbf{q}, z) - \max_{k \neq j} \psi(s, a_k, O, \mathbf{q}, z).$$

Suppose that the word set $\mathbf{c}' = \{c'_m \mid m = 1, \ldots, M\}$ is inserted in the phrase W (W_{T} or W_{L}). Here, W is a sequence of words: $W \triangleq c_1 c_2 \cdots c_{|W|}$, where $|W|$ represents the length of W. The optimal word set $\hat{\mathbf{c}}' = \{\hat{c}'_m \mid m = 1, \ldots, M\}$ and insertion-position set $\hat{\mathbf{p}} = \{\hat{p}_m \mid m = 1, \ldots, M\}$ are obtained as:

$$(\hat{\mathbf{c}}', \hat{\mathbf{p}}) = \operatorname*{argmax}_{c'_m \notin W, \mathbf{p}} d_z(s, a_j, O, \mathbf{q}, z). \tag{12}$$

Therefore, we obtain the following W' after the insertion:

$$W' = c_1 \cdots c_{\hat{p}_1 - 1} \hat{c}'_1 c_{\hat{p}_1} \cdots c_{\hat{p}_2 - 1} \hat{c}'_2 c_{\hat{p}_2} \cdots c_{|W|}. \tag{13}$$

These operations are performed for W_{T} and/or W_{L}, and finally we obtain an updated conceptual structure z':

$$z' = (W'_{\mathrm{T}}, W'_{\mathrm{L}}, W_{\mathrm{M}}). \tag{14}$$

Based on the above, utterance generation is summarized as:

Input Let $\langle O, \mathbf{q}, s \rangle$ be an input set; a scene, behavioral context and user's utterance.

(i) Generate trajectories for all items in the action candidate set A (see (1)), and obtain $\Psi(s, a_k, O, \mathbf{q})$ for every a_k.

(ii) Obtain the optimal action $\hat{a}$ according to (4). If $f(d(s, \hat{a}, O, \mathbf{q})) \geqslant \theta_0$ holds, then execute $\hat{a}$ and terminate. Otherwise, go to (iii).

(iii) Initialize the confirmation target set A' as $A' = A$.

(iv) Let the target action $a_j = \operatorname{argmax}_{a_j \in A'} f(d(s, a_j, O, \mathbf{q}))$. Initialize the number of inserted words, $M = 0$.

(v) Increment $M : M \leftarrow M + 1$ and generate z' according to (14).

(vi) If the updated margin d' satisfies $f(d') \geqslant \theta_0$, go to (vii). Otherwise, go to (vi(a))

(vi(a)) If there exists any word that can be added to z', then go to (v). Otherwise, go to (ix).

(vii) Make a confirmation utterance on a_j. Speech is synthesized according to z'. If W'_{T} or W'_{L} has no change from the original W_{T} or W_{L}, it is not included in the utterance.

(viii) If the user's response is positive, execute a_j and terminate. Otherwise remove a_j from A' and go to (viii(a)).

(viii(a)) If A' is empty, go to (ix). Otherwise, go to (iv).

(ix) Reject s by uttering 'Sorry, I cannot understand' and then terminate.

7. Experiments

7.1. Experimental Setup

To evaluate the proposed method, we conducted experiments in the following three phases: (i) AL phase, (ii) PL phase and (iii) execution phase.

The lexicon used in the three experiments contained 23 words (eight nouns, eight adjectives and seven verbs). Therefore, N_{M} was set as $N_{\mathrm{M}} = 7$ and the number of words regarding appearances, N_{V}, was set as $N_{\mathrm{V}} = 16$. The user taught the names or properties of objects in Japanese (in this paper, the utterances are translated into English) by showing the objects to the robot. Unsupervised learning was used for obtaining the phoneme sequences of the words [14]. Those words had been grounded to the physical properties of objects and motions in the learning phase of the lexicon [14, 16]. Table 1 shows the $\boldsymbol{\gamma}$ values obtained by the learning method, which is explained in an earlier work [23].

For the evaluation, we first collected a database as follows. A subject was told to sit across the table from the robot, as shown in Fig. 1, and make utterances in Japanese to make the robot manipulate objects. Accordingly, 120 pairs of utterance s and scene O were obtained. Each pair was labeled with a^* (that is, the indices of {motion, trajector, landmark}). The average chance performance for all of the data was 2.4% and the average number of words contained in each utterance was 2.6. Hereafter, we call this database DB120. The flow that starts from S_{c}'s utterance and ends with S_{e}'s manipulation is called an episode, where $(S_{\mathrm{c}}, S_{\mathrm{e}}) =$ (user, robot) or $(S_{\mathrm{c}}, S_{\mathrm{e}}) =$ (robot, user).

7.1.1. Setup (i): AL Phase

The objective of Experiment (i) is to obtain the parameters $\mathbf{w}$ in an active learning scheme. The learned parameters were used in Experiment (ii). In Experiment (i), the robot commanded the user by speech to manipulate objects based on the proposed method.

For the baseline method, we used a method that selects an utterance randomly from generated utterances. The difference is summarized as follows:

- Proposed: the optimal utterance is selected from generated utterances based on ELLR.
- Baseline: the utterance is selected randomly.

Table 1.
Parameters of the Ψ function used in the experiment

Weight	γ_1	γ_2	γ_3	γ_4	γ_5
Value	1.00	0.75	1.03	0.56	1.88

The same number of training samples was given to each of the methods. DB120 was divided into four sets, so each subset containing 30 training samples was used for training.

The prior distribution of the parameter w_i was defined as a univariate Gaussian distribution. The parameters of the prior, or hyperparameters, were set as $(m_0, m_1, \tau_0, \tau_1) = (0, 1, 100, 100)$. The hyperparameters (m_0, m_1) were set as $(m_0, m_1) = (0, 1)$ to make the initial ICM function the same as the standard logistic sigmoid function. Let $(L_{\mathrm{T}}, L_{\mathrm{L}}, L_{\mathrm{M}})$ denote the maximum lengths of $(W_{\mathrm{T}}, W_{\mathrm{L}}, W_{\mathrm{M}})$. We set $(L_{\mathrm{T}}, L_{\mathrm{L}}, L_{\mathrm{M}})$ as $(L_{\mathrm{T}}, L_{\mathrm{L}}, L_{\mathrm{M}}) = (3, 3, 1)$.

7.1.2. Setup (ii): PL Phase

Experiment (ii) was aimed at evaluating the effectiveness of using the results of Experiment (i) as the prior distribution. Although the effectiveness remains unclear, since the assumption that the user and robot share the ICM function is not always true, we will clarify the advantages of the proposed method.

In Experiment (ii), 10 different subsets were randomly drawn from DB120. Each subset contained 30 samples and these were used for training. For each training set, another 30 samples were drawn from DB120 and used as a test set.

We compared the cases in which the parameter was obtained by the proposed/baseline method in Experiment (i). To evaluate these methods, we compared test-set likelihood, where 10 different combinations of a training and test set were used. The number of motion failures was also compared in Experiment (ii). We compared the average number of motion failures occurring from the first to the i_{c}th episodes, where i_{c} represents the episode in which a convergence condition regarding log-likelihood $\mathcal{L}$ was met. The convergence condition was defined as $\mathcal{L} > \mathcal{L}_\theta$, where $\mathcal{L}_\theta$ was the threshold. Although we continued the actual experiment after the convergence condition was met, the learning should be terminated at that point for efficiency.

7.1.3. Setup (iii): Execution Phase

Experiment (iii) was aimed at evaluating the decrease in the failure rate in the execution phase. In Experiment (iii), a subject had object manipulation dialogues with the robot. The training and test sets were obtained from another database DB100, which contained 100 samples. Half of the data were used as a training set and the other half were used as a test set. The parameters of the ICM function were trained by the training set (50 samples) and fixed in Experiment (iii).

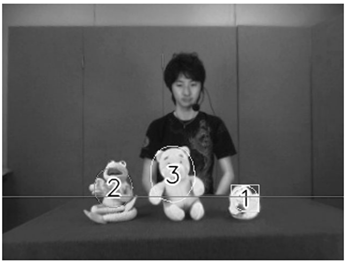
The dialogue was conducted as follows. First, a sample was drawn from the test set (50 samples) and the scene O was reconstructed. Then, the recorded utterance s for the sample was input to the system and a response was selected using the proposed method. If the response was a confirmation utterance, the user made a positive or negative response. An executed action a_k was compared with a^* to determine whether it was correct. An episode ended if the robot executed a motion or the utterance was rejected. In Experiment (iii), θ_0 was set to 0.7.

7.2. Experimental Results

7.2.1. Results (i): AL Phase

First, we address the qualitative results. Figure 6 illustrates an example dialogue between the subject (U) and the robot (R). In this case, the number of actions, $|A|$, was 45. This means that 45 pairs of word sequences and margins are input to the scoring function shown in Fig. 4. Note that the total number of possible word sequences is $\sum_{k=1}^{|A|} N_{\mathrm{s}}(a_k)$, where $N_{\mathrm{s}}(a_k)$ is obtained by (9). Among the pairs, the margin $d = 13.4$ gave the minimum $E(\mathbb{T}^{\langle N\rangle}, e_j)$ in (11). Here, the utterance linked with the margin $d = 13.4$ was '*Jump-over Pooh-doll Kermit* (Make the Pooh doll jump over Kermit)'.

In Fig. 7, the margin of the optimal utterance is plotted against episode that represents the number of robot utterances. In Fig. 7, the dotted line shows an episode in which a motion failure by the user has occurred. Here, we define a motion failure by the user as a case in which $\hat{a} \neq a^*$, where $\hat{a}$ and a^* denote the action taken by the user and the target action, respectively. From Fig. 7, we can see that an ut-



[Situation: Object 1 was manipulated most recently]
R: *Jump-over Pooh-doll Kermit.*
U: (The user makes Object 3 jump over Object 2.)

Figure 6. Dialogue example in the AL phase. The correct action is to make Object 3 (Pooh-doll) jump over Object 2 (Kermit).

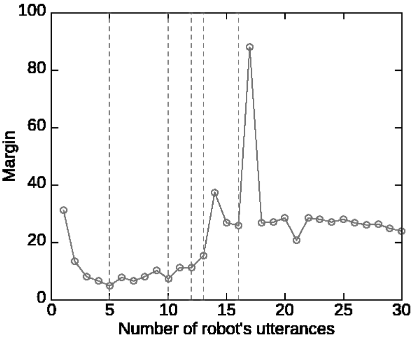


Figure 7. Margin of the optimal utterance. The dotted line shows an episode in which a motion failure by the user occurred.

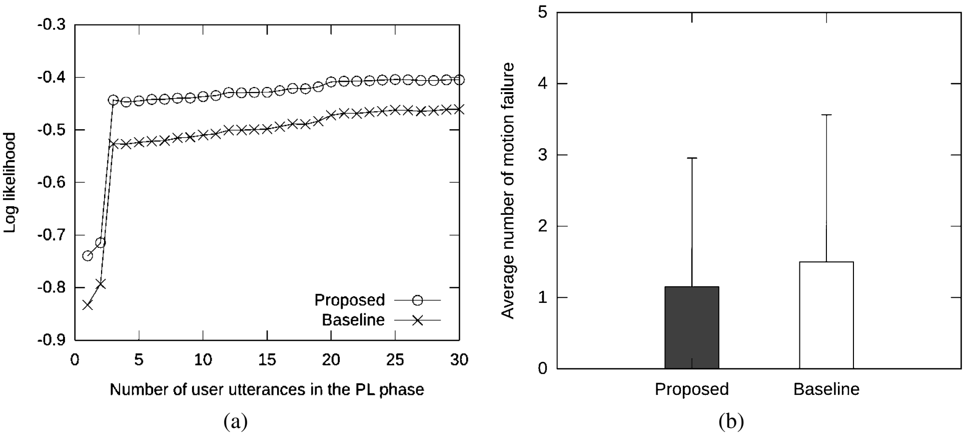


Figure 8. (a) Test-set log-likelihood by the proposed method and the baseline. (b) Average number of motion failures. The average i_c for the proposed method was 6.35, while that for the baseline was 8.25.

terance with a larger margin is selected at the $(i^* + 1)$th episode compared with the i^*th episode, where i^* represents the episode in which such motion failure occurred. This means that an utterance with less ambiguity is selected at the $(i^* + 1)$th episode.

7.2.2. Results (ii): PL Phase

Now we quantitatively investigate the effectiveness of using the parameters **w** obtained in Experiment (i) as the prior distribution. Figure 8a shows the average test-set log-likelihoods of the proposed method and the baseline. The lines show the average log-likelihood, where 10 different combinations of training and test sets were used. The log-likelihood is normalized by the size of the test set. From Fig. 8a, we can see that the performance of the proposed method is better than that of the baseline.

Figure 8b compares the average number of motion failures occurring from the first to the i_cth episodes, where $\mathcal{L}_\theta = 0.5$. From Fig. 8b, we can see that the number of motion failures could be reduced by using the proposed method if we terminate the learning at the i_cth episode. Specifically, the average number of motion failures was 1.15 for the proposed method and 1.5 for the baseline. From a significance test with 10% significance level, we obtained $p = 0.065$. Thus, the reduction in the number of motion failures supports the effectiveness of the proposed method.

Finally, we investigated the effect of the threshold $\mathcal{L}_\theta$. Let N_1 and N_2 denote the number of motion failures by the proposed method and the baseline, respectively. In Fig. 9, $(N_1 - N_2)$ is plotted against $\mathcal{L}_\theta$. $(N_1 - N_2) < 0$ means that the proposed method outperforms the baseline. In Fig. 9, $(N_1 - N_2)$ is averaged over 10 different combinations of training and test sets. From Fig. 9, we can see that $N_1 - N_2 \leqslant 0$ holds under $-1 < \mathcal{L}_\theta < -0.1$. This also supports the effectiveness of the proposed method in reducing motion failures.

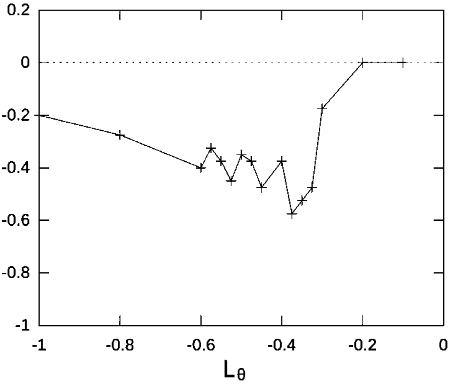


Figure 9. Evolution of the average difference in the number of motion failures against $\mathcal{L}_\theta$. The vertical axis represents the difference in the number of motion failures, $(N_1 - N_2)$. The horizontal axis represents $\mathcal{L}_\theta$. $(N_1 - N_2) < 0$ means that the proposed method outperforms the baseline in terms of the number of motion failures.

7.2.3. Results (iii): Execution Phase

First, we address the qualitative results. Figure 10 shows an example dialogue between the subject (U) and the robot (R). The ICM value is displayed in the circle at the top-right of the figure.

In Fig. 10, the ICM value of the optimal action $\hat{a}$ was small: $f(d) = 0.478 < \theta_0$. Consequently, a confirmation utterance was the optimal response. Therefore, the robot first asked whether '*green box*' was the trajector. Here, the word '*green*' was used to describe the major difference between Object 2 (the green box) and Object 3 (the blue box). In the second confirmation utterance, the word '*blue*' was inserted in the phrase W_T, since this gave the maximum margin. In contrast, the landmark was not mentioned in either generated utterance since no word insertion in W_L had a large influence on the ICM values.

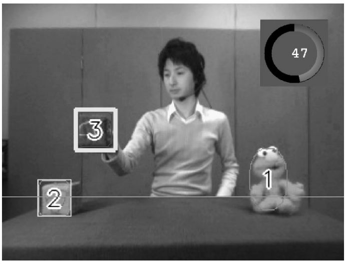
Next, we address the computation time. The average computation time for utterance selection (items (iv)–(vi) explained in the summarized algorithm in Section 6.1) was 1.73 s, where $\theta_0 = 0.999$. The computation was carried out on a Dell Precision 490 (Intel Xeon 2.0 GHz CPU E5355, 4 GB RAM) running the Linux operating system. We can see that the computation time is reasonable under the experimental setup.

Table 2 summarizes the quantitative results of decision making based on the ICM values. In Table 2, P_f, P_r, P_c and T_c represent the incorrect motion execution rate, rejection rate, confirmation utterance rate, and average number of confirmation requests, respectively:

$$P_\mathrm{f} = N_\mathrm{f}/(N_\mathrm{s} + N_\mathrm{f})$$

$$P_\mathrm{r} = N_\mathrm{r}/N_\mathrm{a}$$

$$P_\mathrm{c} = N_\mathrm{c}/N_\mathrm{a},$$



[Situation: Object 2 was manipulated most recently]
U: *Move-closer box Elmo.*
R: *Move-closer green box?*
U: No.
R: *Move-closer blue box?*
U: Yes.
R: (The robot moves Object 3 closer to Object 1.)

Figure 10. Dialogue example (ii). Motion execution with a confirmation utterance. In the original camera image, Object 1 is red, Object 2 is green and Object 3 is blue. The correct action is to move Object 3 (blue box) closer to Object 1 (Elmo).

Table 2.
Evaluation of decision making based on the ICM value

	θ_0			
	0 (speech-only)	0 (baseline)	0.7	0.999
P_f (%)	83.4	12.0	10.4	2.6
P_r (%)	0	0	4.0	24.0
P_c (%)	0	0	12.0	48.0
T_c	–	–	1.17	1.25

where N_a, N_s, N_f, N_c and N_r denote the number of all episodes, episodes in which correct actions were executed, episodes in which incorrect actions were executed, episodes in which confirmation utterances were generated and episodes in which the subject's utterances were rejected (i.e., no actions were executed), respectively. Here, $N_\mathrm{a} = N_\mathrm{s} + N_\mathrm{f} + N_\mathrm{r}$. T_c means the length of interactions; for example, there were two confirmation requests in Fig. 10.

In Table 2, the condition $\theta_0 = 0$ (speech-only) represents the results where the speech understanding (in this paper, 'speech understanding' represents the mapping from an utterance to action a_k) was conducted only by speech, where other modalities such as motion were not used. To show the complexity in reference disambiguation, the correct speech recognition results were given under $\theta_0 = 0$ (speech-only). From Table 2, we can see that P_f was more than 80%. This fact indicates that a speech-only approach would fail to solve this task even if there were no speech recognition errors.

Under the condition $\theta_0 = 0$ (baseline), the robot always executes a motion as a response to the subject's utterance. We can regard this condition as the baseline in which the proposed method was not used. P_f was 12% (6/50) under this condition. Table 2 shows that P_f was less than 12% under other conditions, where the proposed method was used. Specifically, we obtained $P_\mathrm{f} = 2.6\%$ (1/38) when $\theta_0 = 0.999$. This result clearly indicates that the proposed method outperformed the baseline in motion failure rate.

Table 2 reveals that confirmation utterances were generated in at most half of the scenes since P_c was less than 50% in all cases. Table 2 shows that T_c was approximately 1.2 under all conditions other than those where $\theta_0 = 0$.

Finally, we investigate the rejection rate. Table 2 shows that P_r increased with an increase in θ_0. The episodes that ended with rejection can be categorized into two groups: (i) utterances giving $f(d) \geqslant \theta_0$ could not be generated for the scenes and (ii) the subject could not understand the generated utterances. An example of (i) was a scene in which no combination of the learned words could identify the trajector and/or landmark. Specifically, one of identical green boxes could not be identified, since words for spacial relationships such as 'right' or 'below' were not learned in the experiments. An example of (ii) occurred when the generated utterance included the name of an object that did not exist in the scene due to uncertainties in image processing.

8. Discussion

8.1. Vocabulary Size

The robotics community has recently been paying greater attention to many-to-many mapping between language and real-world information [9–13, 24]. Reference [12] presents a translation method between motions and sentences by using recurrent neural networks with parametric biases (RNNPBs). The method proposed in Ref. [13] uses HMMs and language models for translation between motion capture data and sentences.

The vocabulary sizes in these studies are far smaller than in conventional SDS studies. Specifically, the vocabulary sizes in the above studies were 17 [12] and 24 [13]. Our study is inspired by these conventional studies and uses 23 words as vocabulary, which is comparable to their vocabulary sizes.

One of our future directions is to expand the vocabulary size, which is not easy. Setting a realistic target for future studies is important, and we think the vocabulary size will be several hundred.

Currently, approximately 100 words are used in a world-wide standardized benchmarking test for domestic robots, RoboCup@Home [25]. Specifically, in the General Purpose Service Robot test in RoboCup@Home 2010, the vocabulary contained 24 verbs (nine for navigation, five for object search and two others), 44 nouns

(20 object names, four category names and 20 human names) and 22 other words (function words and pronouns). However, no teams succeeded in executing the commands in 2010. Another hint is the target metric set by DARPA's Broad Operational Language Translation (BOLT). The target metric for Grounded Language Acquisition is to make future robots able to execute commands with a 90% completion rate while using 250 objects and 100 actions.

To apply the proposed method to a larger set of words, an efficient search method within a large set of candidate utterances is desirable. Although we do not go into detail since the scope of the paper is an efficient algorithm, there have been many conventional methods such as beam search. Another possibility is to eliminate unnecessary words from candidate words by using visual features. For example, we can eliminate the word 'Elmo' if all objects in a particular scene have a very small likelihood of being 'Elmo'.

8.2. Related Work in Other Research Fields

In the following we discuss related work from a broader perspective, so here the topic is not limited to object manipulation dialogue tasks.

In this study, multimodal information is used for estimating the probability that the user's utterances are successfully understood. On the other hand, confidence measures based only on speech have been proposed in the field of speech recognition and dialogue systems (e.g., Refs [4, 26]). Reference [27] is an extensive overview of confidence measures for speech recognition.

In SDS studies, confidence measures are often used for error handling. Komatani *et al.* used confidence for response selection in an SDS for hotel query tasks [3]. The method proposed in Ref. [28] used confidence based on speech recognition results and generated confirmation utterances by maximizing expected utility. Similar to our method, Ref. [29] uses confidence to model the probability of success in a public bus information system. The main differences of our method from that approach are (i) motion information is used for obtaining confidence and (ii) confidence is used for generating object descriptions.

In the proposed method, the ICM function integrates various information so that speech understanding errors can be reduced. Specifically, the proposed method can compensate for these errors caused by speech recognition by using visual and motion information. Similarly, Lemon *et al.* [19] used confidence for integrating several features obtained from speech recognition results. However, unlike their method, the proposed method uses visual and motion information.

From the viewpoint of maximizing utility, the proposed method is related to reinforcement learning-based SDS (e.g., Refs [30, 31]). Singh *et al.* [30] reported one of the pioneering studies applying reinforcement learning to dialogue management. Recently, the SDS community has been paying greater attention to applying reinforcement learning to dialogue management within the framework of partially observable Markov decision processes [31]. In Ref. [31], the action value is obtained by value iteration, while in the proposed method the expected utility $\mathbb{E}[R_i]$ is obtained based on the estimated probability of success.

In Section 2, we discussed the generation of object descriptions that disambiguate the user's commands. In an object manipulation dialogue task, there can be many candidate expressions for describing an object and the robot should choose a desirable object description. For example in Fig. 1, Object 2 can be referred to as 'red stuff', 'box' or 'red small square box'. However, 'red stuff' is ambiguous since there are two red objects and 'red square small box' is not concise.

The mapping between language and physical/virtual objects has been widely explored in AI and NLG studies [5, 32, 33]. A belief network-based confirmation method was proposed for disambiguating virtual objects such as cups and glasses [32]. In that study by Yamakata *et al.*, they achieved a success rate of 81.8% for object identification, which corresponds to $\frac{N_\mathrm{r}+N_\mathrm{f}}{N_\mathrm{a}}$ in this study. However, in their work, the attribute values of object models are discrete and given by the designer, such as 'Pattern = Floral' and, thus, there is a mismatch in applicability to our task since physical objects are used. The differences between Ref. [32] and this study are (i) we deal with disambiguation including multiple objects and motion, (ii) the attribute values of objects are not given by the designer, and (iii) the proposed method can be applied to real-world situations. The method proposed in Ref. [33] generates object descriptions for rectangles shown on a display. In Ref. [33], word categories are clustered by using an unsupervised learning method and thus the attribute values of objects are not given by the designer. However, that study dealt with neither expression containing verbs nor disambiguation through dialogue.

In the field of NLG, some corpora are used for generating linguistic expressions. The TUNA corpus was constructed to benchmark NLG systems in terms of the similarity between human- and machine-generated expressions. In the TUNA corpus, the inputs to the systems are given by images of furniture and related attribute values (XML text), and the number of attributes and attribute values are small. Specifically, the 'size' attribute of an object is either 'small' or 'large', and the 'type' attribute is either 'chair', 'sofa', 'desk' or 'fan'. These assumptions suffer from the symbol grounding problem, and thus do not match the robotic studies in which a camera and other sensors are used. In other NLG systems, the inputs are also given as text or figures on a GUI [5, 6]. Again, such input does not consider object manipulation dialogue tasks, in which a robot manipulates real-world objects. Therefore, it is not appropriate to apply the above methods to the problem this study tackles.

9. Conclusions

Safe interaction with users is a critically important requirement for assistive robots supporting users in everyday environments. In this paper, we have proposed a method that decreases the risk of motion failures in both the learning and execution phases.

One of the contributions of this study is that we integrated the learning techniques studied in different research fields within a probabilistic framework; the learning of motions has been mainly studied in the robotics community, while the learning of objects has been studied in the computer vision and AI communities. Most conventional studies have dealt with limited areas of robot language acquisition [14], such as phoneme learning, motion learning and visual concept learning. In contrast, we have shown that the proposed method allows physically situated spoken dialogue with robots by integrating learning modules.

Another contribution of this study is the introduction of active learning in a multimodal SDS. This enables the robot to generate utterances that are effective for learning. Some demo video clips of our system can be found at http://mastarpj.nict.go.jp/~ksugiura/video_gallery/lcore_al/index.html.

Acknowledgements

This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (C), 20500186, 2008, and by the National Institute of Informatics.

References

1. P. Dominey, A. Mallet and E. Yoshida, Real-time cooperative behavior acquisition by a humanoid apprentice, in: *Proc. IEEE/RAS Int. Conf. on Humanoid Robotics*, Pittsburgh, PA, pp. 270–275 (2007).
2. K. Sugiura, N. Iwahashi, H. Kawai and S. Nakamura, Active learning of confidence measure function in robot language acquisition framework, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Taipei, pp. 1774–1779 (2010).
3. K. Komatani and T. Kawahara, Flexible mixed-initiative dialogue management using concept-level confidence measures of speech recognizer output, in: *Proc. 18th Conf. on Computational Linguistics*, Saarbrücken, pp. 467–473 (2000).
4. D. Bohus and A. Rudnicky, Sorry, I didn't catch that! — An investigation of non-understanding errors and recovery strategies, in: *Proc. 6th SIGdial Workshop on Discourse and Dialogue*, Lisbon, pp. 128–143 (2005).
5. R. Dale and E. Reiter, Computational interpretations of the Gricean maxims in the generation of referring expressions, *Cogn. Sci.* **19**, 233–263 (1995).
6. P. Jordan and M. Walker, Learning content selection rules for generating object descriptions in dialogue, *J. Artif. Intell. Res.* **24**, 157–194 (2005).
7. K. Funakoshi, P. Spanger, M. Nakano and T. Tokunaga, A probabilistic model of referring expressions for complex objects, in: *Proc. 12th Eur. Workshop on Natural Language Generation*, Athens, pp. 191–194 (2009).
8. P. Lison and G.-J. M. Kruijff, Salience-driven contextual priming of speech recognition for human–robot interaction, in: *Proc. 18th Eur. Conf. on Artificial Intelligence*, Patras, pp. 636–640 (2008).
9. V. Krüger, D. Kragic, A. Ude and C. Geib, The meaning of action: a review on action recognition and mapping, *Adv. Robotics* **21**, 1473–1501 (2007).

10. Y. Sugita and J. Tani, Learning semantic combinatoriality from the interaction between linguistic and behavioral processes, *Adapt. Behav.* **13**, 33–52 (2005).
11. T. Inamura, I. Toshima, H. Tanie and Y. Nakamura, Embodied symbol emergence based on mimesis theory, *Int. J. Robotics Res.* **23**, 363–377 (2004).
12. T. Ogata, M. Murase, J. Tani, K. Komatani and H. G. Okuno, Two-way translation of compound sentences and arm motions by recurrent neural networks, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, San Diego, CA, pp. 1858–1863 (2007).
13. W. Takano and Y. Nakamura, Statistically integrated semiotics that enables mutual inference between linguistic and behavioral symbols for humanoid robots, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, Kobe, pp. 2490–2496 (2009).
14. N. Iwahashi, Robots that learn language: developmental approach to human–machine conversations, in: *Human–Robot Interaction*, N. Sanker (Ed.), pp. 95–118, I-Tech Education and Publishing, Vienna (2007).
15. A. Genkin, D. Lewis and D. Madigan, Large-scale Bayesian logistic regression for text categorization, *Technometrics* **49**, 291–304 (2007).
16. K. Sugiura, N. Iwahashi, H. Kashioka and S. Nakamura, Learning, generation, and recognition of motions by reference-point-dependent probabilistic models, *Adv. Robotics* **25**, 825–848 (2011).
17. K. Sugiura and N. Iwahashi, Learning object-manipulation verbs for human–robot communication, in: *Proc. Workshop on Multimodal Interfaces in Semantic Interaction*, Nagoya, pp. 32–38 (2007).
18. S. Katagiri, B. Juang and C. Lee, Pattern recognition using a family of design algorithms based upon the generalized probabilistic descent method, *Proc. IEEE* **86**, 2345–2373 (1998).
19. O. Lemon and I. Konstas, User simulations for context-sensitive speech recognition in spoken dialogue systems, in: *Proc. EACL*, Athens, pp. 505–513 (2009).
20. C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, Berlin (2006).
21. N. Roy and A. McCallum, Toward optimal active learning through sampling estimation of error reduction, in: *Proc. 18th Int. Conf. on Machine Learning*, Williamstown, MA, pp. 441–448 (2001).
22. D. Lewis and W. Gale, A sequential algorithm for training text classifiers, in: *Proc. Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Dublin, pp. 3–12 (1994).
23. N. Iwahashi, Interactive learning of spoken words and their meanings through an audio–visual interface, *IEICE Trans. Inform. Syst.* **91**, 312 (2008).
24. T. Kollar, S. Tellex, D. Roy and N. Roy, Toward understanding natural language directions, in: *Proc. 5th ACM/IEEE Int. Conf. on Human–Robot Interaction*, Nara, pp. 259–266 (2010).
25. Robocup@Home Rules & Regulations, available: http://www.ai.rug.nl/robocupathome/.
26. T. Kawahara, C. Lee and B. Juang, Flexible speech understanding based on combined key-phrase detection and verification, *IEEE Trans. Speech Audio Process.* **6**, pp. 558–568 (1998).
27. H. Jiang, Confidence measures for speech recognition: A survey, *Speech Commun.* **45**, 455–470 (2005).
28. T. Misu and T. Kawahara, Bayes risk-based optimization of dialogue management for document retrieval system with speech interface, in: *Proc. INTERSPEECH*, Antwerp, pp. 2705–2708 (2007).
29. D. Bohus, B. Langner, A. Raux, A. Black, M. Eskenazi and A. Rudnicky, Online supervised learning of non-understanding recovery policies, in: *Proc. IEEE/ACL Workshop on Spoken Language Technology*, Palm Beach, Aruba, pp. 170–173 (2006).
30. S. Singh, M. Kearns, D. Litman and M. Walker, Reinforcement learning for spoken dialogue systems, *Adv. Neural Inform. Process. Syst.* **12**, 956–962 (2000).
31. J. Williams and S. Young, Scaling POMDPS for spoken dialog management, *IEEE Trans. Audio Speech Lang. Process.* **15**, 2116–2129 (2007).

32. Y. Yamakata, T. Kawahara and H. Okuno, Belief network based disambiguation of object reference in spoken dialogue system for robot, in: *Proc. 7th Int. Conf. on Spoken Language Processing*, Denver, CO, pp. 177–180 (2002).

33. D. Roy, Learning visually grounded words and syntax for a scene description task, *Comp. Speech Lang.* **16**, 353–385 (2002).

Appendix: Whether to Confirm: Decision Making Based on Expected Utility

Let a^* be the action that the user intended to indicate by uttering s. In the execution phase, it is not desirable for the robot to execute an incorrect action a_k ($\neq a^*$) for safety reasons. A confirmation request to the user before the execution of an action can prevent the robot from executing an incorrect action. The ICM function can be used as a criterion for making a decision about whether a confirmation request is needed prior to executing the optimal action $\hat{a}$.

Now we consider the problem of making optimal decisions in response to the user's utterances. In the proposed method, the ICM value is used for the decision making. Specifically, the robot can make a confirmation utterance to the user whether $\hat{a}$ should be executed when the ICM value for $\hat{a}$ is smaller than a threshold.

Although using a threshold based on the ICM value is not the only solution, it has an advantage over a threshold based on the margin d. In the latter case, an appropriate threshold has to be chosen for each user, since γ is specific to each user. However, this approach's design cost is not small. On the other hand, a threshold based on the ICM value is applicable to other users. This is because the mapping between d and the probability of success is learned. In the following, we explain the method for optimal decision making for the whether-to-confirm problem.

We assume that the response is either the execution or confirmation of an action. Let b_1 be a response as a motion and b_2 be a response as a confirmation utterance. The ICM function $f(d)$ models the probability that the utterance is correctly recognized under the margin d.

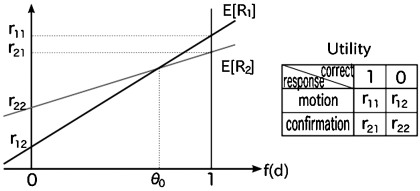
The expectation of utility R_i for the response b_i $(i = 1, 2)$, $\mathbb{E}[R_i]$, is estimated as:

$$\mathbb{E}[R_i] = r_{i1} f(d) + r_{i2}(1 - f(d)), \tag{A.1}$$

where r_{i1} and r_{i2} denote the utility for b_i in the cases of $\hat{a} = a^*$ and $\hat{a} \neq a^*$, respectively. For example, r_{11} represents the utility for executing the correct motion, and r_{12} represents the utility for executing the wrong motion. The utility matrix is shown in Fig. A.1.

Now, we assume $r_{12} < r_{2i} < r_{11}$ $(i = 1, 2)$. This magnitude relation (see Fig. A.1) implies that 'the utility of motion failure is smallest and the utility of motion success is largest'. Note that b_1 represents a motion response, and r_{12} represents the utility for b_1 where $\hat{a} \neq a^*$. Under this condition, the equation $\mathbb{E}[R_1] = \mathbb{E}[R_2]$ has the solution $f(d) = \theta_0$ as:

$$\theta_0 = \frac{r_{22} - r_{12}}{(r_{11} - r_{21}) + (r_{22} - r_{12})}, \tag{A.2}$$



response \ correct	1	0
motion	r_{11}	r_{12}
confirmation	r_{21}	r_{22}

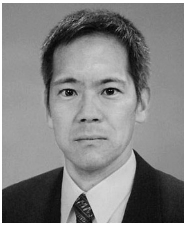
Figure A.1. Relationship between ICM value and expected utility.

where $0 < \theta_0 < 1$. Therefore, we can use θ_0 as the threshold for selecting the optimal response $\hat{b} = \text{argmax}_i\, \mathbb{E}[R_i]$.

About the Authors

Komei Sugiura is an Expert Researcher at the National Institute of Information and Communications Technology (NICT), Japan. He received his BE degree in Electrical and Electronic Engineering, and MS and PhD degrees in Informatics from Kyoto University, in 2002, 2004 and 2007, respectively. From 2006 to 2008, he was a Research Fellow at the Japan Society for the Promotion of Science, and he has been with NICT since 2008. His research interests include robot language acquisition, machine learning, spoken dialogue systems, sensor evolution and service robots.

Naoto Iwahashi received the BE degree in Engineering from Keio University, Yokohama, Japan, in 1985. He received the PhD degree in Engineering from Tokyo Institute of Technology, in 2001. In 1985, he joined Sony Corp., Tokyo, Japan. From 1990 to 1993, he was at Advanced Telecommunications Research Institute International (ATR), Kyoto, Japan. From 1998 to 2003, he was with Sony Computer Science Laboratories Inc., Tokyo, Japan. From 2004 to 2010, he was with ATR. In 2008, he joined the National Institute of Information and Communications Technology, Kyoto, Japan. His research areas include machine learning, spoken language processing, human–robot interaction, developmental multimodal dialog systems and language acquisition robots.



Hisashi Kawai received the BE, ME and DE degrees in Electronic Engineering from the University of Tokyo, in 1984, 1986 and 1989, respectively. He joined Kokusai Denshin Denwa Co. Ltd, in 1989. He worked for ATR Spoken Language Translation Research Laboratories, from 2000 to 2004, where he worked on the development of a corpus-based text-to-speech synthesis system. He worked again for KDDI R&D Laboratories Inc., from 2004 to 2009, where he was engaged in the management of research and development of speech information processing, speech quality control for telephone, speech signal processing and acoustic processing. Since April 2009 he has been with the National Institute of Information and Communications Technology, where he is working on the research and development of speech-to-speech tranlation systems. He is a Member of the IEICE, Acoustical Society of Japan and IEEE.

Satoshi Nakamura received his BS from Kyoto Institute of Technology, in 1981, and PhD from Kyoto University, in 1992. He was an Associate Professor in the Graduate School of Information Science at Nara Institute of Science and Technology, in 1994–2000. He was Director of ATR Spoken Language Communication Research Laboratories, in 2000–2008. He is an ATR Fellow. He launched the world's first network-based commercial speech-to-speech translation service for 3G mobile phones, in 2007. He is currently the Director General of Keihanna Research Laboratories and the Executive Director of the Knowledge Creating Communication Research Center, National Institute of Information and Communications Technology, Japan. He received Yamashita Research Award, Kiyasu Award from the Information Processing Society of Japan, Telecom System Award, AAMT Nagao Award, Docomo Mobile Science Award in 2007, Telecom System Award, ASJ Award for Distinguished Achievements in Acoustics, and the Commendation for Science and Technology by the Minister of Education, Culture, Sports, Science and Technology.