Spoken Dialogue Control Based on a Turn-minimization Criterion Depending on the Speech Recognition Accuracy

YASUDA Norihito and **DOHSAKA Kohji** and **AIKAWA Kiyoaki**
NTT Communication Science Laboratories
3-1 Morinosato-Wakamiya, Atsugi, Kanagawa, 243-0198 Japan
{yasuda, dohsaka}@atom.brl.ntt.co.jp, aik@idea.brl.ntt.co.jp

Abstract

This paper proposes a new dialogue control method for spoken dialogue systems. The method configures a dialogue plan so as to minimize the estimated number of turns to complete the dialogue. The number of turns is estimated depending on the current speech recognition accuracy and probability distribution of the true user's request. The proposed method reduces the number of turns to complete the task at almost any recognition accuracy.

1 Introduction

A spoken dialogue system determines user requests from user utterances. Spoken dialogue systems, however, can't determine a user's request only from an initial utterance, because there is a limitation to automatic speech recognition and recognition errors are unavoidable. Thus, most spoken dialogue systems confirm a user's utterance or demand the information that is lacking in order to determine user's request. Such dialogues for confirmation or demand between the system and the user are called "confirmation dialogues". Long confirmation dialogues are annoying, so more efficient confirmation is desirable. To measure the efficiency of the dialogue, we use the number of turns (exchanges), where of course, the fewer number of turns is better.

In practical applications, the system can accepts multiple types of user requests like "making a new appointment", "changing a schedule", and "inquiring about a schedule". If the user request type is different, the required information for determining the user request is also different. Sometimes the user request type is ambiguous due to recognition errors, and various types of user requests are possible. In such a case, it is important for the system to choose the type of user request it will confirm at first, since it will be useless to confirm items that are required for unlikely type of request.

The recognition accuracy affects the efficiency in other cases. For example, if there are multiple items to be confirmed, intuitively, it seems efficient to confirm all of them at once. However, the system must include candidates for all attributes in recognition vocabulary, which cause more recognition errors. Moreover, even though there is only one misrecognized item in confirmed items, the user might just say coldly "No", and the system cannot know that what are correct items.

Several efficient dialogue control methods have been proposed (Niimi and Kobayashi, 1996; Litman et al., 2000). But there is no previous works that take into account multiple types of user requests and recognition accuracy during confirmation, which changes what to be confirmed without domain-specific rules or training.

To prevent needlessly long confirmation dialogues even if the system can accepts multiple types of user request, our method estimates the expected number of turns to a certain use request type and the approximated probability distribution of user request types. The expected number of turns can be derived from the required vocabulary for confirmation and base recognition accuracy under certain vocabulary size.

2 Method

Overview First, we describe about a system to which we assume this method will be applied. The system has belief state which is represented by the set of attributes, their values, and the certainty of the values. The certainty is in [0 .. 1], and the certainty for the determined value is 1. That is, if the user replies "Yes" to the confirmation, the system changes the certainty for that value to 1. In practice, we can use the score from the recognition engine as this certainty. The system changes the recognition vocabulary according to the attributes to be confirmed at each confirmation. At any given time, the system either confirms or demands some attribute(s); it doesn't confirm and demand at the same time. Any values required in order to determine the user request are explicitly-confirmed without exception. Words that are irrelevant to the present confirmation are excluded from the recognition vocabulary. The system knows the base recognition accuracy under a certain vocabulary size, which is used to estimate the recognition accuracy.

Our method can be divided roughly into five parts; the first three parts are used to obtain the expected number of turns, granting that the user request type are already known, the fourth part is used to approximate the probability distribution of the user request, and the last part is used to decide the next action to be taken by the system.

The system needs to know only three sorts of information: 1) the vocabulary for each attribute; 2) the meaning constraints among words like "If the family name of the person is Yasuda, then his department must be accounting"; and 3) the required information for each type of user request like "To cancel an appointment; the day and the time are required". No other domain-specific rules or training are necessary.

Guessing the Recognition Accuracy Here we consider how to estimate the recognition accuracy during confirmation from confirmation target. Once attributes for confirmation are decided, the recognition vocabulary will consist of the words accepted by the attributes and general words for moving the dialogue along that are at least necessary to progress the dialogue such as "Yes", "No", etc. We call the recognition accuracy at this time the "attribute recognition accuracy".

We adopt the rule of thumb that the recognition error rate is in proportion to the square root of vocabulary size (Rosenfeld, 1996; Nakagawa and Ida, 1998). Thus, the approximated attribute recognition accuracy can be derived from the number of words accepted by the attributes.

Note that the attribute recognition accuracy can't be estimated beforehand, because the candidates for some attributes are dynamically change, as a result of the meaning constraints among words; if the value of one attribute is fixed, then candidates for other attributes will be limited to values that satisfy the constraints. Besides, the degree of limitation varies with the values. The relation between the user's family name and department is such an example.

Turn Estimation to Determine Some Attributes Next we consider how to estimate the expected number of turns for determining some attributes using the approximated attribute recognition accuracy.

We assume that the user's reply to the confirmation must contain the intention that corresponds to "Yes" or "No", and the intention must be transmitted to the system without fail. Then, the expected number of turns to complete confirming for some attributes is equal to the expected number of turns in the case that the confirmation is incorrect (i.e. misrecognized). Therefore, we can derive the number of expected turns to complete confirming T_c and demanding T_d for some attributes by the following expression:

$$T_c = \sum_{t=1}^{\infty} tr(1-r)^{t-1} = \frac{1}{r}$$

$$T_d = T_c + 1 = 1 + \frac{1}{r}$$

where r denotes the attribute recognition accuracy for attributes that are to be confirmed.

Turn Estimation to a Certain User Request Type Here we estimate the expected number of turns, granting that the type of user request is already known.

If the user request type is fixed, the required attributes for that type are also fixed. By comparing the belief state with these attributes, we can represent the required actions to determine the user request by a set of pairs made up of attributes and actions for the attribute (confirmation or demand). Once this set of pairs is given, we can choose the optimal plan, because we can estimate the expected turns of any permutations of any partitions of this set. The expected number of turns for this optiomal plan is used as the expected number of turns for a given user request type.

Probability Distribution of User Request Types Here, we consider how to estimate the relevance between the belief state and each user request types.

As it is hard to obtain the actual probability distribution, we define the degree of relevance between the belief state and each user request type as an approximation.

Let a_i, v_i, c_i be the i-th attribute, the value of a_i, and the certainty of v_i respectively. We define the relevance $Rel(S, R_j)$ between the belief state S and the user request type R_j as

for any v_i which can be accepted by R_j:

$$Rel(S, R_j) = \frac{1}{N_{G_j}} \sum \frac{c_i}{M_{v_i}}$$

where N_{R_j} denotes the number of required attributes in user request type R_j, and M_{v_i} denotes the number of user requests that accept the value v_i.

Choosing the Next Action Even if there is a highly possible user request type, choosing confirmation plan for it is not always best, if the expected number of turns for that request is very large. In such case, confirming another type of request that is easily confirmed and medium possibility may better.

We assume that when the user request type guessed by the system is not the real user request type, the number of turns required to know that the guess is incorrect is equal to the number of turns when the guess is correct and finish confirming the contents.

Let p_{R_i} be the probability of user request type R_i, and t_{R_i} be the expected number of turns to user request type R_i.

From permutations of request types, our method chooses the optimal order $a(1), a(2), \ldots, a(n)$ such that the expression $p_{R_{a(1)}} t_{R_{a(1)}} + p_{R_{a(2)}}(t_{R_{a(1)}} + t_{R_{a(2)}}) + \ldots + p_{R_{a(n)}}(t_{R_{a(1)}} + \ldots + t_{R_{a(n)}})$ is minimal. Then our method chooses the action that appears first in the optimal plan for request type $R_{a(1)}$ as the next action.

3 Experiments

We evaluated the proposed method by simulation. In the simulation, the system conversed with a simulated user program. Simulation with a simulated user enables rapid prototyping and evaluation (Eckert et al., 1998). The conversation was not done by exchanging spoken language, but by exchanging attribute-value pairs.

Simulated User Program The simulated user program works in the following steps:

1. Select a request. The request never changes throughout the dialogue
2. Tell the system the request or a subset of the request
3. Respond Yes or No if the system confirms
4. Give corrections at random if confirmation contains errors
5. Respond to the demand from the system
6. Tell the system that there is no information if the system refers to attributes with which the user is not concerned

Specification of Test Task We prepared a fictitious task for simulation. This task accepts six types of user demand. There are six attributes, and two of them have meaning dependence like the family name and department. The numbers of persons, family names, and departments are 3000, 1000, 300 respectively.

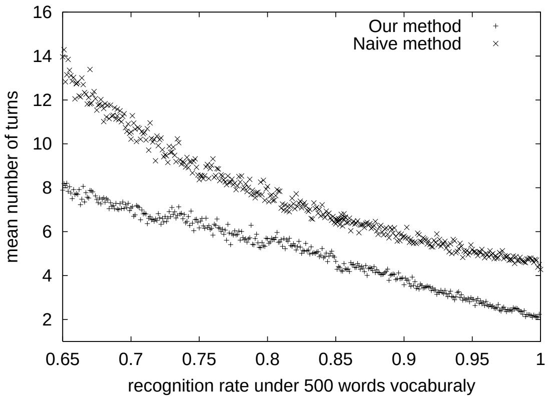

Figure 1: Average number of turns to complete a dialogue

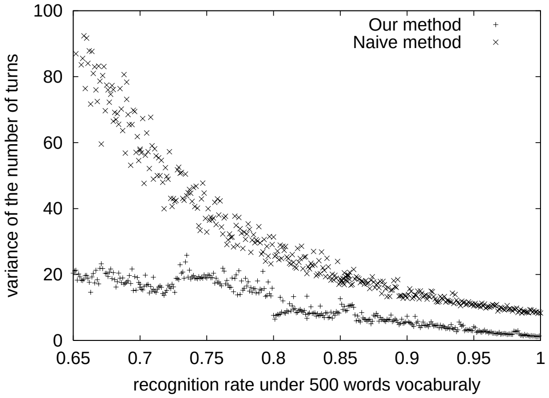

Figure 2: Variance of the number of turns to complete a dialogue

Comparison with a Naive Method For comparison, we prepared a naive confirmation dialogue control method, with the following specifications:

1. If the user request can be fixed uniquely and there are unbound attributes required for that request, demand those attributes one by one.
2. If there are values that are not confirmed, confirm them one by one.
3. If the user request type can't be fixed yet, demand a value for an attribute in the order of the number of user request types that require that attribute.

Experimental Results Figures 1 and 2 show the average number of turns and its variance out of 1000 diaglogue. We can see from these figures that our method can complete dialogues in shorter turns than other methods under various levels of recognition accuracy. In addition, the variance is small in almost every range, which illustrates the stability of our method.

4 Conclusion

A new dialogue control method is proposed. The method takes into consideration the expected number of turns based on the guessed recognition accuracy and the approximated probability distribution of user requests.

We don't have to write domain-specific rules manually by using this method. We can thus easily transfer domain of the system.

We evaluated our method by simulation. The result shows that it can complete dialogues in shorter turns than conventional methods under various recognition accuracy.

Acknowledgements

We thank Ken'ichiro Ishii, Norihiro Hagita, and all our colleagues in the Dialogue Understanding Research Group for useful discussions.

References

Wieland Eckert, Esther Levin, and Roberto Pieraccini. 1998. Automatic evaluation of spoken dialogue systems. In *TWLT13: Formal semantics and pragmatics of dialogue.*

Diane J. Litman, Michael S. Kearns, and Marilyn A. Walker. 2000. Automatic optimization of dialogue management. In *COLING.*

Seiichi Nakagawa and Masaki Ida. 1998. A new measure of task complexity for continuous speech recognition. *IEICE*, J81-D-II(7):1491–1500(in Japanese).

Yasuhisa Niimi and Yutaka Kobayashi. 1996. Dialog control stragey based on the reliability of speech recognition. In *International Conference on Spoken Language Processing*, pages 25–30.

R. Rosenfeld. 1996. A maximum entropy approach to adaptive statistical language modeling. *Computer, Speech and Language*, 10:187–228.