Example-based Spoken Dialogue System with Online Example Augmentation

Hiroya MURAO†,‡, Nobuo KAWAGUCHI†,§, Shigeki MATSUBARA†,§, Yukiko YAMAGUCHI†, Kazuya TAKEDA‡,§, Yasuyoshi INAGAKI**
† Digital Systems Development Center BU, SANYO Electric Co., Ltd., JAPAN
‡ Center for Integrated Acoustic Information Research, Nagoya University, JAPAN
§ Information Technology Center, Nagoya University, JAPAN
** Faculty of Information Science and Technology, Aichi Prefectural University, JAPAN
murao@hr.hm.rd.sanyo.co.jp

Abstract
In this paper, we propose a new method to expand an example-based spoken dialogue system to handle context dependent utterances. The dialogue system refers to the dialogue examples to find an example that is suitable to promote dialogue. Here, the dialogue contexts are expressed in the form of dialogue slots. By constructing dialogue examples with the text of utterances and the dialogue slots, the system handle context dependent dialogue. And we also propose a new framework of spoken dialogue, named “GROW architecture” that consists of the dialogue system and a Wizard-of-OZ (WOZ) system. By using the WOZ system to add dialogue examples via network, it becomes efficient to augment dialogue examples.

1. Introduction
With increasing computing power, the dialogue models based on the dialogue corpus are being proposed[1, 2]. In prior research, the authors have proposed a spoken-dialogue control technique using dialogue examples with the aim of performing flexible dialogue control during information retrieval dialogue[3]. This technique uses input speech data and supplementary information corresponding to the input speech such as search queries to form “dialogue examples” which are used to decide system action. Although many dialogue systems create search queries using keywords included in input utterances, it often happens that there is no valid keyword in the input because users’ requests are not clear or descriptive. In such cases, the dialogue system might fail to create correct search queries. Using our proposed framework, however, the system has the potential to be able to create search queries for such failed utterances by referring to the dialogue examples. Meanwhile, a system using this technique will not work effectively without a large volume of example data. In our previous work, the supplementary information was manually assigned to the collected text of human-to-human dialogue and that required considerable labor. We addressed this problem and proposed a new technique to collect dialogue example data from the dialogue performed between a human subject and a pseudo-spoken-dialogue system based on the Wizard of OZ (WOZ) scheme[3]. By this method, dialogue speech and supplementary information can be extracted from the log of WOZ system directly. But, there are two problems still outstanding. One is the problem that we have discussed only context independent utterances and the system can’t handle dialogue context. And the other is that it needs some processes to convert the log data of the WOZ system to the examples used in the dialogue system.

In the following sections, we will explain how to deal with the context dependent utterances in the example-based dialogue. Then, we will propose a new framework named “GROW architecture” that enables efficient augmentation of dialogue example data. Lastly we will describe the prototype system and experimental results.

2. Dialogue processing based on examples
2.1. Model of information retrieval dialogue
Given a scenario in which a human operator and a user execute dialogue for information retrieval and the operator searches an information database and returns information to the user, the dialogue can be modeled as shown in Fig. 1. The elements of this model are described below.

1. Request: The user tells the operator the contents of an inquiry.
2. Retrieval: The operator receiving the user’s request generates a query after referencing domain knowledge and current dialogue context and then processes the query indirectly by manipulating a search tool.
4. Update context: Dialogue context is updated.
5. Reply: The operator returns a reply to the user based on search results and dialogue context.
Setting up information flow in this way allows us to view operator behavior in the following way. Specifically, the operator in Fig. 1 makes two decisions in the dialogue process.

Decision 1: Generate a query after listening to user speech

Decision 2: Generate a reply after receiving search results

Here, an experienced operator would use more than just the superficial information obtained from user speech. To generate a query or reply that best suits the user’s need at that time, the operator would also make use of domain knowledge, dialogue context, and the search results themselves. In other words, this kind of dialogue processing can be viewed as a mapping operation from input information such as user speech and domain knowledge to output information such as a query. With this in mind, we considered whether a “decision” to guide such dialogue could be automatically performed by referring to actual examples of behavior manifested by an experienced human operator. In short, we decided to store a large volume of dialogue examples, i.e., mapping information, and then determine output information for certain input information on the basis of mapping information stored in similar dialogue examples.

2.2. Generation of queries and replies based on examples

The two “decisions” performed during the information retrieval dialogue can be expressed as a mapping between the following input and output information.

- For generating a query:
  - **Input**: User speech and dialogue context
  - **Output**: Query
- For generating a reply:
  - **Input**: User speech, dialogue context, and search results
  - **Output**: Reply

It is therefore necessary to save those items that cover such input/output information as well as context update information. The following subsections describes the procedure for generating a query or reply with respect to input speech by referencing an example database.

2.2.1. Query generation process

For the input speech, the system extracts an example which is the most similar to the input and the dialogue context at the time. It then adjusts the query in that example to fit the current conditions and generates a new query.

2.2.2. Reply generation process

The system performs a search based on the generated query and receives search results. It then extracts the most similar example from the example database with respect to the input speech, the dialogue context, and the search results at the time of receiving the search results. Finally, the system adjusts the reply in that example to fit the current conditions and generates a new reply.

3. Context handling in example-based spoken dialogue system

In this section, we describe our prototype of example-based spoken dialogue system. The task is to search for shop information inside an automobile. In our previous work[3], we discussed only context independent utterances. Here, we will go further and propose a new method for handling context dependent utterances.

3.1. Dialogue slot

In information retrieval dialogue, dialogue context information is considered to be included in the database query which is generated during the dialogue. We adopted dialogue slot[4] to express such information and handle dialogue context. An example of slot is shown in Fig. 2. Slot data consists of attributes like “(Shop) Name” and values like “Jumbo Burger”. Each field in slot is designed to be in one-to-one correspondence to each field in shop information database for search so that slot data can be transformed to a database query directly.

3.2. System configuration

The following describes the components of this system with reference to Fig. 3.

Dialogue Example Database (DEDDB): Consists of large volume of dialogue examples. Each dialogue example is a set of the following: 1) Transcribed text of user’s speech, 2) Contextual Slot (C-Slot): Context information expressed in slot form (ex. in Fig.2) before user’s input utterance is made, 3) Transitional Slot (T-Slot): A transition of context which occurs at the dialogue (ex. in Fig.2), and 4) Text of reply. Dialogue text is analyzed morphologically, and words essential in promoting the dialogue (e.g., shop name, food name) are assigned word class tags in the word class database.

Word Class Database (WCDB): Consists of words essential to the current task and classes given to them according to meaning. Word classes are determined empirically based on dialogue in the dialogue corpus.

Shop Information Database (SIDB): Consists of a collection of information on nearly 800 restaurants and shops in Nagoya. It’s implemented in SQL database.

Speech Recognition: Uses “Japanese Dictation Toolkit[5]”. The N-gram language model was created from the previously collected human-to-human dialogue corpus[6].

Query Generation: Extracts from the DEDDB the example closest to the current input speech and the dialogue context, and creates a query by referring the example.

Search Execution: Accesses the SIDB using the generated query and obtains search results.

Reply Generation: Extracts from the DEDDB the example closest to the input speech, the dialogue context and the search results, and creates a reply by referring the example.

Speech Synthesis: Outputs reply speech using a Japanese text-to-speech software.

Internal Slot (I-Slot): Contains context information of the system. The configuration of fields is the same as C-Slot and T-Slot.
Figure 3: Configuration of example-based dialogue system

Figure 4: Flow of dialogue process

3.3. Operation
The following describes the flow of system operation (Fig.4).

Step 1: Extracting similar example for query For a speech recognition result, the system extracts the most similar example from the DEDB(<1> in Fig. 4). The similarity is defined as the weighted summation of two scores below: 1) Similarity between input text and the text of user’s speech in the example. 2) Similarity between I-Slot and C-Slot in the example. Each similarity is calculated based on the number of matched words considering word class information.

Step 2: T-Slot Modification The T-Slot in the extracted example is modified on the basis of input speech and word class information (<2>).

Step 3: Updating Dialogue Context The system overwrites the modified T-Slot at the I-Slot and dialogue context is updated (<3>).

Step 4: Search From the updated I-Slot, the system creates SQL query(<4>) and executes search (<5>).

Step 5: Extracting similar example for reply After receiving the search result, the system once again extracts the most similar example from the DEDB(<6>). The following two similarities are used for the calculation of similarity in addition to similarities in <1>: 1) Similarity between number of search results and that in the example. 2) Similarity between modified I-Slot and a slot made by overwriting T-Slot at C-Slot in the example.

Step 6: Reply Modification Reply is created by modifying the reply in the extracted example on the basis of search result and word class information (<7>).

4. GROW Architecture
For the example-based dialogue system to work effectively, it is necessary to prepare a large volume of example data. In our previous work, we proposed to use WOZ (Wizard of OZ) scheme to collect example data efficiently [3]. In the information retrieval dialogue performed between a human subject and a pseudospoken-dialogue system based on WOZ scheme, the log information such as query or search results were saved and used to construct dialogue example database. But it needed some offline processes to convert the log data of the WOZ system to the examples used in the dialogue system. Here we will propose a new style of collecting dialogue examples named “GROW architecture” which enables online augmentation of dialogue examples. As Fig. 5 shows, dialogue system and WOZ system are connected to each other with network, and speech recognition result, slot data and reply created by the dialogue system are transferred to the WOZ system via the network. The “wizard”, or operator of the WOZ system, refers the received data and corrects them when necessary, then send them back to the dialogue system in the form of the dialogue examples.

5. Evaluation

5.1. Test set
We collected dialogue data using “GROW Architecture” to construct a test set. 66 utterances by 4 speakers were collected. In the data collection sessions, the subjects started dialogue after viewing one of 12 “Prompting panels” which contains pictures indicating user’s situation at that time (ex. in Fig.6). Each data in the test set consists of transcription of subject’s utterance \(^1\) and C-Slot provided by the wizard at that time. C-Slot is used to restore dialogue context. The test set is divided into 3 groups by contents, as in Table 1.

5.2. Dialogue example collection
To confirm the effects of the augmentation of dialogue examples by “GROW architecture”, four sets of DEDB were constructed and used in the experiment: 1) BASE database, as baseline, which were constructed from previously collected human-

\(^1\) As a first step of evaluation, we used transcribed text but speech recognition result to see the effectiveness of the dialogue process.
5.3. Process of evaluation

For each data in the test set, C-Slot is used to restore dialogue context and input utterance is applied to the dialogue system. And the query and the reply created by the system are subjectively evaluated.

5.4. Results

The result is shown in Table 2. On the whole, we can see that success rates for generating queries and replies increase by adding dialogue examples. Especially, the success rate of reply generation for class 3 reaches more than 80% and we can say that the method is effective for the utterances in class 3. By investigating the results in detail, we found some errors in the DEDB which are caused by wizard’s operational errors. Because of the errors, the performance of the system deteriorates. For example, in the query correct rate for class 1, the correct rate of the “B+O+C” is lower than the “B+O” although “B+O+C” has more examples. Therefore, it is apparent that we need to consider the error factor of the wizard.

6. Conclusion and future work

This paper has proposed a new method to handle context dependent utterances in example-based spoken dialogue, and a new architecture that consists of dialogue system and WOZ system to augment dialogue example effectively. The experimental result showed that the success rate for generating queries and replies improved as the number of the examples increase. But it was also apparent that we need to consider the error factor of the wizard.

Table 2: Evaluation result

<table>
<thead>
<tr>
<th></th>
<th>DEDB (# of examples)</th>
<th>test set (# of utterances)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query</td>
<td></td>
<td>ALL</td>
</tr>
<tr>
<td>correct rate</td>
<td>B</td>
<td>(138)</td>
</tr>
<tr>
<td>Reply</td>
<td></td>
<td>B+O (258)</td>
</tr>
<tr>
<td>correct rate</td>
<td>B+C (238)</td>
<td>66.7%</td>
</tr>
<tr>
<td></td>
<td>B+O+C (358)</td>
<td>74.2%</td>
</tr>
<tr>
<td>Query</td>
<td></td>
<td>B</td>
</tr>
<tr>
<td>Reply</td>
<td></td>
<td>B+O (258)</td>
</tr>
<tr>
<td>correct rate</td>
<td>B+C (238)</td>
<td>53.0%</td>
</tr>
<tr>
<td></td>
<td>B+O+C (358)</td>
<td>57.6%</td>
</tr>
</tbody>
</table>

In future, we plan to have another evaluation with larger volume of example data, with larger number of subjects. And we also plan to analyze the failed cases in more detail to clarify the cause of errors and to improve the algorithm.

7. Acknowledgments

This work is partially supported by a Grant-in-Aid for COE Research of the Ministry of Education, Culture, Sports, Science and Technology, Japan (No.11CE2005).

8. References