Example-based Spoken Dialog Processing for Guidance Robots

Cheongjae Lee, Seokhwan Kim, Minwoo Jeong, Sangkeun Jung
Donghyeon Lee, and Gary Geunbae Lee
Department of Computer Science and Engineering, POSTECH, Korea
e-mail:(lcj80, megaup, stardust, hugman, semko, gblee)@postech.ac.kr

Abstract − In this paper, we introduce a natural language interface for human-robot interaction (HRI) to develop building guidance robots. We have developed POSTECH Spoken Dialog System for Guidance Robots (POSSDS-GR). POSSDS-GR consists of automatic speech recognizer, spoken language understanding, dialog manager, natural language generator, and text-to-speech synthesizer. Each module is designed and implemented for making an effective and practical spoken dialog system. Our system is implemented with some dialog examples which are labeled to learn each module. We believe that our spoken dialog system will be a good interface to communicate between human and robot with natural language.

Keywords − Human-Robot Interaction, Spoken Dialog System, Example-based Dialog Modeling

1. Introduction

Today's robots require human-robot interaction (HRI) capabilities designed for the increasing variety of environments and contexts in which they operate. More human-like communication capabilities are necessary for robots operating in everyday settings such as home, office, shopping, and airport environments. In this case, HRI is essential in enabling robots to collaborate with humans to accomplish complex tasks. Recently, spoken dialog systems have been attractive to communicate between human and a large variety of robots because natural language is one of the most natural, efficient, and flexible means for people to communicate with each other. The objective for developing spoken dialog systems is to provide a natural way for any user to operate robots to achieve domain-specific goals such as information seeking and decision making.

In this paper, we have developed POSSDS-GR, a spoken dialog system for building guidance service to make guidance robots user-friendly and intelligent. In the past few years, spoken dialog systems have been used in various applications by their rapid increase in performance and decrease in cost [1]. There have been some studies on spoken language dialog interfaces for supporting to HRI capabilities in service robots [2].

POSSDS-GR consists of several sub-modules which are general in most of spoken dialog systems for other domains, such as automatic speech recognition module (ASR), spoken language understanding module (SLU), dialog management module (DM), natural language generation module (NLG) and text-to-speech synthesis module (TTS). In this paper, we describe the major methodologies behind POSSDS-GR.

2. POSSDS-GR: POSTECH Spoken Dialog System for Guidance Robots

POSSDS-GR consists of a set of appropriate modules that are designed to be connected to each other according to the order of execution. An overview of the system is shown in Fig 1. The overall system aims to output the synthesized spoken response corresponding to an input utterance spoken by the user.

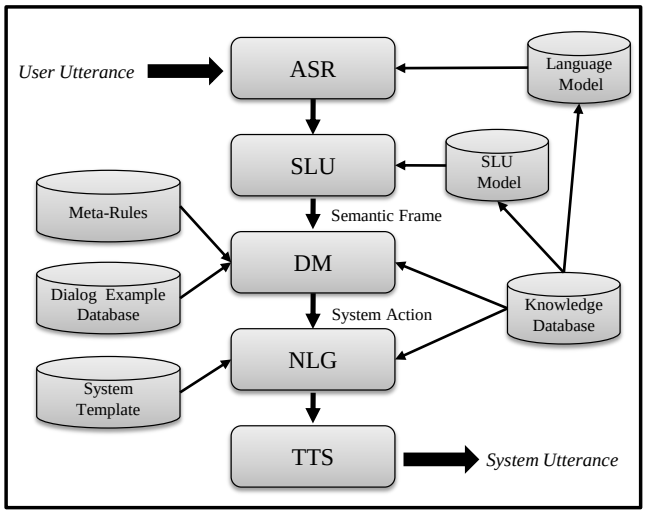

Fig. 1. Overview of POSSDS-GR system architecture

In order to handle the speech input, the ASR module is operated first, which recognizes the user utterance. The recognition result is used for the input of the SLU module. The SLU module extracts semantic concepts from the user utterance and constructs the pre-defined semantic frame by using extracted results. Then, the dialog manager predicts system responses based on observed evidences such as the semantic frame and the discourse history. The discourse history is a set of semantic frames in one dialog session. The result of the dialog manager is represented by corresponding system action tags selected from the pre-defined system action tag set. Each system action tag is defined by considering the expected response of the system at each situation of the dialog. The manner of the system response is determined by considering the system action tag in the NLG module. The NLG module produces

literal system utterances by assembling retrieved records from the knowledge database with system utterance templates which are determined by given system action tags. The final result of the overall system is produced by the TTS module, which synthesizes the robot's speech from the literal utterance. Each module in POSSDS-GR is based on domain-independent methodologies, with the consequence that it can be easily applied to other domains.

In following subsections, we describe detailed properties and methodologies of each module.

2.1 Automatic Speech Recognizer

The speech recognizer in POSSDS-GR was developed based on Hidden Markov Model Toolkit (HTK). The recognizer uses a pre-trained dialog acoustic model and adopts the domain-specific language model for guidance service. To build the language model, the candidate utterances that have high probability of being spoken by users are required. We generate the candidate utterances automatically by using the dialog examples in the existing example database and the retrieved results from the knowledge database. Table 1 shows an example of the automatically generated candidate utterances. Each dialog example record in the example database has the utterance itself as well as the tagged information on the named entities contained in the utterance. The candidate utterances are generated by replacing each named entity in the existing utterance with the corresponding entity in the retrieved results. The candidate utterances are used for building the domain-specific language model of the speech recognizer.

2.2 Spoken Language Understanding

The SLU module of POSSDS-GR was constructed by a concept spotting approach which aims to extract only the essential information for predefined meaning representation slots [3]. The semantic frame is made up of these slots including dialog act, main action, and component slots for the application domain. An example of the semantic frame for the guidance robot domain is shown in Table 2.

Each slot value is selected from the corresponding predefined tag set shown in Table 3. While the dialog act tag represents the grammatical function of the utterance, the main action tag mainly comprises its semantic function.

Table 1 Examples of the automatically generated candiadate utterances

An Existing Utterance
Where is a **toilet** on the **second floor**? [ROOM_NAME =toilet], [FLOOR=second floor]
Retrieved Results
[ROOM_NAME = **auditorium**], [FLOOR = **first floor**] [ROOM_NAME = **conference room**], [FLOOR = **third floor**]
Candidate Utterances
Where is an **auditorium** on the **first floor?** Where is a **conference room** on the **third floor?**

Due to the characteristics of the dialog act and the main action tag, each dialog act tag deals with domain-independent concepts, while each of the main action tag is defined for the domain-specific task of the dialog manager for the guidance robot domain. The component slots are used for representing named entities in the utterance.

We regarded the SLU problem as a classification problem, which can be solved by statistical machine learning frameworks. To build a statistical model for the SLU problem, we should prepare the training corpus containing utterances that have high probability of being spoken by users. We can easily create a training corpus by reusing the candidate utterances that are used for building the language model of the speech recognizer, which is referenced in the previous subsection.

2.3 Dialog Manager

To develop an effective and practical spoken dialog system, we have proposed the Example-Based Dialog Modeling (EBDM) for automatically predicting the next actions that the system executes. For an EBDM, we should automatically make an example database from the dialog corpus. The Dialog Example DataBase (DEDB) is semantically indexed to generalize the data in which the keys for indexing dialog examples can be determined according to state variables chosen by a system designer for domain-specific applications. The DEDB retrieves dialog examples which are similar to the current state. When there is no example, the dialog expert has some relaxation strategies according to the genre and the domain of the dialog. The expert can relax particular variables that have been earlier used to search the dialog example. The

Table 2 Example of the semantic frame

User Utterance	Where is a toilet on the second floor?
Dialog Act	WH_QUESTION
Main Action	SEARCH_LOC
Component Slots	[ROOM_TYPE=toilet] [FLOOR=second floor]

Table 3 List of the predefined slots in the semantic frame for the building guidance robot

Dialog act		
WH_QUESTION	YN_QUESTION	STATEMENT
ACCEPT	REJECT	REQUEST
Main Action		
SEARCH_LOC	GUIDE_LOC	INFO_LOC
SEARCH_PER	SEARCH_PHONE	SEARCH_MAIL
Component slot		
ROOM_NAME	ROOM_NUMBER	ROOM_TYPE
FLOOR	BUILDING_NAME	PER_NAME
PER_TITLE		

aim of each relaxation strategy is to exclude some constraints for a partial match. The examples from the partial match may be less similar to the current dialog situation. However, this relaxation strategy is required for solving the data sparseness problem. Once the relevant example or examples have been selected using the query keys, we can predict the next actions on the current dialog state. We should choose the best one by using the utterance similarity which includes the lexico-semantic similarity and the discourse history similarity. The lexico-semantic similarity is defined as a normalized edit distance between lexico-semantic utterances of the current user and retrieved examples. We also define the degree of the discourse history similarity which is a cosine measure between the binary vectors that are assigned with the value 1 if the slot is already filled, and 0 otherwise. Given two similarity measures, the utterance similarity can be expanded using interpolation with empirically defined weights for each application. Fig 2 illustrates an overall strategy of the example-based dialog modeling. From the retrieved examples, the dialog manager determines the system action tag from the pre-defined tag set shown in the Table 4.

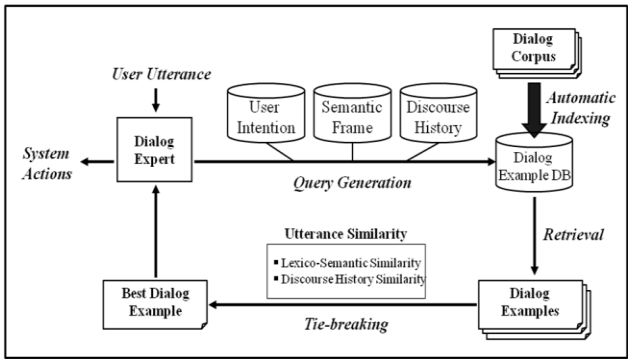

Fig. 2. Example-based dialogue modeling strategy

Table 4 List of the predefined system action tags

System Action		
Salutate	Guide	Ask_Guide
Inform_RoomNumber	Inform_Position	Inform_PerName
Inform_RoomName	Inform_RoomPhone	Inform_PerPhone
Inform_Floor	Inform_Description	Inform_Email

2.4 Natural Language Generator

The NLG module outputs the literal system utterances based on the system action tag and the system utterance template. Each system action tag has at least one utterance generating template which is constructed manually. The natural language generating task is advanced by filling slots in the template with proper contents such as retrieving results from the knowledge database, slot values in the semantic frame, and constituents in the discourse history. Table 5 shows an example of this task.

Table 5 Example of the system utterance generation

System Action Tag	Inform_Position
Utterance Template	The [ROOM_NAME] is located on the [ROOM_POSITION] on the [FLOOR]
Slot Values	[ROOM_NAME=toilet] [ROOM_POSITION=left side of the elevator] [FLOOR=second floor]
System Utterance	The toilet is located on the left side of the elevator on the second floor.

2.5 Text-To-Speech Synthesizer

Given a text form of system utterance, text-to-speech synthesizer makes a synthetic speech. To make synthetic speech natural, we used conversational style speech corpus to train our synthesizer. Our synthesizer is based on the concatenation of speech segments without prosodic modification. In other words, the quality of syntactic speech is purely dependent on selected segments. Therefore, we adopted several natural language processing technologies such as morphological and syntactic analysis so that appropriate targets are generated. Our synthesizer consists of two main components which are a natural language processing module and a prosody generation module. The natural language processing module contains a preprocessing module, a morphological and syntactic analysis module, and a grapheme-to-phoneme conversion module [5]. The prosody generation module contains a phrase break prediction module, a duration prediction module, an intensity prediction module, and a pitch contour generation module [6]. All the components of the prosody generation module are essential to synthesize natural synthetic speech.

3. Implementation

We implemented POSSDS-GR to be separated into three units of programs including the engine part and the graphical user interface (GUI) part. The engine part contains all of the above-mentioned modules such as ASR, SLU, DM, NLG, and TTS. The engine part was developed by using standard C++ library and can run under both Linux and Windows platform, but the GUI part of the system can only be operated under the Windows platform, because it is implemented by using Microsoft Visual Studio's MFC library. Fig 3 shows a screen shot of the POSSDM-GR, which is displayed through the GUI part of the system on the Windows platform.

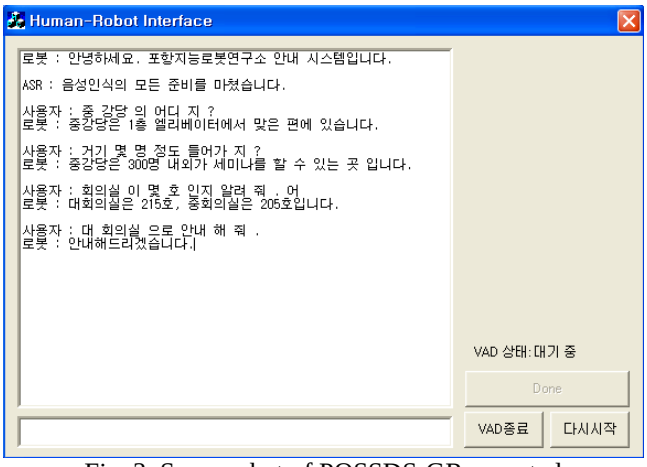

Fig. 3. Screen shot of POSSDS-GR operated

Acknowledgements

This work was supported by grant No. RTI04-02-06 from the Regional Technology Innovation Program of the Ministry of Commerce, Industry and Energy (MOCIE).

References

[1] J. Allen, D. Byron, M. Dzikovska, G. Ferguson, L. Galescu, and A. Stent, “Toward conversational human-computer interaction,” AI Magazine, vol. 22, no. 4, 2001, pp 27-37,.

[2] H. Asoh, T. Matsui, J. Fry, F. Asano, and S. Hayamizu, “A spoken dialog system for a mobile office robot,” in Proceedings of the European conference on Speech Communication and Technology, 1999, pp. 1139-1142.

[3] J. Eun, C. Lee, and G.G. Lee, “An Information extraction approach for spoken language understanding,” in Proceedings of the 8th International Conference on Spoken Language Processing, 2004, pp. 2145-2148.

[4] C. Lee, S. Jung, J. Eun, M. Jeong and G.G. Lee, “A Situation-based Dialogue Management Using Dialogue Examples,” in Proceedings of the international conference on acoustics, speech and signal processing 2006, vol. 1, 2006, pp. 69-72.

[5] J. Lee, S. Kim, and G.G. Lee, “Grapheme-to-phoneme conversion using automatically extracted associative rules for Korean TTS system,” in Proceedings of International Conference on Spoken Language Processing, 2006, pp. 1405-1508.

[6] S. Kim, J. Lee, and G. Lee, “Incorporating second-order information into two-step major phrase break prediction for Korean,” in Proceedings of International Conference on Spoken Language Processing, 2006, pp. 1487-1490.