EXAMPLE-BASED ERROR RECOVERY STRATEGY FOR SPOKEN DIALOG SYSTEM

Cheongjae Lee, Sangkeun Jung, Donghyeon Lee, and Gary Geunbae Lee

Pohang University of Science and Technology
Department of Computer Science and Engineering
Pohang, South Korea
{lcj80, hugman, semko, gblee}@postech.ac.kr

ABSTRACT

Error handling has become an important issue in spoken dialog systems. We describe an example-based approach to detect and repair errors in an example-based dialog modeling framework. Our approach to error recovery is focused on the re-phrase strategy with a system and a task guidance to help the novice users to re-phrase well-recognizable and well-understandable input. The dialog system gives possible utterance templates and contents related to the current situation when errors are detected. An empirical evaluation of the car navigation system shows that our approach is effective to the novice users for operating the spoken dialog system.

Index Terms— Error Handling, Error Detection, Error Recovery, Example-based Dialog Modeling

1. INTRODUCTION

The development of spoken dialog systems involves human language technologies which must cooperate in order to answer user queries. Since the performance in human language technologies such as automatic speech recognition and natural language understanding have been improved, this advance has made it possible to develop spoken dialog systems for many different application domains.

Nevertheless, there are inevitable bottlenecks for practical spoken dialog systems. One of the critical problems which must be considered by the dialog manager is the propagation of errors through prior modules. Errors in spoken dialog systems are prevalent due to speech recognition or language understanding errors. The recognition module must process the spontaneous speech with noisy environments. Consequently, the recognized utterance by this module inherently incorporate some errors. The recognition errors in practical systems are further aggravated by the large vocabulary and large variability of the user. The understanding module could also make its own errors which are mainly due to the lack of coverage of the semantic domain when faced with strange inputs. Finally, the semantic representation provided to the dialog manager might also cause system response errors. These errors appear across all domains and dialog genres.

To avoid these errors, a basic solution is to improve the accuracy and robustness of the recognition and understanding process. However, the spoken dialog system should also be able to adopt mechanisms for detecting and repairing potential errors at the conversational level since the development of the perfect conventional systems is impossible. The goal of error handling through human-computer communication is to maximize the user's satisfaction of using the system to guide for the repair of the wrong information by human-computer interaction. Error handling is a more serious issue for the users who are not experienced at spoken dialog systems. Empirically, we have observed that the experts (i.e. developers or experienced users) can operate these systems with a low error rate. In contrast, novice users suffer from more errors to handle the spoken dialog system. They show two critical problems for using the system. One of them is that they do not know the functionality and the coverage of information serviced by the system. The other is that they do not know what and how to say for operating the system at the current situation. However, the spoken dialog system deployed in real world should be broadly used by the novices to the experts.

In this paper, we introduce example-based error recovery strategies to be helpful for beginners to operate the spoken dialog systems. The basic idea of our approach is for the novices to receive guidelines what and how to say for achieving their goals. This is an extended idea from the Computer-Assisted Language Learning (CALL) [1]. In the case of CALL system, when students cannot proceed the current dialog scenario, a tutor gives hints for the students to speak appropriately at the next turn. Similarly, when the user wants to access information of interest using the spoken dialog systems, he/she can operate the system easily by learning how to use the system via example-based error recovery strategies. We begin by giving some related works of error handling in spoken dialog systems. After that we describe an example-based

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).

dialog modeling for our dialog manager. Then the example-based error recovery method is explained step by step. After that we show the experimental results for evaluation of our error recovery approaches. Finally, we draw conclusions and make suggestions for future works.

2. RELATED WORKS

Error handling for spoken dialog systems involves a number of stages that include error detection and error recovery [2] [3]. Several approaches have been proposed to detect and handle the errors generated in the recognition and understanding processes. The most commonly used measure for error detection is a confidence score [4][5]. The decision to engage these strategies is typically based on comparing the confidence score against the manually preset threshold. However, the confidence scores are not entirely reliable and dependent on the noisy environments and the user types. In addition, false acceptance, where the confidence score exceeds the threshold but really an error occurs, is more problematic as it may not be easy for the user to correct the system and put the dialog back on track. Thus, it can bring some problems at the level of dialog management. Recently, the n-best hypotheses of the recognition and understanding modules are considered to estimate the belief state with the uncertainty in the framework of Partially Observable Markov Decision Processes (POMDPs) [6].

At the level of the dialog manager, some error recovery strategies, i.e., explicit/implicit confirmation and re-phrasing, can be adopted to repair these errors. An explicit confirmation takes the form of a question that asks explicitly for confirmation of the main slots of the task (i.e. ”origin”, ”destination” and ”date” in the flight reservation system). This may be accompanied by a request to answer with ”yes” or ”no”. Dialog manager can also use an implicit confirmation in which the system embeds in its next question a repetition of its understanding of what the user said in the response to the previous question. Explicit and implicit confirmations are good strategies to repair the information which is not reliable by computing the confidence scores on various levels including the phonetic level, the word level, and the utterance level. In these cases, the user says a partial phrase or a short utterance to acknowledge and confirm the dialog state. However, the deficiency of context may makes new errors in recognizing and understanding the user's utterance. In addition, the distribution of user behaviors in coping with errors shows that users in the successful error recoveries use significantly more rephrasing than attempt to repair a chain of errors [7]. From these reasons, we believe that the re-phrase strategy is more successful to repair errors in the dialog manager. However, all of repeating the previous utterance cannot correct the errors for the system to manage the user's utterance. In particular, the novice users have potential problems of out-of-vocabulary (OOV) and out-of-utterance (OOU). In the view of novice users, the system should help to speak well-recognizable and well-understandable utterances at the current situation. Ideally, one of the best error recovery strategies is that the users can gradually learn how to operate the dialog system. In this paper, we propose the example-based error recovery strategy to achieve these goals for the users.

3. EXAMPLE-BASED DIALOG MODELING

Our error recovery strategy is implemented based on an Example-Based Dialog Modeling (EBDM) which is one of generic dialog modelings technology [8]. We begin with a brief overview of the EBDM framework in this section. We have proposed the EBDM for automatically predicting the next actions that the system executes inspired by the Example-Based Machine Translation (EBMT) [9]. The EBMT is a translation system in which the source sentence can be translated by the similar example fragments within a large parallel corpus without knowledge of the language's structure. We think that the idea of EBMT can be extended to determine the next system actions by finding the dialog examples within the dialog corpus. The system action can be selected by searching the similar user utterance with the dialog state which is defined as the relevant internal variables that affect the next system action.

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. Figure 1 illustrates how to map each utterance pairs (user-system utterances) on dialog corpus onto semantic records on DEDB. 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 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.

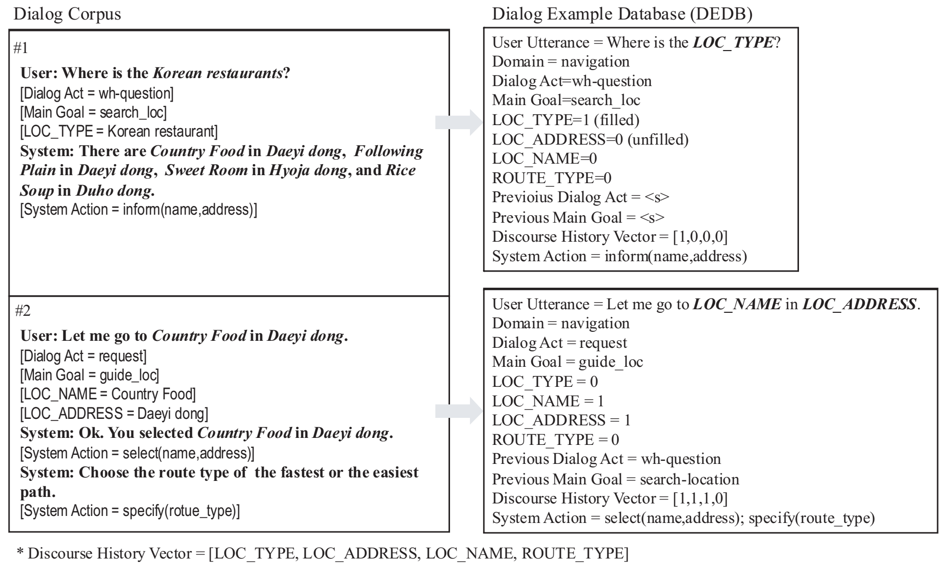

Fig. 1: Indexing scheme for dialog example database on car navigation domain.

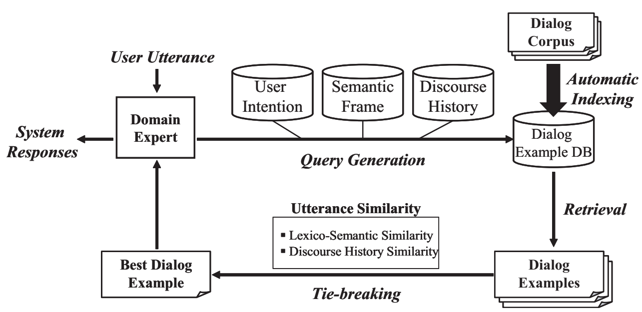

Fig. 2: A strategy of the example-based dialog modeling.

Figure 2 illustrates an overall strategy of the example-based dialog modeling. The main advantage of EBDM methodology is the ability to quickly produce a deployable dialog model for several applications. To date, it has been used to construct a large number of systems spanning multiple domains and genres [10].

4. EXAMPLE-BASED ERROR RECOVERY

In practice, three stages are required for a successful error recovery:(1) the ability to detect the potential errors, (2) a set of error recovery strategies, and (3) a mechanism for engaging these strategies at the appropriate time. In this section, we describe the errors of the EBDM framework and propose our method to overcome these errors using an example-based error recovery.

4.1. Error Detection of EBDM

With early error detection, the system detects that something is wrong in the user's current utterance and takes immediate steps to address the problem. Early detection of errors has mainly focussed on speech recognition errors and understanding errors. However, errors may occur at the stages of recognition and understanding as well as dialog management. In this paper, we more focus on late error detection at the level of dialog management. Although the current results of the recognition and understanding modules may incorporate some errors, our dialog manager initially attempts to search the similar examples and the contents by using the current dialog frame and discourse history. In this case, there are some situations for detecting potential errors of the dialog manager such as the following three cases:

- *No Example*: No dialog example is retrieved despite both exact and partial matches are used.
- *No Content*: No information is accessible to the domain knowledge database using the slot values of the current dialog frame.
- *No Slot*: The understanding module cannot extract any slot value from the user utterance.

The case of *No Example* means that the system cannot find similar examples to determine the next system action. When the recognition or understanding errors occur, an erroneous utterance is different from the dialog examples within the DEDB, that is, the utterance similarity falls below the threshold. We regard this situation as having potential errors since the utterance may be semantically or grammatically incorrect.

The basic idea of handling *No Example* error is that the system provides the utterance template which is pre-trained to build the modules.

The case of *No Content* occurs when the contents of domain knowledge database are not retrieved given current constraints. If the user does not know the slot values of the interest and the user's utterance contains OOV, then the recognition and understanding modules cannot work correctly. For example, the phenomenon of using acronyms is frequently observed in Korean language, but the acronym may be OOV for the system to cause errors in the recognition, understanding, and content searching. In this case, the system can recommend some contents related to the current dialog frame.

All of *No Slot* situations are not potential errors since an user utterance may have no slot information inherently like *"Yes"* or *"What time?"*. However, most of the utterances in task-oriented dialogs should contain some slot values to provide information for querying the database. Thus, if the number of extracted slots is zero, we can determine that the utterance is erroneous only when the number of retrieved examples is zero. If the error is detected, the system triggers the error recovery strategies to re-phrase the user utterance at the current situation. If the number of retrieved examples is over zero, it may need different strategy.

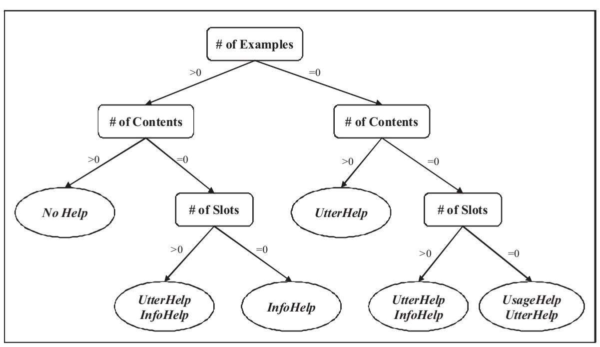

Fig. 3: Decision rule tree for triggering error recovery strategies.

So far, we explained three different situations for detecting errors at the level of dialog manager. To handle these situations, we define domain-independent decision rules as shown in Figure 3. For example, if *No Example*, *No Slot*, and *No Content* occur at the same time, we assume that the user did not experience at this system. Thus, the system should first say the functionality of this system and the extent of information for using this system. Furthermore, only *No Example* is occurred when the current utterance contains some recognition or understanding errors. To recover this situation, the system gives an utterance template of what the user could say at this situation. Thus, the users can say only an utterance fitted to the models of each module in the spoken dialog system since this template is pre-trained into the recognition, understanding, and dialog management.

4.2. Error Recovery Strategy

In our system, the error recovery strategies can be defined as four help types based on the type of errors addressed in Section 4.1. When no error is detected, *NoHelp* is triggered by an error handler which takes the responsibility for managing dialogs to handle errors. In this case, the system can successfully find the similar example with a high utterance similarity. Then, the system actions are correctly predicted and the system utterances are generated by a template-based natural language generation.

For the situation of *No Example*, we define an *UtterHelp* error recovery strategy. The system gives an example of what the user could say at this point in the dialog. Since the user utterance of the example database has an utterance identifier (UID) and a dialog identifier (DID), the system can search possible user utterance using the UID and DID of previous utterance in a discourse history. With its semantic keys (i.e. dialog act, main goal, and discourse history vector) of possible user utterance, the system tries to find the most appropriate template at the current situation using the utterance similarity between the erroneous utterance and the example template as following examples:

> *User: Please inform me a category of the restaurants that serves Korean food.*
> *ASR output: Please me a car of a restaurant that Korean food*
> *SLU output: [REQUEST, GUIDE-LOC, FOOD-TYPE=Korean foods]*
> ***[Error Detection: No Example]***
> *System: You can say "Please give me a category of the restaurants that serve [FOOD-TYPE]." to search restaurants of [FOOD-TYPE].*

In this example, some errors have occurred in the recognition and understanding module. Consequently, the system cannot search for a similar example over the threshold of the utterance similarity, which causes a *No Example* error. Then, the system tries to search possible templates and prompt an utterance template to achieve user's goal at the current situation. This template is well-trained to recognize and understand with a low error rate.

When the *InfoHelp* is triggered at *No Content* situation, the system recommends some candidates of the contents which can be retrieved at this situation. First, the system tries to search the information using the dialog frame by querying the knowledge database with each slot value. Some of them can be successfully matched. However, when the understanding error occurs, the system may not find any information using a certain slot, for example, when the particular slot value of *Kyoto* is extracted for the slot name of *Nation*. The target slot name is selected by its pre-defined priority and the number of contents corresponding to the slot name (i.e. *Nation* is a target slot name to inform content values by the *InfoHelp*.). To select alternative contents to recommend, we try to use a global sequence alignment with the confusion matrix of the phonemes [11]. Using the syllable- and phone-level alignments, we can select lexically and phonetically similar content values and also solve the acronym problem of Korean . In the following example, there is no content related to the *FOOD-TYPE* of *Korean fish* which is recognized incorrectly. After detecting *No Content*, the system selects the item of *FOOD-TYPE* as the target slot name and then prompts possible content values similar to the *Korean fish* using the global alignment.

> *User: Please inform a category of the restaurants that serves Korean fish.*
> *ASR output: Please inform me a category of restaurant that Korean fish*
> *SLU output: [REQUEST, SEARCH-LOC, FOOD-TYPE=Korean fish]*
> ***[Error Detection: No Content]***
> *System: There is no content on "Korean fish". You can say "Korean food", "Korean beef", and "Japanese fish" as the slot of [FOOD-TYPE].*

Additionally, the system provides an *UsageHelp*, a longer help message, which includes possible functions and usages of the system, as well as an explanation of the current situation. This *UsageHelp* strategy is important for the novice users who do not know anything about the system. The system gives a different level of system guidance for the novice users according to the status of discourse history vector. For example, if no slot is filled, then the system says the overall usage for the system like the following example:

> *User: When can I eat foods of Korean.*
> *ASR output: When it fish Korean*
> *SLU output: [WH-QUESSTION, SEARCH-PHONE]*
> ***[Error Detection: No Example, No Content, No Slot]***
> *System: This is a car navigation system to select your destination in Pohang city. You can say about the address, name, and type of the location. For example, "Let me go to [LOC-NAME]."*

In this example, the system assumes that the user has not experienced at this system since all of error situations are detected. Then, the *UsageHelp* is triggered by the decision rule tree.

5. EXPERIMENTS AND RESULTS

We have developed a spoken dialog system for a car navigation using the EBDM framework. This is a spoken dialog system that provides support for the information and selection of the desired destination. We have first collected about 160 place names in Pohang city using the web and tour guide books. The information contents include its address, location type (i.e. restaurant, garage, gas station, hospital, etc.) and phone number. Using the role-playing, about 10 people collected the human-human dialog corpus including about 530 utterances based on a set of pre-defined scenarios related to its task. Then, ten novice users and five expert users of spoken dialog system were asked to participate in a preliminary system evaluation. System developers belonged to the expert users. To evaluate our approach for handling errors, we asked the users with 5 scenarios consisting of different tasks about the car navigation domain. We did not instruct the novice users on the system and task guidance. The system was tested with 75 dialogs (5 dialogs for each user) using the text and real speech input. First, we evaluated the Task Completion Rate (TCR) without error recovery strategies as an user evaluation of both text (WER=0.0%) and speech input (WER≈20.0%) according to the user type. The results of the experiments are shown in Table 1. We observed that the TCR of the expert users is higher than that of the novice users because the experts know what to say and how to use the system. The average number of turns ($\#AvgTurn$) of the expert is also shorter, which shows 5.48 turns per a dialog using the speech input. However, the $\#AvgTurn$ of the novice users is 5.68 turns per a dialog and the errors is more frequently detected as the average number of error detection ($\#DetErr$) is 1.56 errors per a dialog using the speech input.

Input Type	TCR (%)	$\#AvgTurn$	$\#DetErr$
Text	96	4.28	0.36
Speech	80	5.48	1.32

(a) Expert User

Input Type	TCR (%)	$\#AvgTurn$	$\#DetErr$
Text	84	5.16	1.44
Speech	76	5.68	1.56

(b) Novice User

Table 1: User evaluation according to the user type and the input type.

To verify the feasibility of our error recovery strategies for the novice users, we also measured TCR after integrating the example-based error recovery method on the same environment as shown in Table 2. Our approach shows more effectiveness to the novice users. The value of TCR for the novice users is increased from 76% to 84% using the speech input although the errors are highly detected. However, the value of $\#AvgTurn$ is also slightly increased as the turns of the error recovery strategies were added.

The number and success rate of each error recovery strategy is shown in Table 3. The success rate of error recovery was evaluated by whether or not the system utterances are helpful for the users to re-phrase with the guidance of sys-

System	Input	TCR (%)	$\#AvgTurn$	$\#DetErr$
Recovery (-)	Text	84	5.16	1.44
	Speech	76	5.68	1.56
Recovery (+)	Text	92	5.88	2.12
	Speech	84	6.28	2.12

Table 2: User evaluation of the novice users for error recovery strategies.

tem's helps. The *UtterHelp* was highly successful to speak a proper utterance to proceed current dialog by the dialog manager.

Recovery Type	Number		Success Rate (%)	
	Text	Speech	Text	Speech
UtterHelp	25	16	92.00	87.50
InfoHelp	16	26	75.00	61.54
UsageHelp	12	11	58.33	54.55

Table 3: The number and success rate of each error recovery strategy.

6. CONCLUSION AND FUTURE WORKS

In this paper, we propose an example-based error recovery strategies for the EBDM framework. We defined the error situations at the stage of dialog manager to detect and classify errors without the confidence scores. Then, we used some domain-independent decision rules to invoke the proper error recovery strategy for providing the user template and contents based on discourse history. Users can re-prompt well-recognizable and well-understandable utterances using the template and contents. We believe that this approach is effective to the novices who do not know anything about the dialog system with the text or speech input in real world. In addition, this approach overcomes the limitation of the original EBDM framework for no example situation. There are some issues for future research of our example-based error recovery. One of them is how to incorporate the error recovery strategy into the user modeling to classify the skill level of the users. To reject the erroneous utterances which are semantically and grammatically correct, the error detection should be also improved by combining with reliable utterance verification method. Finally, we should evaluate the learning effect for the novices to operate the spoken dialog systems using our approach across several dialog systems.

7. REFERENCES

[1] C.A. Chapelle, *Computer applications in second language acquisition*, Cambridge University Press Cambridge, 2001.

[2] M. McTear, I. O'Neil, P. Hanna, and X. Liu, "Handling errors and determining confirmation strategies-an object-based approach," *Speech Communication*, vol. 45, pp. 249–269, 2005.

[3] G. Skantze, "Exploring human error handling strategies: implications for spoken dialogue systems," in *Proc. of ISCA Workshop on Error Handling in Spoken Dialogue Systems*, 2003, pp. 71–76.

[4] W.K. Lo and F.K. Soong, "Generalized posterior probability for minimum error verification of recognized sentences," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2005, pp. 85–88.

[5] T.J. Hazen, S. Seneff, and J. Polifroni, "Recognition confidence scoring and its use in speech understanding system," *Computer Speech and Language*, vol. 16, pp. 49–67, 2002.

[6] J.D. Williams and S. Young, "Partially observable markov decision processes for spoken dialog systems," *Computer Speech and Language*, vol. 21, pp. 393–422, 2007.

[7] J. Shin, S. Narayanan, L. Gerber, A. Kazemzadeh, and D. Byrd, "Analysis of User Behavior under Error Conditions in Spoken Dialogs," in *Proceedings of the International Conference on Spoken Language Processing*, 2002.

[8] C.J. Lee, S.K. Jung, J.H. Eun, M.W. Jeong, and G.G. Lee, "A situation-based dialog management using dialog examples," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2006, pp. 69–72.

[9] M. Nagao, "A frame work of a mechanical translation between japanese and english by analogy principle," in *Proceedings of the international NATO symposium on artificial and human intelligence*, 1984, pp. 173–180.

[10] C.J. Lee, S.K. Jung, M.W. Jeong, and G.G. Lee, "Chat and goal-oriented dialog together: a unified example-based architecture for multi-domain dialog management," in *Proceedings of the IEEE Spoken Language Technology Workshop*, 2006, pp. 194–197.

[11] S.B. Needleman and C.D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of Molecular Biology*, vol. 48, pp. 443–453, 1970.