A SITUATION-BASED DIALOGUE MANAGEMENT USING DIALOGUE EXAMPLES

Cheongjae Lee, Sangkeun Jung, Jihyun Eun, Minwoo Jeong, and Gary Geunbae Lee

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

ABSTRACT

In this paper, we present POSTECH Situation-Based Dialogue Manager (POSSDM) for a spoken dialogue system using both example- and rule-based dialogue management techniques for effective generation of appropriate system responses. A spoken dialogue system should generate cooperative responses to smoothly control dialogue flow with the users. We introduce a new dialogue management technique incorporating dialogue examples and situation-based rules for the Electronic Program Guide (EPG) domain. For the system response generation, we automatically construct and index a dialogue example database from the dialogue corpus, and the proper system response is determined by retrieving the best dialogue example for the current dialogue situation, which includes a current user utterance, dialogue act, semantic frame and discourse history. When the dialogue corpus is not enough to cover the domain, we also apply manually constructed situation-based rules mainly for meta-level dialogue management. Experiments show that our example-based dialogue modeling is very useful and effective in domain-oriented dialogue processing.

1. INTRODUCTION

In the past few years, spoken dialogue systems have been used in various applications for natural and convenient interface with users [1], and recently, the interest in spoken dialogue systems has been sharply increasing. In the dialogue system, the dialogue manager is the central component. The dialogue manager accepts spoken input from the user, produces system responses to be communicated to the user, interacts with external knowledge sources, and generally controls the dialogue flow.

Most of the previous dialogue management systems have been developed with the finite state-based approach [2]. In this approach, a system response is determined by the fixed state transition in advance. It is usually used for the rapid prototyping of dialogue system for strong-typed interactions. However, the problem is that it is not flexible enough to handle various natural language dialogue phenomena, because the next state of the dialogue is strictly determined by the fixed state-transition network in this model. The domain portability is also poor because the whole finite state model should be redesigned for a new domain. Recently, some researchers proposed the frame-based approach which is suitable for form-filling tasks in which the system asks the user a series of questions to gather information and then consults the external knowledge source [3]. To determine the system's next actions based on the contents of the frame, human-computer dialogues are controlled by manually designed rules. Although this approach permits more flexible dialogues, it has problems similar to the finite state-based approach such as enormous human effort and low domain portability. Plan-based modeling can handle greater complex tasks than the finite state-based and frame-based dialogue modeling [4]. Although it attempts to model the goals of the user and the computer by using a dialogue planning scheme, it has not been developed as a commercial system because it attempts to control too flexible and complex tasks, resulting in low performance. This approach again, so far, is not free from the same enormous manual effort and low domain portability problem since it relies to the plan recipe-based rules.

There have also been some studies to improve domain portability, and O'Neil et. al. proposed an object-oriented dialogue system by modularization in [5]. However, it also requires much human effort to encode many handcrafted rules.

In this paper, we suggest a situation-based dialogue management technique using dialogue examples in order to overcome these restrictions. The basic idea underlying the situation-based dialogue manager is based on a frame-based model but the dialogue flows are dynamically managed by the current situation in order to manage more natural human-computer dialogues. It is also an object-oriented architecture that can easily build a multi-domain dialogue system. To eliminate the human effort for dialogue model building, we have developed an example-based technique to automatically construct and index an effective domain-dialogue model.

2. POSSDM: POSTECH SITUATION-BASED DIALOGUE MANAGER

The purpose of our dialogue management is to be practical and flexible enough to control a natural human-computer dialogue, and to provide domain portability in order to allow uses in various applications. Inspired by the work of O'Neil et. al. [5], and motivated to overcome the conventional dialogue systems' weaknesses, we developed a situation-based spoken dialogue management using the following two dialogue modeling principles:

- Dialog management should be state-transition free and based on the current situation for system response generation.
- Domain-dependent dialog management should be based on a specific domain expert for more efficient management.

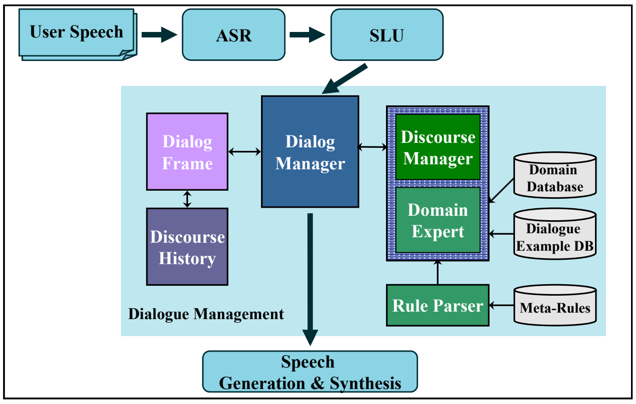

Figure 1: Overview of POSSDM System Architecture

An overview of POSSDM is shown in Fig. 1, which is similar to the one in [5]. Each domain has its own domain expert and language generation module. As most of these domain-specific dialogue models are learned automatically from the dialogue examples, we can easily develop systems for new domains. The dialogue manager achieves the task completion goal of a specific domain task through a series of interactions with the users by using the results of the SLU (Spoken Language Understanding) dialogue frames. The discourse history is a set of dialogue frames in one dialogue session. The dialogue manager produces domain-independent system concepts in order to generate domain-specific system responses using the discourse manager and the domain expert based on the current situation of the dialogue. The system concepts are converted to domain-specific system responses in the language generation module of each domain.

To determine the system responses, we consider the current situation of the dialogue instead of relying on a finite state transition network. The "situation" used in our system means the current dialogue states including a current user utterance, user intention, semantic frame, and discourse history. The situation-based dialogue management leads the dialogue using the rules which reflect the current situation of the dialogue.

3. EXAMPLE-BASED DIALOGUE MODELING

3.1. Rule-based dialogue modeling

In rule-based dialogue modeling, the dialogue system has to construct well-designed domain rules. Our system also uses three kinds of situation-based rules to generate system responses under the current situation. But most dialogues are processed using dialogue examples in our system as described in section 3.2, so we only need some meta-level rules to cover the general situation.

- **Situation-action rules**: rules for describing the system's actions under the current situation.
- **Constraint-relax rules**: rules for relaxing some constraints on database queries.
- **Frame-reset rules**: rules for restarting a new dialogue frame for the case of domain switching and dialogue closing.

When the current situation satisfies the situation-action rules, the dialogue system generates system responses by accessing the external knowledge source. For example, EPG system needs to inform the user with the television schedule database. If the program fails to be retrieved for the user, the system should give alternative programs that the user can select by the constraint-relax rules. Frame-reset rules determine whether the current utterance is independent of previous dialogues for new dialogue restarting or domain switching

For effective situation-based dialogue management, we need to construct enough rules manually for domain specific dialogue models, which is often time consuming. In this paper, we propose an example-based dialogue modeling technique to avoid theses limitations.

3.2. Example-based dialogue modeling

The example-based technique has the advantage of being more effective and domain portable because it is able to automatically generate system responses from dialogue examples.

3.2.1. Indexing and Querying

For the dialogue models, we should automatically make a dialogue example database from dialogue corpus. To minimize corpus annotation, we construct the database with the previously tagged corpus for training the SLU model. Because the SLU corpus is re-used, we only annotate the discourse history vector. The discourse history represents a binary vector of frame slot-filling up to the current dialogue. The keys for indexing the dialogue examples are also used as the query keys to search for the matched examples from the example database. An dialogue example in EPG domain is shown in the Fig 2.

Utterance	그럼 SBS 드라마는 언제 하지? (Geu-leom SBS deu-la-ma-neun eon-je ha-ji?) Then, when do the SBS dramas start?
Dialog Act	Wh-question
Main Action	Search_start_time
Component Slots	[channel = SBS, genre = 드라마]
Discourse History	[1,0,1,0,0,0,0,0,0]

Figure 2: Example of Tagging the Dialogue Corpus

The constraints on the indexing and search are extracted from the current dialogue situation such as user intention (dialogue act plus main action), semantic frame (component slots), discourse history, and lexical string of the utterance. However, in some cases, we need to relax the constraints to do a partial match. The relaxed constraints only involve dialogue act and main action because system responses mainly depend on the user intention of the current utterance.

3.2.2. Utterance Similarity

When the retrieved dialogue examples are not unique, we choose the best one using the utterance similarity computation. The utterance similarity values include the lexico-semantic similarity and the discourse history similarity. The lexico-semantic similarity is defined as an edit distance between utterances of users and the examples. In our research, component slots are assigned to predefined slot names in the domain specific SLU [8]. To measure the lexico-semantic similarity, the slot values are replaced by its slot names (Fig 3). The degree of the discourse history similarity is a cosine measure between the binary vectors that are assigned the value 1 if the slot is filled, and 0 otherwise.

User Utterance	그럼 SBS 드라마는 언제 하지? (Geu-leom SBS deu-la-ma-neun eon-je ha-ji?) Then, when do the SBS dramas start?
Component Slots	[channel = SBS, genre = 드라마]
Lexico-Semantic Input	그럼 [channel] [genre] 는 언제 하 지

Figure 3: Example of Lexico-Semantic Representation

Although the dialogue examples are able to generate appropriate responses for most dialogue situations, some situations require the meta-rules for leading the dialogue. For example, if the retrieved dialogue example result is absent, the system should give an alternative suggestion. To deal with these special situations, some manually designed situation-based meta-rules were used together. Fig 4 illustrates an overall strategy of the example-based dialogue modeling.

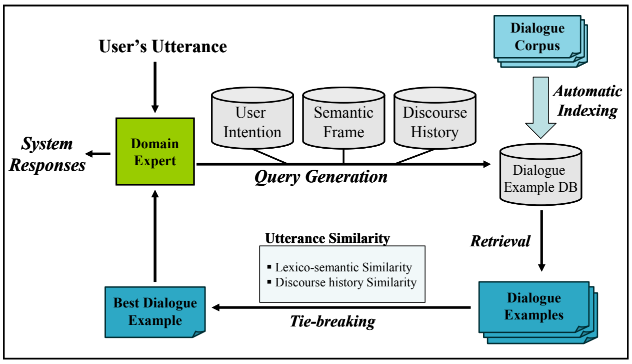

Figure 4: Example-based Dialogue Modeling Strategy

4. EXPERIMENTAL RESULTS

4.1. Dialogue corpus

We constructed the dialogue example database from the EPG dialogue corpus. This corpus consists of 380 user utterances of 88 Korean TV-guide dialogues. Each utterance is annotated for training the SLU model, e.g. dialogue act, main action, and semantic slot-value pair. We also annotated the discourse history vector manually for selecting dialogue examples.

4.2. Speech recognizer and SLU

Our speech recognizer was developed based on HTK (Hidden Markov Model Toolkit). The recognizer used a pre-trained dialogue acoustic model and adopted the EPG domain specific language model. The performance was word error rate (WER) 15.3% in this domain. The SLU module of POSSDM was constructed by a concept spotting approach which aims to extract only the essential information for predefined meaning representation slots [6]. These slots include dialogue act, main action, and component slots for EPG domain (e.g. channel, program, genre, time, etc). The F-measure of the SLU in this domain is shown in Table 1.

Slot	Textual Input (WER 0.0%)	Spoken Input (WER 15.3%)
Dialogue Act	95.33	85.34
Main Action	93.50	81.78
Component Slot	90.85	80.12

Table 1: The performance of SLU

4.3. Experiments on dialogue modeling

Our dialogue system can be evaluated both from the text inputs and the speech inputs. Firstly, we calculated the example-matching rate (EMR) and success turn rate (STR) to evaluate the example-based dialogue modeling for the text inputs. EMR designates the average success rate of the example match for the user utterance input for each case of exact match and partial match, and STR designates the average success rate of the response correctness. We asked the 5 test volunteers to use our dialogue system with 10 random text inputs in the EPG domain, and averaged their results. Table 2 shows that the exact matching examples are more successful than the partial matching, and that most dialogues can be covered by our example-based model including the partial match. This means that the proposed model guarantees that the dialogue manager smoothly controls the dialogue flow, because the dialogue examples which were collected by the dialogue corpus would successfully cover the dialogue in the EPG domain.

Example Match Type	EMR	STR
Exact Match	0.42	0.90
Partial Match	0.52	0.73
No Example	0.06	0.33

Table 2: The example-matching rate (EMR) and the success turn rate (STR)

Secondly, we also measured the user satisfaction to verify the practical usability of POSSDM for text and speech input. We asked each 5 test volunteers to assign 5 different EPG tasks. The volunteers evaluated every system's response in each dialogue turn. We evaluated the performance of our dialogue system based on [7]. For including the evaluation of our example-based modeling, the user satisfaction was defined with the linear interpolation of three different measures: user perception of Task Completion Rate (TCR), Mean Recognition Accuracy (MRA), and STR instead of the elapsed time. Each was weighted by a factor of 1/3, so that the maximum value of the user satisfaction is one.

Evaluation	Textual Input	Spoken Input
TCR	0.92	0.76
STR	0.88	0.65
MRA	1.00	0.85
User Satisfaction	0.93	0.75

TCR : User Perception of Task Completion Rate
STR : Success Turn Rate
MRS : Mean Recognition Accuracy
User Satisfaction = αTCR + βSTR + γMRA

Table 3: The dialogue performance of POSSDM

As we can see in Table 3, our system's user satisfaction was 0.93 for textual input and 0.75 for spoken input, which means our dialogue system is very useful in the EPG domain. It also shows our example-based dialogue modeling is feasible and effective.

5. CONCLUSION AND DISCUSSION

This paper has proposed the situation-based dialogue management and the example-based technique for dialogue modeling. By automatically constructing the example-based dialogue model from the dialogue corpus, we guarantee to develop an effective and practical spoken dialogue system. The experimental results using 88 dialogues from the EPG domain have shown the feasibility of our techniques in POSSDM. In this paper, we have shown the single-domain dialogue system for the EPG task. However, we could easily develop the dialogue system for other domains and tasks based on our methods.

6. ACKNOWLEDGEMENT

This research was supported as a Brain Neuroinformatics Research Program sponsored by the Ministry of Commerce, Industry and Energy of Korea.

7. 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, pp 27-37, 2001.

[2] McTear, "Modelling Spoken Dialouges with State Transition Diagrams : Experiences with the CSLU Toolkit," in *Proceedings of the 5th International Conference on Spoken Language Processing*, vol. 4, pp. 1223-1226, 1998.

[3] O. Lemon, A. Gruenstein, A. Battle, and S. Peters, "Multi-tasking and Collaborative Activities in Dialogue Systems," in *Proceedings of the 3rd SIGDIAL Workshop on Discourse and Dialogue*, pp. 113-124, 2002.

[4] J. Chu-Carroll, "MIMIC: An Adaptive Mixed Initiative Spoken Dialogue System for Information Queries," in *Proceedings of the 6th Applied Natural Language Processing Conference*, pp. 97-104, 2000.

[5] I. O'Neil, P. Hanna, X. Liu, D. Greer, and M. McTear, "Implementing advanced spoken dialogue management in Java," *Speech Communication*, vol. 54, no. 1, pp 99-124, January 2005.

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

[7] D. J. Litman and S. Pan, "Empirically evaluating an adaptable spoken dialogue system," in *Proc*eedings of the 7th International Conference on User Modeling, pp. 55-64, 1999.