CHAT AND GOAL-ORIENTED DIALOG TOGETHER: A UNIFIED EXAMPLE-BASED ARCHITECTURE FOR MULTI-DOMAIN DIALOG MANAGEMENT

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

Pohang University of Science and Technology

Departement of Computer Science and Engineering

Pohang, South Korea

{lcj80, hugman, stardust, gblee}@postech.ac.kr

ABSTRACT

This paper discusses development of a multi-domain conversational dialog system for simultaneously managing chats and goal-oriented dialogs. In this paper, we present a UMDM (Unified Multi-domain Dialog Manager) using a novel example-based dialog management technique. We have developed an effective utterance classifier with linguistic, semantic, and keyword features for domain switching and an example-based dialog modeling technique for domain-portable dialog models. Our experiments show that our approach is very useful and effective in multi-domain dialog system.

Index Terms— dialog management, multi-domain dialog system, domain spotter, example-based dialog modeling

1. INTRODUCTION

Recent dialog systems have been applied for use in various applications for natural and conversational interfaces in mobile and ubiquitous environments. Such dialog systems are now able to provide services for telematics, home networking, or intelligent robots. These systems gradually have been capable of supporting multiple domains and accessing information from a broad variety of sources and services to operate on not only one but across several restricted domains.

There are some difficulties in building a dialog management system for the processing of dynamic multi-domain dialogs. One of them is to identify user's domain of interest correctly and to switch between domains smoothly without requiring the high cost of building the dialog model for each different application domains. In addition, traditional dialog systems are mainly designed to provide information-seeking services to a relatively structured and back-end database (such as travel reservation information), where users have well-defined and task-oriented goals. However, there is another genre of dialog domain such as chats and small talks where users do not have any specific goal but want to have social interactions with the dialog systems. These chat and small talk domains should be smoothly integrated into the goal-oriented dialog domains for more human-like dialog management. In this perspective, we designed and implemented a UMDM (Unified Multi-domain Dialog Manager) where we can handle different genre of dialog domains such as chat and goal-oriented dialogs together in a uniform manner. The unified architecture for multi-domain dialog system is capable of managing multi-domain and multi-genre dialogs using a domain portable and practical multi-domain dialog modeling technique.

THIS RESEARCH WAS SUPPORTED BY THE MIC, KOREA, UNDER THE ITRC SUPPORT PROGRAM SUPERVISED BY THE IITA (IITA-2005-C1090-0501-0018), AND ALSO SUPPORTED BY THE BK 21 PROGRAM OF THE MINISTRY OF EDUCATION, KOREA.

2. RELATED WORK

Most of the previous dialog systems have been focused on solving goal-oriented dialogs in the domain-specific dialog management. However, these systems are not capable of managing chat-style natural dialogs such as our daily social conversation. If a user speaks task-independent utterances without any specific goal, the dialog system can not handle them flexibly. Some works have been directed towards predicting an out-of-domain utterances and filtering out them via dialog manager [1]. However, this approach does not directly improve the flexibility of the dialogs. In our approach, we propose a unified architecture in which chats and goal-oriented dialogs are managed together with the same dialog modeling technique for a natural human-computer interface. There have also been some efforts to study the issue of domain identification and switching for multi-domain dialog systems [2]. In these approaches, the domain of utterance is classified by either linguistic and syntactic features (e.g. POS tag, n-gram) or high-level linguistic features (e.g. dictionary, ontology). In our approach, we propose a feature-based classification using several sets of linguistic, semantic, and keyword features.

For goal-oriented dialog system, in our previous work, we have presented an example-based dialog modeling approach for flexible domain-portable dialog models [3]. In this paper, we augment our previous dialog management architecture for managing chats and goal-oriented dialogs for multi-domain application.

3. SYSTEM ARCHITECTURE

The goal of our dialog management is to be practical and flexible enough to control a natural human-computer conversation, and to improve domain portability in order to allow uses in various diverse applications. Compared to traditional dialog systems, the UMDM is a novel approach with improved flexibility and naturalness of dialog flows in human-computer interaction. We propose a new technique for unifying the dialog system of chat and various goal-oriented dialogs. An overall architecture of the proposed system is illustrated in Figure 1. This unified dialog manager utilizes a spotting module for determining the agent and the domain of the current user utterance. The generic/domain-specific NLU (Natural Language Understanding) module and the agent/domain spotter module will be described in detail in the following section.

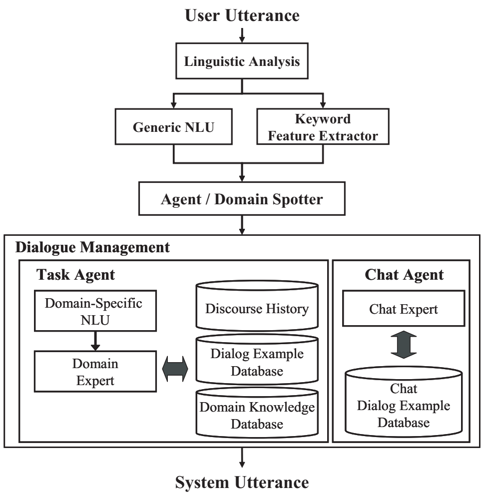

Fig. 1. An overall architecture of UMDM

4. TWO-LEVEL NATURAL LANGUAGE UNDERSTANDING

In the UMDM, the NLU module was constructed by a concept spotting approach which aims to extract only the essential information for predefined meaning representation slots as previously described in [4]. The predefined slots include dialog act, main action, and component slots for a domain-specific task (Fig. 2). To manage both chats and goal-oriented dialogs, our dialog system has two different levels of the language understanding components: generic NLU and domain-specific NLU. We define high-level semantic representation for generic dialog act and main action. We can annotate generic dialog acts of our dialog corpus based on DAMSL (Dialog Act Mark-up in Several Layers) schema [5]. However, we need to define some tags for labeling main actions for high-level domain-independent NLU based on semantic roles of the utterance.[1] The domain-independent NLU result is used for semantic features in detecting the current agent and the domain, and for indexing the dialog example database of the chat corpus. The semantic features are helpful to determine the current agent and the domain because the semantic tags are usually used in domain-limited. On the other hand, a low-level domain-specific NLU is used for managing the domain-specific goal-oriented dialogs in the task agent. The domain-specific NLU model is necessary to improve the performance of the NLU because the result is augmented with the generic NLU which is more difficult. The slot values of the low-level main action are assigned from one of the classes which classify the main application actions in a specific domain.

Dialog Act			
WH-Question	Request	Yes-No-Question	Say
Say_noun	Say_prop		
Main Action			
Search_program	Search_channel	Search_day	Search_starttime
Search_endtime	Search_currenttime	Serach_currentdate	Move_channel
Alarm	Record	TV_on	TV_off
Component Slot			
Genre	Channel	Date	Start_time
End_time	Cast	Day	Program

Fig. 2. The list of the predefined slots in the semantic frame in EPG (Electronic Program Guide) domain

5. AGENT AND DOMAIN SPOTTER

One of the key-issue in our system is to support a multi-domain goal-oriented dialogs as well as chats. In this section, we present our spotting module employed for determining the agent and the domain at current user's turn.

5.1. Agent Spotter

We divide user utterances into chat and goal-oriented dialog at each turn. In most cases, user utterances have domain-specific intention to access a back-end domain knowledge database. For this reason, the goal-oriented dialogs should involve more complex processing than a chat using a domain-specific expert. Our agent spotter decides whether a chat or a task agent will be used in further dialog processing. To determine the type of agent, we investigate various feature sets that can be used in the stochastic classifier to improve the performance of the agent spotter. We used the linguistic features (e.g. word, POS tag, n-gram) and semantic features (e.g. generic dialog act and main action) together.

5.2. Domain Spotter

In our system, we consider the combined method of using keyword-based and feature-based approaches for automatically classifying the domains. First, we apply the TF*IDF algorithms, which is commonly used in information retrieval for weighting content words. For utterance classification, this is very efficient in detecting the domains, but the performance of TF*IDF scheme is not high enough for smooth domain switching. In addition, this method makes no sense when used alone because some utterances contain no keyword at all. To improve the performance of the domain spotter, we again use the linguistic and semantic features similar to the agent spotter. In addition, we extract the keyword features within a user utterance including the N-best keyword and the N-best domain class predicted from the TF*IDF weighting scheme. The feature sets are all summarized in Table 1.

6. EXAMPLE-BASED DIALOG MODELING

To determine the system response, we consider the current situation of the dialog instead of relying on a finite state transition network. The situation-based dialog management leads the dialog using the rules which reflect the current situation of the dialogs. How-

[1] We define 15 dialog acts and 21 main actions for understanding domain-independent user utterances. For instance, there are dialog acts such as *statement, what-question, yes-no-question, etc*, and the main actions such as *introduce, greet, desire, search, recommend, abandon, etc*.

Feature Set	Description
Linguistic Feature	word, POS tag, N-gram
Semantic Feature	dialog act, main action
Keyword Feature	N-best keyword, N-best domain class (from TF*IDF weight)

Table 1. Features used in agent and domain spotter

ever, the rules for multi-domain dialog system need intensive human effort to redesign for the new domain. To reduce manual design of the domain-specific rules, we propose an example-based dialog modeling for automatically generating domain-specific rules of dialog management. The example-based dialog modeling has the advantage of being more effective and domain portable because it is able to automatically generate system responses from dialog example database.

6.1. Indexing and Querying

For the dialog models, we should automatically make a dialog example database from dialog corpus. To minimize corpus annotation, we construct the database with the previously tagged corpus for training the NLU model. Because the NLU corpus is re-used, we only need to annotate the discourse history vector [3]. The discourse history represents a binary vector of the frame slot-filling up to the current dialog. The keys for indexing the dialog examples are also used as the query keys to search for the matched examples from the example database. A dialog example in an EPG domain is shown in Figure 3.

Utterance	그럼 **KBS** 드라마는 언제 하지? Then, when do the **KBS dramas** start?
Dialog Act	When-question
Main Action	Search_start_time
Component Slots	[channel = KBS, genre = 드라마]
Discourse History	[1,0,1,0,0,0,0,0,0]

Fig. 3. An example of the tagging of the dialog corpus

The constraints on the indexing and search are extracted from the current dialog situation such as domain, user intention (dialog 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 example match. The relaxed constraints only involve dialog act and main action because system responses mainly depend on the user intention of the current utterance.

6.2. Utterance Similarity

When the retrieved dialog 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 a normalized edit distance between utterances of user and the examples. In normalized edit distance, we define a cost function C as the following equation:

$$C(i,j) = \begin{cases} 0 & \text{if } w_{1,i} = w_{2,j} \text{ and } t_{1,i} = t_{2,j} \\ 1 & \text{if either } w_{1,i} = w_{2,j} \text{ or } t_{1,i} = t_{2,j} \\ 2 & \text{if } w_{1,i} \neq w_{2,j} \text{ and } t_{1,i} \neq t_{2,j} \end{cases}$$

$w_{1,i}$: the base form of morpheme of user's utterance at position i
$w_{2,j}$: the base form of morpheme of example's utterance at position j
$t_{1,i}$: the POS tag of user's utterance at position i
$t_{2,j}$: the POS tag of example's utterance at position j

The degree of the discourse history similarity is a cosine measure between the binary vector that are assigned the value 1 if the component slot is already filled, and 0 otherwise. We use this measure to determine the similarity between the discourse history of the user and the example utterances with respect to the current dialog situation. Given two similarity measures, the utterance similarity can be expanded using a linear interpolation with the same weight of 0.5. Figure 4 illustrates an overall strategy of the example-based dialog modeling.

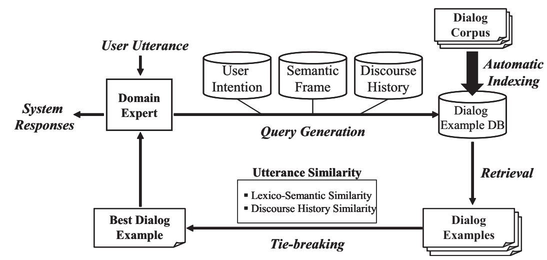

Fig. 4. A strategy of the example-based dialog modeling

7. EXPERIMENTS AND RESULTS

7.1. Dialog Corpus

We constructed a dialog corpus of chats and goal-oriented dialogs mixed together. The chat corpus contains 2377 user utterances in 10 domains. To collect the chat corpus, we used a commercial chatting agent in Korea. On the other hand, the domain of goal-oriented dialogs is one of the 2 domain tasks for car navigation and EPG domain in mobile devices. Annotators collected the dialog data including 513 user utterances based on a set of pre-defined scenarios, relating to the car navigation and the EPG domain.

7.2. Spotting Evaluation

The performance of the proposed spotting technique was evaluated for combination of feature sets using ME(Maximum Entropy) classifier. First, the agent spotter in our system is evaluated with the ME classifier using linguistic and semantic features. Typically, linguistic features including word, POS tag, and n-gram feature are applied for the utterance classification as a baseline. We measured the accuracy of the agent spotter incorporating our semantic features. Table 2 shows the performance of the agent spotter in our system. When only linguistic features were considered, the accuracy was close to 96.69%. However, combining semantic features including generic semantic tags yields an accuracy of 98.09%, which is a significant increase. From this result, we conclude that the agent spotter depends on generic dialog act and main action.

Feature Set		Accuracy (%)
Baseline (only Linguistic Feature)		96.69
Semantic Feature	+ Dialog act	97.39
	+ Main action	98.09

Table 2. The accuracy of the agent spotter

Similarly, the performance of the domain spotter is evaluated by the feature sets with the ME classifier. For the baseline performance, we evaluated only using the TF*IDF weighting alone. Then, 2-best hypotheses of the keywords and the domain classes by calculating the TF*IDF weights were applied to investigate the effect of the keyword features. The performance of the domain spotter is shown in Table 3. Adding the keyword features yields an accuracy of 86.18% which is a significant improvement. These results show that our spotter is valid in detecting both the agent and the domain for smooth domain switching in our system.

Feature Set		Accuracy (%)
Baseline (TF*IDF)		72.88
Linguistic Feature		77.47
Sematic Feature		77.92
Keyword Feature	+ 2-best keyword	78.87
	+ 2-best domain class	86.18

Table 3. The accuracy of the domain spotter

7.3. Dialog Modeling Evaluation

We calculated the coverage of the dialog examples using the example-matching rate (EMR) and the success turn rate (STR) for evaluation of the example-based dialog modeling. EMR designates the average success rate of the example match for user utterance input for each case of the exact match and the partial match, and STR designates the average success turn rate of the response correctness. We asked the 4 test volunteers with the pre-defined scenarios to evaluate our system using 422 user utterances which are independent of our previous 2890 utterances used in building the dialog example database. Table 4 shows that the exact matching examples are more successful than the partial matching. If no exact matching example is retrieved, a partial match is performed under relaxed constraints which only involve dialog act and main action. Although there are some failures of the system response, most of the dialog situation can be covered by our example-based model including the partial match (EMR 96%). This means that the proposed model guarantees that the dialog manager smoothly controls the dialog flow, because the dialog examples which were automatically collected by the dialog corpus would successfully cover the independent dialogs both in chatting and the domain-specific tasks.

Example Match Type	EMR	STR
Exact Match	0.60	0.69
Partial Match	0.36	0.52
No Example	0.04	0.06

Table 4. The example rate (EMR) and success turn rate(STR)

To verify the practical usability of our system, we also measured the user perception of the task completion rate and the success turn rate in the aspects of goal-oriented dialogs. The 4 test volunteers evaluated every system's response in each dialog turn and the user perception of the task completion. As we can see in Table 5, our system is very successful in multi-domain goal-oriented dialogues and our example-based dialog modeling can be applied to various applications in multi-domain information seeking dialogs.

Evaluation	Goal-oriented Dialog
Success Turn Rate	0.75
Task Completion Rate	0.81

Table 5. The goal-oriented dialog evaluation of UMDM

8. CONCLUSION

This paper has proposed a unified example-based dialog management architecture to manage multi-domain chats and goal-oriented dialogs together. We have also introduced the effective spotting techniques for agent and domain switching, and the example-based technique for domain portable dialog modeling. The experimental results have shown the feasibility of our techniques. Two main possible extensions can be possible in the future: one is to apply to other goal-oriented tasks (e.g. errand and travel reservation domains) and the other is to develop a dialog management workbench to help effective collect and annotate the dialog corpus.

9. REFERENCES

[1] I. Lane, T. Kawahara, T. Matsui, and S. Nakamura, "Topic classification and verification modeling for out-of-domain utterance detection," in *Proceedings of the ICSLP*, 2004, pp. 2197–2200.

[2] C. Chelba, M. Mahajan, and A. Acero, "Speech utterance classification," in *Proceedings of the ICASSP*, 2003, pp. 280–283.

[3] C. Lee, S. Jung, J. Eun, M. Jeong, and G.G. Lee, "A situation-based dialog management using dialog examples," in *Proceedings of the ICASSP*, 2006, vol. 1, pp. 69–72.

[4] J. Eun, M. Jeong, and G.G. Lee, "A multiple classifier-based concept-spotting approach for robust spoken language understanding," in *Proceedings of the EUROSPEECH*, 2005, pp. 3441–3444.

[5] M. Core and J. Allen., "Coding dialogs with the damsl annotation scheme," in *Working Notes of the AAAI Fall Symposium on Commnicative Action in Humans and Machines*, 1997, pp. 28–35.