Methods to Improve Its Portability of A Spoken Dialog System Both on Task Domains and Languages

Yunbiao Xu, Fengying Di*, Masahiro Araki**, Yasuhisa Niimi***

*yunbiao@163.net, difengying@hotmail.com
College of Computer Science and Information Engineering,
Hangzhou University of Commerce, Hangzhou, China
** araki@dj.kit.ac.jp, yasuhisa2133@ybb.ne.jp
Department of electronics & information science,
Kyoto Institute of Technology, Kyoto, Japan

Abstract

This paper presents the methods to improve its portability of a spoken dialog system both on task domains and languages, which have been implemented in Chinese and Japanese in the tasks of sightseeing, accommodation-seeking guidance. Such methods include case frame conversion, template-based text generation and topic frame driven dialog control scheme. The former two methods are for improving the portability across languages, and the last one is for improving the portability across domains. The case frame conversion is used for translating a source language case frame into a pivot language one. The template-based text generation is used for generating text responses in a particular language from abstract responses. The topic frame driven dialog control scheme makes it possible to manage mixed-initiative dialog based on a set of task-dependent topic frames. The experiments showed that the proposed methods could be used to improve the portability of a dialog system across domains and languages.

1. Introduction

With their rapid increase in performance and decrease in cost, computers have fast become a ubiquitous part of our lives, and have been more and more widely used to access, share and exchange scientific, cultural, social and economic resources available not only in the local community but also in the global community. Human-machine dialog systems, as one of communication applications to help people to realize above activities, in the past decades, have been being witnessed to be true in some limited domains. In 1989, MIT developed a prototype of conversational systems [1], and since then such technologies have been used in some common domains, such as Air Travel Information Service (ATIS). But not all such dialog systems could be freely and easily ported across languages and task domains.

As reported in [2], a spoken dialog system generally consists of two large parts, a speech interface part including speech recognition and synthesis, syntactic and semantic analysis, and response generation, and a dialog control part including discourse analysis and dialog control. Both parts are dependent on the language and the task.

To improve the portability both on languages and task domains, two techniques have been investigated. One is to develop the direct translation of semantic feature structures, for example, of a language into these of another language. Such a technique was used in ASURA[3]. The other is to use a pivot language in which the translation of a language into another was realized by first translating a language into a pivot language and then translating the pivot language into another. This technique was used in JANUS [4] and a few multi-lingual spoken dialog systems such as MIT Voyager system [5].

We used the pivot language technique to support the portability of a dialog system across languages for the following reasons.

The first aim of this paper is to make the dialog control part independent of the language. For this purpose we devised to make the discourse analysis of dialogs by using a paradigm of the case frame as a pivot language. This is because it has been used to represent the meaning of sentences, it contained fragments of information necessary for our discourse analysis in the compact forms, and the translation of case frames of a language into case frames of a pivot language is much easier than the translation in other formalisms such as parse trees.

The second aim of this paper is to make the dialog control part independent of the task. For this purpose, in recent years, multi-domain dialog systems such as [6] were reported. [8] used a design strategy separating task-dependent factors to form a Task Description Table (TDT) from the core dialog engine, but tasks dealt with in [6] were simple slot-filling tasks. In this paper, we also tried to separate the algorithm for the dialog control including the discourse analysis from several knowledge sources dependent on the task. The algorithm for the dialog control adopted in this paper is based on the fact that topics in a goal-oriented dialog tend to move according to a task-dependent structure [7]. For a given task we construct a set of topic frames, each of which is composed of a few related topics that might appear in dialogs on the task. A topic frame also contains the method for how to interact with a user about the topics described therein. As a dialog proceeds, several topic frames are activated corresponding to topics included in user's utterances and forms a dialog history on the topic.

Based on the proposed scheme, we built two dialog systems for Chinese and Japanese, both managing spoken dialogs in several task domains, and conducted dialog experiments. The results showed that the proposed dialog control scheme was promising for improving the portability of a dialog system across languages and task domains.

In the following sections, we will briefly illustrate the case frame conversion, the topic frame driven dialog control scheme and the template-based text generation respectively. Then, in section 5, we will show experiment evaluation followed by the conclusion and the future work in section 6.

2. Case Frame Conversion

2.1 A Case Frame

As reported in [3], the dialog controller extracts a topic and a dialog act from a case frame based on a set of task-dependent knowledge bases, performs some necessary actions, creates an

abstract response that contains a template name and necessary arguments. To enable different speech interfaces share a common dialog controller, we embedded a case frame converter to convert such a source language case frame (called *a SLW case frame*) into a pivot language one (called *a PLW case frame*) or vice versa.

Fig.1 shows a paradigm of syntax analysis of Chinese sentence "*wo xv yao jing dian jie shao*(I need the sightseeing guide)". The second line of Fig.1 contains a semantic representation that is described by a list of four terms, (1) a word used to reflect whether the utterance is an affirmative answer [Yes/Ok], a negative answer [No], an ambiguous reply [uhm], or empty item [non], we call such word a *Lead-word*, (2) a main verb word, (3) a modality information specifying the tense and the aspect properties of an utterance, and (4) a set of case elements (hereafter called slots). Each slot indicates one of such relations between a main verb word and a noun phrase/word filled in its slot, like "Agent", "Object" just as seen in Fig.1. Our system only conveys the list containing above four terms from the speech interface to the dialog controller. Apart from the *Lead-word*, we call the list containing only the rest three terms a case frame.

In Fig.1, we can find semantic markers such as "Personal Pronoun" and "FunctionalName" for the noun words of "*wo*(I)" and "*jing dian jie shao*(sight-seeing guide)" respectively. All noun phrases/words included in an utterance are assigned to some slots of the case frame based on semantic markers of the noun phrases/words.

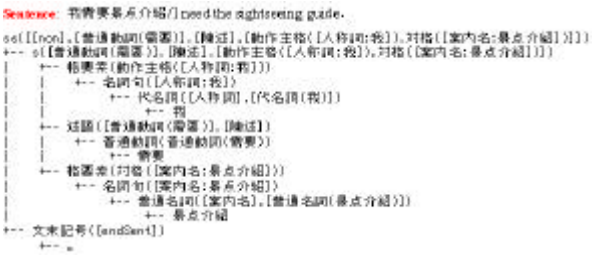

Figure 1. A paradigm of syntax analysis in our system

Here, we show the model of a case frame by *(V, M, C)* where *V* denotes the main verb word, *M* denotes the modality information that contains one to several modality elements, i.e.,

$M = [M_0, M_1, ..., M_n]$.

C denotes one to several case elements, i.e.,

$C = case\_id_1(case_1), ..., case\_id_k(case_k)$.

It also can be wrote as $C = [C_0, C_1, ..., C_k]$.

2.2 Syntactic Analysis

In our system, three kind of knowledge source including a simple thesaurus, a set of case frames and a set of syntactic rules for acceptable sentences are used to parse an utterance. All words that our system can accept are clustered into several groups in POS firstly at shallow level, and then are semantically grouped and organized as a tree structure like a thesaurus at deep level. The concepts of the upper nodes closing to the root node in this structure play roles of semantic markers in interpretation of utterances. Just as seen in Fig. 1, a slot of the case frame is always filled by a single pair-value with the formalism like as [A:B], or by a compound pair-value with the formalism like as [A:B:[C:D]], where A and C are semantic markers in the thesaurus, B and D are detail noun words/phrases contained in user's utterance. More complex recursions can be used to represent relation between two adjacent hierarchical pair-values. It is easy to find such compound pair-values in the experiments showed in section 5, such as U203. In the following sections, we simplified the pair formalism from [A:B:[C:D]] into [A:[C:D]] if A is the same as B.

The syntactic analysis of utterances is performed based on the case grammar in which the meaning of a sentence is represented by a case frame associated with a main verb of that sentence. In our system, case frames and syntactic rules are integrated in a framework of the definite clause grammar (DCG). Comparing with the context free grammar (CFG), one of the differences is that DCG is able to use enhancement items, just as shown in Fig.2.

```
ss(Lead, V, M,C)→
   Lead,{ member(Lead,[yes,no,uhm,ok,non]) },
   s(V,M,C),
    EndSymbol.
s( V, [ M0,..,Mn], [C0,..,Ck] )→
    case_element[ C0 ],
    { case_frame( V,[M0,..,Mn],[C0,..,Ck] ) },
    verb[ V,M0,..,Mn ],
    …
     case_element[ Ck ].
```

Figure 2. A paradigm of DCG grammar

In Fig.2, the second line means that the value of the syntactic constituent "Lead" is a member of the list "yes, no, uhm, ok" or null (non). The 6th and the 7th lines mean that the syntactic constituent C_0 is just the same as the case constituent C_0.

Just as seen in Fig.1, the language parser outputs a hierarchical data structure in which a case frame was attached.

2.3 Implementation of Case Frame Conversion

Our previous Japanese dialog system, SDSKIT-3 [7], conveyed directly whole the syntactic hierarchical structure of an utterance to its dialog controller to make the discourse analysis. To promote SDSKIT-3, we first improved the DCG grammar rules to enable each case frame to contain complete information, and then improved the analysis strategies of the discourse to enable to determine the topic, and the dialog act of user's utterance depending only on the case frame and the *Lead-word* without any additional information.

A case frame converter was adopted in our proposed dialog system. The converter translates a SLW case frame into a PLW one through two kinds of processes orderly and recursively. The first one is called case constituent ordering/pruning process based on a set of rules, and the second one is called noun word replacing process based on a translation table.

3. Topic Frame Driven Control Scheme

3.1 Topic Frame and Dialog Act

Our spoken dialog system has a set of 'topic frames' as a knowledge source on topics. A topic frame forms mutually related topics into a frame that might appear in a task domain. For example in sightseeing dialogs, the name of a hotel, the room charge, the location and so forth form a hotel frame. In our system, a topic frame whose data structure was depicted in Fig.4 consists of a topic frame name, a status id, and one to several slots whose data structure was depicted in Fig.3. Each slot consists of a slot name, a value field (usually empty), a PreCommand and a PostCommand that were triggered before and after the slot has been filled respectively. The priority

item reflects the priority of the slot as to being filled.
The value field is filled by one of a word, a numerical value and a pointer to other topic frame. The ValueType field describes this information. Especially the value field is filled by a word, the semantic marker of the word is given to the ValueType field. The current system has four methods to fill an unfilled slot: (1) to ask a user, (2) to retrieve instances from a database, (3) to use the default value attached to the slot, and (4) to link one of other topic frames. The PreCommand field is described by one of these methods. When a slot has been filled, the dialog controller first takes one of three actions: (1) to move the control to the slot with the next highest priority, (2) to move the control to the topic frame just linked, and (3) to present the information required by a user. The PostCommand field is described by one of these actions.

```
typedef struct slot_message
{   char *SlotName;          // slot name
    char *Value;             // value to fill in
    char *ValueType;         // semantic marker
    char *PreCommand;
    char *PostCommand;
    int     Priority;
    struct slot_message *prior, *next;
} SlotMessage;
```

Figure 3. The data structure of a slot in a topic frame

```
typedef struct Frame
{   char *FrameName; // name of the topic frame
    char *status;        // suspend, closed, ongoing
    int FrameId;
    SlotMessage *SlotMessagePointer;
    struct Frame *prior, *next;
} FRAME;
```

Figure 4. The data structure of a topic frame

For a task domain, a set of task-dependent topic frame is predefined. Since some slots were permitted to be filled with some other topic frames, the set of the topic frames forms implicitly a set of a tree structure, which we call a static topic tree simply. We assume that topics in a dialog move along this tree structure as a dialog proceeds. Thus a dialog forms a subtree of the static topic tree. We call this tree *a dynamic topic tree* which represents the history of a dialog on the topic. We are aware that each utterance in a dialog has its own purpose, that is, an intention a user wants to convey to the dialog system. Hereafter we call this a dialog act. The spoken dialog system has a knowledge source on dialog acts, a state transition network in which a state corresponds to a dialog act. This network describes possible transitions of dialog acts through dialogs. Thus, the discourse history on dialog acts is represented by a state in this network.

3.2 Discourse Analysis & Handling of Topic Frame

Now we are ready to explain the discourse analysis through which the topic and the dialog act of an utterance are identified. The discourse analysis in our system consists of the bottom-up and the top-down analysis.
We use two tables in the bottom-up analysis on the topic. One is called a topic_slot table that describes relations among the name of topic frame, a slot name in the topic frame that should be filled with a word, the semantic marker of that word. For example, a topic frame of "temple", its slot of "builder", and the semantic marker "person". The other is called a verb_focus table that is formed from case frames of verbs. In task domains we treat, a slot filler of a verb with a specific case marker tends to be focused as a topic, that is a filler of some slot of a topic frame. The verb_focus table describes this relation. For example, a verb "build", a slot filler of "agent" a topic frame of "temple", and its slot of "builder". Bottom-up candidates for the topic are decided by looking for each slot filler of the case frame of an utterance up in these two tables. When the bottom-up analysis has produced more than a candidate, the top-down analysis on the topic is invoked.
The top-down analysis uses the dynamic topic tree that represents the discourse history on the topic, and the nodes of which are slots of topic frames on which have been mentioned in the dialog. A certain node in the dynamic topic tree is specified as the current node, which means a subdialog is currently held on the slot corresponding to the node. The top-down analysis searches for some of the bottom-up candidates in the dynamic topic tree equidistantly from the current node. The distance between two nodes is calculated by repeatedly applying the following rules; the distance is equal to 1 if the two nodes are sibling, and 2 if they are a parent and a child.
The node that has the shortest distance from the current node is decided as the topic of the utterance under consideration. When more than a condidate remain, the dialog controller makes a confirmation to the user.
The top-down analysis on the dialog act is simple. Top-down candidates are all the dialog acts which can be reached from the current state in the state transition network on the dialog act.
As for handling of the topic frame, generally speaking, the behavior of the dialog controller is quite simple when it is to speak to a user. It searches for unfilled slot in the dynamic topic tree in the depth-first way from the current node, and tries to fill that slot according to the method described in its PreCommand field. When the slot has been filled, the action described in its PostCommand field is conducted. This action may change the current node. For example, it is the case the action is "to move the control with the next highest priority". In such a case, the dialog controller restarts the depth-first search from the new current node.
When the dialog controller is interpreting user's utterances, that is, conducting the discourse analysis, it searches equi-distantly in the dynamic topic tree as mentioned in the previous section. This may causes to change the current node. In this case the depth-first search also restarts from the new current node.

4. Template-Based Text Generation

Contrasting with a machine translation system, a dialog system works in limited domains, and its response sentences are relatively predictable. One effective approach to the sentence generation for such a dialog system is to concatenate templates after filling slots by applying recursive rules along with appropriate constraints (person, gender, number, etc.) [1]. Based on this idea, we designed a set of templates for Chinese and Japanese text generators respectively. Each one consists of a sequence of word strings and/or slots to be filled by the arguments resulted from preceding process. Table 1 shows several templates for examples listed in section 5. In Table 1, the lowercases "x" and "y" are replaced by the uppercases "X" and "Y" respectively according to the conversion rules of distinct languages, i.e., the lowercases "x" and "y" specify source language words, and the uppercases "X" and "Y" specify pivot language words. Each template could generate a text sentence by selecting one randomly from one to four candidates after their slots have been filled by the input arguments to enable the output be more flexible in perception.

Table 1. Several templates for generating response text

Template name	Input	Text sentence (for different languages)
promInit2PlsSelect	X	Chinese: 這是x 系統。y 中、汝需要那一項服務？ Japanese: こちらはxシステムです。yについて、何なりとお尋ね下さい。
promDayItinerary	X	Chinese: 現在讓我們來確定第x天的行程計画。 Japanese: x日目のコースを決めましょう。
promAlternative	X, Y	Chinese: x 和 y 中、汝期望那一個? Japanese: x と y と、どちらがよろしいですか？

5. Experimental Evaluation

It is not very clear how to design an evaluation metric for above methods. However, we thought it is worthwhile to report the correct responses resulting from our dialog system and the bookkeeping list of the dialogs to demonstrate the feasibility of the proposed methods.

Since the dialog controller has been extended into a language-independent one, we took the language adopted by such a dialog controller as pivot one. To test the dialog system in a general lab room, we invited two groups of subjects who are students studying in the disciplinary of computer speech application but not familiar with any spoken dialog system before the experiments, one group consists of 15 Chinese native subjects, and another group consists of 15 Japanese native subjects. The experiments are conducted in the task domains including sightseeing, accommodation-seeking guide respectively.

After viewing a short demo of this spoken dialog system, the subjects were asked to talk with the dialog system as natural as possible just as to talk with a familiar person. To judge whether the response of the dialog controller are correct or not, we designed a trace interface to show the text sentences resulting from the speech recognizer and the response text generator respectively at real time. Experiments showed that, (1) among 558 correctly recognized Chinese utterances, 471 utterances got fitting responses from the dialog controller; (2) among 537 correctly recognized Japanese utterances, 463 utterances got fitting responses from the dialog controller.

S101: promInit2PlsSelect(京都観光案内,RetX)
→ 這是京都市導遊系統。景点介紹、交通指南、住宿介紹中、汝需要那一項服務？ (This is a guide system in Kyoto city, may I help you in sightseeing guide, traffic guide, or accommodation seeking?)
U101: 我需要景点介紹。(I need the sightseeing guide.)
→ [non], [普通動詞(需要)], [陳述], [動作主格([人称詞:我])，对格([案内名:景点介紹])]
→ [伝達], [観光], [案内:[観光:[]]]
S102: inquireDays(RetX)
→ 汝打算化幾天時間？(How many days do you plan to spend?)
U102: 我打算化両天時間。(I will spend two days.)
→ [non], [普通動詞(化)], [希望], [動作主格([人称詞:我])，期間格([期間:両天時間])]
→ [伝達], [期間], [観光:[期間:[2]]]
S103: promDayItinerary(1,RetX), promAlternative(社寺,庭園,RetY)
→ 現在讓我們来確定第一天的行程計画。寺廟和園林中、汝期望那一個？(Ok, let's plan the first day's itinerary. What are you interested in temples or gardens?)

Figure 5. Chinese experiment in sightseeing domain

As an example, we would show several actual dialogs in Fig.5. For the purpose of understanding, the user's utterances and the generated sentences will be labeled with "Uxxx" and "Sxxx" respectively and will be annotated with corresponding English sentences. We listed the response codes that resulted from the dialog controller, and in which "RetX" and "RetY" are the returned arguments. For the input stream of speech interfaces, we listed user's utterances followed by corresponding *LeadWord* and its SLW case frames, and then followed by the corresponding interpretation results of the utterances that consist of three terms, including dialog act, dialog topic and the focus case element. To read easily, in the following, we use the symbol "→" to identify the beginning of new lines.

6. Conclusion and Future Work

This paper described the methods to improve its portability of a spoken dialog system. The proposed three kernel methods include case frame conversion and template-based text generation to promote the portability across languages, topic frame driven dialog control scheme to promote the portability across domains. So, through separating language-dependent and task-dependent knowledge sources from the system to construct external knowledge bases, the cost to port a dialog system across languages and domains will be confined only in replacing such knowledge bases without changing the dialog controller module. The Chinese and Japanese experiments in several domains demonstrated that; (1) The case frame conversion method is one of rapid and effective methods to promote a dialog system into a multilingual one. (2) The proposed dialog control scheme is able to treat different domains. It is a task-independent dialog control scheme. (3) The proposed template-based text generation method is able to generate very natural sentences, and it is easy to utilize. Therefore, these methods are feasible and effective to improve the portability across domains and languages.

Although the experiments are successful, however, our system is by no means complete. The main reason why a correctly recognized utterance failed to get a fitting response from the dialog controller is either out of the set of case frame definition or out of search range of topic frames. So, in the future, we will extend the vocabulary size and the set of case frame definition, improve the design of the dialog controller.

7. References

[1] Victor Zue, "Conversational interfaces: advances and challenges", Proceedings of 5th European conference on speech communication and technology, pp.KN-9-KN-18. (1997)

[2] Yunbiao Xu, Masahiro Araki, Yasuhisa Niimi, "A multilingual-supporting dialog system using a common dialog controller", Proceedings of the 7th European conference on speech communication and technology (Eurospeech' 2001), pp. 1283-1286 (2001)

[3] Tsuyoshi Morimoto, Toshiyuki Takezawa, Fumihiro Yato, Shigeki Sagayama, and el., "ATR's speech translation system: ASURA", Proceedings of 3rd European conference on speech communication and technology, pp. 1291-1294 (1993)

[4] M.Woszczyna, N.Coccaro, A.Eisele, A.Lavie, and el., "Recent advances in JANUS: A speech translation system", Proceedings of 3rd European conference on speech communication and technology, pp. 1295-1298 (1993)

[5] James Glass, Giovanni Flammia, David Goodine, Michael Phillips, "Multilingual spoken-language understanding in the MIT Voyager system", Journal of Speech Communication, Vol.17, pp.1-18, ISSN:0167-6393 (1995)

[6] Yi-Chung Lin, Tung-Hui Chiang, Heui-Ming Wang, Chung-Ming Peng, Chao-Huang Chang, "The design of a multi-domain Mandarin Chinese spoken dialog system", Proceedings of ICSLP98, Volume 2, pp.41-44 (1998)

[7] Niimi Y., Takinaga N., Nishimoto T., "Dialog Management in a Spoken Dialog System, SDSKIT-3", Proc. of SPECOM' 98, pp.91-96 (1998)