A Dialogue Library for Task-oriented Spoken Dialogue Systems

Masahiro Araki†, Kazunori Komatani‡, Taishi Hirata‡and Shuji Doshita¶

†Department of Electronics and Information Science,
Kyoto Institute of Technology, Kyoto 606-0962 JAPAN
‡Graduate School of Informatics, Kyoto University
¶Department of Science and Technology, Ryukoku University

Abstract

In this paper, we describe a dialogue library for task-oriented spoken dialogue systems. This library was designed for a general task domains. We have classified several domains into three abstract tasks according to the direction of information flow. Subdialogue of the abstract task can be represented by the library which is combined to each application, such as, slot-filler, database and XML. The portability of this library was evaluated in slot-filling task.

1 Introduction

The dialogue management module of current practical dialogue systems tend to be implemented by using relative simple rules, such as, state transition or dialogue grammar [1]. These rules were designed for particular task domain and were paid little attention to the modularity. As a result, it is difficult to port the dialogue manager to another task domain. However, based on recent success of speech recognition technology, spoken dialogue interface is demanded as user-friendly interface for several task domain. Therefore, the needs of the way of easy construction of domain model are now increasing.

The question is how we can implement dialogue manager independent to particular task domain. Our approach to this question is to construct dialogue library for general task domain. The first step of constructing general library is to classify several domains according to the direction of information. As a consequence of the analysis, we have got three abstract tasks: slot-filling, database search and explanation. Next, we specified the type of subdialogue and the type of interaction in each abstract task by referring task-oriented dialogue corpus [2]. In the last step, we examine the portability of this library by constructing a dialogue manager for new domain.

This paper was constructed as follows. Section 2 explains each abstract task according to the direction of information. Section 3 describe the representation of our dialogue library and the application interface. Section 4 describes the evaluation of portability of our approach in the most simplest case. Conclusion and future works are given in Section 5.

2 Abstract Tasks

There are many interactive task domain which is considered to be implemented by spoken dialogue system; e.g. telephone shopping, on-line trade, on-line banking, book order, route direction, tourist information, etc. These task domains can be classified by considering the direction of information flow (see Table 1). In other words, the type of subdialogue and the type of interaction depends on who has the information achieving the purpose.

In what follows, we explain each abstract task.

2.1 Slot-filling

A slot-filling task (Figure 1) is most simple task of dialogue system. Typical domain of this task are telephone shopping, on-line banking, on-line trade, etc. The user has his/her purpose and knows almost all information in order to achieve the task. Most subdialogue of this task are devoted to filling the value at the slots (Figure 2) .

An example dialogue taken from telephone shopping is shown in Figure 3.

Back-end application of this task is slot-filler. Each slot has slot name and may have the constraint to the value. The task is completed when all the necessary slot are filled by appropriate value.

Table 1: Class of task domain

Information flow	abstract task	example domain
user → system	slot-filling	telephone shopping on-line trade on-line banking
user ↔ system	database search	book order bibliography search
user ← system	explanation	route direction instruction manual

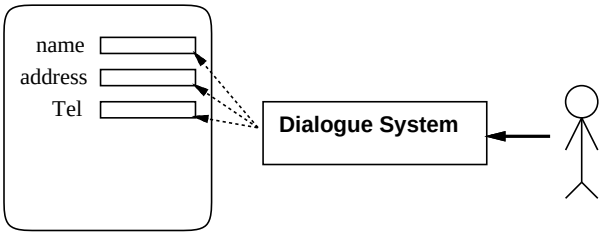

Figure 1: Slot filling task

```
(Opening)
(Slot-filling) +
   (Spell_check)
   (Constraint_violation)
   (Form_selection)*
(Confirmation)*
(Closing)
```

Figure 2: Structure of slot-filling task

O: (Operator) Your telephone number please.
C: (Customer) 0262, 55, 2344.
O: 0262, 55, 2344. Is it right?
C: Right.
O: So, what is your favorite commodity?
C: A steel rack. Code number XC-6.

Figure 3: Example dialogue of slot-filling task (telephone shopping)

2.2 Database Search

A database search task (Figure 4) is popular task for the implemented and demonstrated spoken dialogue systems. Typical domain of this task are Air Traffic Information Service, on-line book order, information retrieval, etc. The user has his/her purpose but knows partial information (Figure 5).

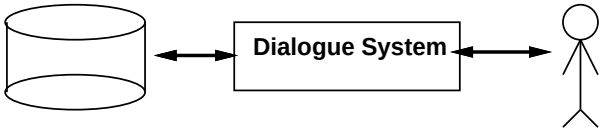

Figure 4: Database search task

```
(Opening)
(Identify_purpose)
(Search)
   (User_clarification)*
   (System_clarification)*
(Search_complete) | (Confirmation)*
(Closing)
```

Figure 5: Structure of database search task

An example dialogue taken from literature retrieval is shown in Figure 6.

Back-end application of this task is database management system (DBMS). The way of information exchange between dialogue system and DBMS is SQL and output record (status). The dialogue proceeds with narrowing the candidate records. The task is completed when the user gets suitable record(s).

2.3 Explanation

An explanation task (Figure 7) has different feature from above two tasks. Typical domain of this task are route

C: (Client) I'd like to find out the paper about dialogue management.

S: (System) There are 2041 articles in database. You have to add another information.

C: Then, add XML and SGML in keyword.

S: There exist no article. The added keywords may be too specific.

Figure 6: Example dialogue of database search task (literature retrieval)

direction, instruction of VCR, etc. The user does not has or has little knowledge, if any, about the content. Most subdialogue of this task are devoted to explanation and confirmation (Figure 9).

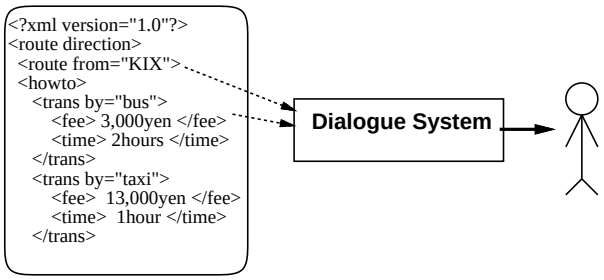

Figure 7: Explanation task

An example dialogue taken from route direction is shown in Figure 8.

C: (Client) I'm at Osaka airport and I'd like to go Kousei-nenkin Hall. How do I go from here?

S: (System) You can take bus or train.

C: Then tell me which train shall I take.

S: You can choose JR train or City train.

C: Which is faster?

Figure 8: Example dialogue of explanation task (route direction)

Back-end application of this task is XML (eXtensible Markup Language) interpreter. The content of explanation task is written by XML. It means that the document which is written by XML can be used in explanation task. Most subdialogue of this task are devoted to giving information and confirming it. The task begins by identifying which part of XML document to be commu-

```
(Opening)
(Identify purpose)
(Explanation)
  (Ask_attribute)
  (Ask_items)
(Closing)
```

Figure 9: Structure of explanation task

nicated, and is completed when the information which is specified to be communicated to the user are explained.

2.4 More Complex Task

Needless to say, above three type of task cannot cover all the domain of spoken dialogue systems. However, we think such tasks and combination of tasks can cover a number of practical domain, which has manageable vocabulary size and manageable pattern of utterance and subdialogue.

For example, the domain of trouble shooting on software installation can be represented combining database search task and explanation task. The first half of the dialogue on software installation trouble shooting should be devoted to identifying the problem (FAQ database search task) and the next half should be operational instruction (explanation task) (see Figure 10).

C: (Client) I have installed TeX and pLaTeX package but could not generate Postscript file.

S: (System) What is your Operating System.

C: Kyoto Linux.

S: Which version do you use?

C: Version 2.2.

S: Can you generate dvi file.

C: Yes, I can. But could not generate Postscript file.

S: Did you install dvi2ps?

C: Yes.

**** Database search phase has finished and Explanation phase begins. ****

S: Then, it may be font problem. Check if pk font file is installed?

C: How do I check it?

Figure 10: Example dialogue of complex task (software installation trouble shooting)

3 Dialogue Library for Subtask Achievement

We have designed dialogue library according to subtask of each abstract task. This can be regarded as dialogue rule and the way of interaction with back-end applications. In this section, we explain our library description language and show example of dialogue library.

3.1 Library description language

Library description language consists of a control structure (if ... then ... else ..., while, switch, etc), utterance types and system-to-application interfaces.

Utterance types are based on discourse tagging scheme (see Figure 11, 12) developed by Japanese Discourse Tagging Working Group [3].

System-to-application interfaces are implemented by function in the library description language. The example of the function are next_empty_slot, insert_slot_value, next_empty_field, etc.

3.2 Dialogue library for abstract task

We have written down the dialogue library for each abstract task by referring the task-oriented dialogue corpus which has discourse tag. The part of library for slot-filling task, database search task and explanation task are shown in Figure 13, 14, and 15 respectively.

4 Portability Evaluation

The dialogue library for slot-filling task is constructed by refering the telephone shopping dialogue corpus. In order to evaluate the portability of the dialogue library, we examine whether this slot-filling library suit the another task. We selected target task is on-line banking. Because we do not have on-line banking dialogue corpus, we have written down the possible dialogue of on-line banking (48 utterances, 16 interaction unit). In principle, one interaction unit (typically, system's initiate and user's response) corresponds to one element of library.

As a result of the evaluation, the library covered 13 interaction unit (81%) but failed 3 interaction unit which is shown in Figure 16.

Case 1 and 2 are derived from the difference of application structure. Telephone shopping usually has one slot form and all slots are filled by user. On the other hand, on-line banking has some slot form (set up a fixed deposit, buy trust fund, money transfer, etc) and has

- Dialogue management
  - *Open*
    The utterance located in the beginning of dialogue which does not contribute problem solving in task, e. g. greeting, signal of beginning of recording.
  - *Close*
    The utterance located in the end of dialogue which does not contribute problem solving in task, e. g. greeting, giving an address of thanks, signal of end of recording.
- Initiate
  - *Request*
    Influencing hearer's future actions. Hearer must reply agree or disagree to speaker.
  - *Suggest*
    Influencing hearer's future actions. Hearer does not have to reply agree or disagree.
  - *Persuade*
    Influencing both speaker and hearer's future actions. Hearer must reply agree or disagree to speaker.
  - *Propose*
    Influencing both speaker and hearer's future actions. Hearer does not have to reply agree or disagree.
  - *Confirm*
    A question about the content which speaker already knows by dialogue context or common sense knowledge.
  - *Yes-No question*
    A question which demands the answer yes or no.
  - *Wh-question*
    A question which demands the value or some expressions.
  - *Promise*
    Mentioning about speaker's future actions.
  - *Demand*
    Mentioning about speaker's goal state.
  - *Inform*
    Telling to hearer about speaker's belief, opinion, etc.
  - *Other assert*
    Expressing gratitude, apology, etc.
  - *Other initiate.*
    Dialogue control statement, etc.

Figure 11: Discourse Tag set (1)

- Response
  - *Positive*
    Expressing affirmation to the *Yes-No question* or acceptance to the *Request* and *Persuade*.
  - *Negative*
    Expressing denial to the *Yes-No question* or refusal to the *Request* and *Persuade*.
  - *Answer*
    Telling the value to hearer's *Wh-question*.
  - *Hold*
    Not directly expressing the response when speaker have to reply something.
  - *Other response*.
    Directly refuse giving response, etc.
- Follow up
  - *Understand*
    Showing acknowledgement or evaluate response. (This tag appears only after response utterance.)
- Response with Initiate
  The element of this category is represented as *Response Type / Initiate Type*.

Figure 12: Discourse Tag set(2)

slot dependencies. This pattern of slot-filling can be regarded as the most complex one. Therefore, by tuning our library to such type of task, we think that many slot-filling domain can be covered.

Case 3 is derived from another problem. As a matter of fact, we cannot write down all the error-and-recovery situation. This is a defect of our approach. However, if we develop learning method which can aquire various pattern of interaction from tagged dialogue corpus, the library gradually covers several pattern of dialogue.

5 Conclusion

In this paper, we proposed a dialogue library method for implementing task-oriented spoken dialogue systems. We have classified several domains into three abstract tasks according to the direction of information flow. Dialogue library was developed for each abstract task and examined its portability in slot-filling task.

Our future work are as follows:

- Developing library acquiring method by tagged dialogue corpus.
- Implementing more general application interface to DBMS and XML interpreter.

```
Slot-filling{
    SLOT = next_empty_slot()
    if(SLOT == NIL)
        then filled-up
        else S: Wh-question
    if(U : Wh-question about "SLOT is VAL")
        insert_slot_value(SLOT, VAL)
        if(SLOT needs SpellCheck)
            Spell_check(VAL)
        Slot-filling
    else
        S: Hold/Wh-question
}

Form_selection{
    S: Yes-No question
    if(U: Positive)
         another_form
         add_form()
         Slot-filling
    else(U: Negative)
         Confirmation
}

Spell_check(VAL){
    S: Wh-question (how spell(VAL))
    U: Answer
    if(not recognize spell(VAL))
        SpellCheck(VAL)
    S: Understanding
}

Confirmation{
    ALL_SLOT = confirm()
    S: Inform (about "ALL_SLOT")
    S: Yes-No question
    if(U: Negative/Inform)
       insert_slot_value(SLOT,VAL)
       Confirmation
    else(U: Positive)
       Closing
}
```

Figure 13: Dialogue Library of Slot-filling (part)

```
Identify_purpose{
    U: Demand | Wh-question
    if(understand)
        then S: Positive
             Search
        else S: Hold/Wh-question
}

Search{
    while((FIELD = next_empty_field()) not empty)
      S: Wh-question
      if (U: Hold/Wh-question)
        then User_clarification(COND)
        else if (U: Wh-question)
             then insert_field_value(FIELD, VAL)
        consistency_check
}

User_clarification(COND){
    U: Hold/Wh-question (about COND1)
    Result = retrieve(WHERE COND and COND1)
    if (RESULT not replyable)
      then System_clarification(COND and COND1)
      else S: Answer (about RESULT)
}

System_clarification(COND){
    (S: Inform)
    S: Hold/Wh-question (COND and COND1)
    if (U: Hold/Wh-question)
      then User_clarification(COND and COND1)
      else if (U: Answer (about VAL))
        then retrieve(WHERE COND and FIELD=VAL)
             if (RESULT not replyable)
               then System_clarification(COND)
               else S: Answer (about RESULT)
}
```

Figure 14: Dialogue Library of database search (part)

```
Identify_purpose{
    U: Demand | Wh-question
    if(understand)
        then S: Positive
             Explanation
        else S: Hold/Wh-question
}

Explanation{
    switch(type(nowpage))
    case: enumerate
      S: Inform
    case: itemize
      if (many_items)
        then S: Wh-question
          if (U: Answer)
            then continue
            else (U: Hold/Wh-question)
                 Ask_attribute
        else S: Wh-question
          switch
          case: (U: Answer)
            continue
          case: (U: Hold/Wh-question)
            Ask_attribute
          case: (U: Hold/Wh-question)
            Ask_items
        S: Inform
}

Ask_attribute{
    U: Wh-question (about ATTR)
    ATTR = sub_attr_search(items)
    S: Answer (ATTR)
}

Ask_items{
    U: Wh-question (about ITEMS)
    ITEMS = items_name(items)
    S: Answer (ITEMS)
}
```

Figure 15: Dialogue Library of explanation

1. case 1
   A: Are you Mr. Kazuma Yamazaki?
   B: Yes.
2. case 2
   A: Which transaction do you deal?
   B: Transfer please.
3. case 3
   A: How much?
   B: 35,000Yen.
   A: You do not have enough money in your account.
   B: How much do I have?
   A: You have 24,888Yen.
   B: Then, transfer 20,000Yen please.

Figure 16: Failed interaction unit

References

[1] E. Mayer, M. Mast, and S. LuperFoy (ed.). Dialogue Processing in Spoken Language Systems. Springer, 1997.

[2] S. Itahashi. Simulated Spoken Dialogue Corpus (CDROM), Research on Understanding and Generating Dialogue by Integrated Processing of Speech, Language and Concept, Vol. 1 – 4, Ministry of Education, Science and Culture, Japan, Grant-in-Aid for Scientific Research on Priority Areas, 1994, 1995.

[3] A. Ichikawa, M. Araki, M. Ishizaki, S. Itabashi, T. Itoh, H. Kashioka, K. Kato, H. Kikuchi, T. Kumagai, A. Kurematsu, H. Koiso, M. Tamoto, S. Tutiya, S. Nakazato, Y. Horiuchi, K. Maekawa, Y. Yamashita, and T. Yoshimura. Standardising annotation schemes for japanese discourse. In *Proc. 1st Int'l Conf. on Language Resourse and Evaluation*, 1998.