THE DESIGN OF DIALOGUE MANAGEMENT IN A MIXED INITIATIVE
CHINESE SPOKEN DIALOGUE SYSTEM ENGINE

XianFang WANG, LiMin DU
{wangxf,dulm}@iis.ac.cn
Center for Speech Interactive Information Technology
Institute of Acoustics, Chinese Academy of Sciences
17 Zhongguancun Rd, PO BOX 2712, Beijing, 100080

ABSTRACT

In this paper, we propose a domain-transparent design of dialogue management in a mixed initiative Chinese spoken dialogue system engine. This design pushes the domain-dependent parts of the dialogue management to the external task configure file, leaving the dialogue manager independent of the domain. The task configure file consists of a set of states each of which is associated with a task action and the constraint to apply the action, not the internal and external resources available for the system. Thus, the count of the states is decreased. It is convenient for designing the dialogue system in a specified domain and porting it to another domain, which is only need to replace the task configure file, leaving the dialogue manager unchanged. Applying this design, the effort of porting a spoken dialogue system across different domain can be relieved.

1. INTRODUCTION

With the development of spoken language technology, a friendly spoken interface will be the next generation man/machine interface [1]. Many spoken dialogue systems have been demonstrated to be effective in various applications, such as travel information service, stock information service. However, there are still many obstacles in the way of using the spoken interface in our everyday life. Significant barriers remain in building effective spoken dialogue system [2]. Speech recognition is a probabilistic process that cannot be guaranteed to be completely accurate and errors are inevitable. Nature language understanding and dialogue processing are still quite limited in dealing with such errors. Additionally, the large variability in the way by which users express themselves in spontaneous speech is difficult to be dealt [3]. The success of a spoken dialogue system depends crucially on a carefully designed interface that can overcome the limitation of current spoken language technology. In order to maximize user satisfaction, the dialogue flow should be natural and efficient.

Dialogue flow is controlled by the component of dialogue management. The task of this component is to determine whether sufficient information has been elicited from the user for task transaction, that is, to communicate with other application to retrieve the required information, to return that information back to the user. This component is also responsible for detecting and repairing breakdowns in the dialogue through verifications, confirmations and corrections. The criterion of judging a spoken dialogue system to be successful and usable is whether it can provide the correct information by controlling the dialogue flow in the way as nature as possible. The meaning of correct information is that the system should collaborate with the user in order to reach mutual understanding of what the user said.

The most straightforward design of dialogue management is to hard-coding the response actions into the system. Although this method is able to build a prototypical system easily, it is difficult to port the system to another domain and to apply this method in a complex dialogue application. Another approach is based on a Finite State Transition Network (FSTN) [4,5]. In this approach, the system directs the user through a predetermined network of states to accomplish the task. Each state specifies a system prompt and a set of allowed responses that the user can make. The determination of the next state is a function of the current state and the user’s response. The domain-dependent part of the dialogue management is pushed to FSTN to make the dialogue manager able to applying a domain-transparent design. Another advantage of this method is that the system prompts encourage the user to say something from a limited set of the possible responses. This simplifies speech recognition and almost eliminates the need for natural language understanding. Various toolkits and authoring environments have been produced based on this method, such as, the Rapid Application Developer (RAD) provided by the Center for Spoken Language Understanding at the Oregon Graduate Institute [6]. However, there are some shortcomings in this method. One disadvantage of this approach is that it tends to be a system initiative dialogue system in which the users are restricted to the system’s predefined path. Because all the dialogue paths are specified in advance, there is no way of managing deviations from these paths. It is not natural and convenient way for most users. Furthermore, it is difficult to associate every dialogue state with corresponding actions and transition arcs because of the very large number of dialogue state that should be associated with the internal and external resources available for the dialogue management in those dialogue systems. The designer must account for all dialogue behavior between the system and the user. It is laborious work for a complex application. Another serious problem is that the user’s answer often over-informative, i.e. it provides more information than required by the system questions. Because the next state is the function of the current state and the information items provided by the user, the redundant information is ignored. The user must reply these information items again in the consequent dialogue flow.
In this paper, we introduce a domain-transparent design of dialogue management in a mixed initiative Chinese spoken dialogue system engine. This method overcomes some shortcomings of the methods discussed in the section above. Being the same as the method based on FSTN, this design pushes the domain-dependent part of the dialogue to the external task configure file, leaving the dialogue manager independent of the tasks. The task configure file also consists of a set of states. But each state is associated with a task action and the constraint to apply the action, not the internal and external resources available for the system in the method based on FSTN. The task action can be regarded as a service provided by the system. Thus, the count of the states is decreased. It is easier to design the dialogue system in a specified domain and port it to another domain, which only replaces task configure file, leaving the dialogue manager untouched.

This paper is organized as follows. First, we introduce the architecture of our spoken dialogue system in section 2. Then, the details of the proposed method are presented. One example is given in the following section. Finally, some discussions are made.

2. DIALOGUE SYSTEM ARCHITECTURE

![Diagram of our dialogue system](image)

The diagram of our spoken dialogue system is represented as figure 1. In our system, the speech input is obtained through the public telephone network. Speech Recognizer recognizes the user's input and generates a lattice of N-best candidate. The component of language understanding then analyzes the meaning of the user and represents it via frame-based representation. Based on these frame representations and dialogue context, dialogue manager fills the information items and decides the user's intention. Then dialogue manager determines whether sufficient and believable information has been elicited from the user for task transaction, if not, detects and repairs breakdowns in the dialogue through verification, confirmations and corrections. If all information required has been elicited, dialogue manager calls the domain-related component and complete the task that user require. The domain-related component generates the response according to the result of transaction. The component of language generation generates the response sentence. The component of text to speech (TTS) transforms it to speech that is consequently returned to the user via telephone. These components except dialogue manager are briefly described in the following sub-sections.

2.1 Speech Recognizer

The speech recognizer in our system is a speaker-independent Chinese spontaneous speech recognition system. In this system, the non-speech events in spontaneous speech are classified and modeled in recognition process to increase the performance of the system. Given the speech recognizer is work in specified domain, some tactics for limited vocabulary are used in this system. The interface between speech recognizer and the component of language understanding is N-best paradigm.

2.2 Language Understanding Component

The task of the component of language understanding is extracting the semantic representation of the content that speaker said. We apply the frame-based representation technique for semantic representation in the component of language understanding. The main knowledge source is the semantic information implemented in the form of case frames each of which contains a set of slots or case values. The constraints are applied at the phrase level. The recognition errors do not provoke the entire parse to fail, simply prevent the specific pattern being matched. Applying constraint at the phrase level is more flexible than recognizing sentence as a whole. Some rather simple mechanisms are applied for handling repetitions and auto-corrections. The dialogue expectations are used in the process of language understanding. The output of the component of language understanding is frame-based representation.

2.3 Language Generator

The task of language generator is to generate text from the response template. To avoid that the system looks like mechanical, more than one template can be specified for one response action. The system selects one randomly to generate the response text.

2.4 Text to Speech

To response user in voice, a Chinese text-to-speech system developed by Institute of Acoustics is integrated in our system. This TTS system is designed for generous uses and takes the
plane text as its input. To increase the performance of TTS, a markup language is developed to transfer additional information that can make the synthesized speech correct and sound more nature.

2.5 Domain-related component

The domain-related component is designed for providing the service supplied by the system, e.g. retrieving information from the database, operating the device. The interface between the domain-related component and the dialogue manager is the callback function. The dialogue manager transfers the information items required and calls the callback function provided by the domain-related component. The domain-related component returns the response to the dialogue manager.

3. DOMAIN-TRANSPARENT MIXED INITIATIVE DIALOGUE MANAGEMENT

In the conversational process, the dialogue manager cooperates with the user to satisfy the user’s requests. To create a portable system, the design of the dialogue manager should apply a domain-independent framework. The domain-dependent factors used for decision-making should be extracted out of the dialogue manager to form an external domain knowledge database. The domain-dependent factors are the services that the system provides and the information items that are necessary for complete the service. The framework of dialogue management should supply a method to specify these services and these information items.

In the following sections, the diagram of dialogue management is introduced first. Then, an example of railway ticket ordering system is given to illustrate how to apply this framework.

3.1 Diagram of dialogue management

In this framework, the system developers can easily specify the dialogue strategy by filling the task action and the conditions to apply the action. The basic idea is that the information items required to accomplish a service is limited and definite. If the user’s intention has not been identified, the system needs to query the user’s intention. When the user’s intention has been identified, it only needs to check whether the value of the information item has been filled or whether the reliability of the value is high enough. Thus, the constraints of the task action are only the information items to be checked. If the value of the information item has not been filled or the reliability is not high enough, the system only needs to query the users, and the system response templates are also appointed. When the check is passed, the domain-related component is called to complete the service. Such a paradigm is quite suitable to the information retrieval system.

3.2 An railway ticket ordering system

The railway ticket ordering system provides the train ticket ordering service. At the same time, the system must provide the train ticket query service to accord the user facility (To simplify the discussion, we assume that the system just permit the user to query whether there are tickets). Thus, the system must know that whether the user want to buy the ticket or just query the ticket information. The information items required are different in these two services. As shown in table 1, the information items required for ordering ticket service include departure station (D_ST), arrival station (A_ST), depart time (D_T), ticket count to order (T_C) and train number (T_N). But for the query service, the information items required include departure station (D_ST), arrival station (A_ST),
depart time (D_T) and train number (T_N). The information item ticket count is not necessary.

<table>
<thead>
<tr>
<th>D_ST</th>
<th>A_ST</th>
<th>D_T</th>
<th>T_C</th>
<th>T_N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Order</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

Table 1. Information items required for services

The symbol ‘+’ means necessary, the symbol ‘-’ means unnecessary.

The system needs to define a response template to query the intention of the user. For one information item, one query response template and one confirmation response template are defined to generate the response. For one service provided by the system, one callback function is given. Thus, for the example above, the domain-related component should provide two functions: one of them is charged with ordering ticket and the other is responsible for query the ticket information.

4. DISCUSSION

Applying this framework, domain-dependent factors used in decision-making are extracted out of the dialogue manager to form an external domain knowledge database, leaving the dialogue manager itself independent of the task. Moreover, porting the dialogue manager to a different domain only requires replacing the domain knowledge database without the effort of re-design.

The basic idea of the design is that the system’s response should be the function of the user’s intention and the information items that have been provided by the user. There is no concept of dialogue state, just the information extracted from the input of the user. The process of information interactive can be regarded as a form-filling process. The user’s intention and the information items are filled during the interactive process between the system and the user. If there are no enough information to transact the task, the system just query the information items that are necessary for task transaction. If the confident of the information items is not high enough, the system just verifies the information items.

One criticism of this method concerns the complexity of the task to be accomplished in the dialogue. This method is suitable for these tasks modeled as form-filling tasks in which the system finds values for slots in a query pattern. Dialogues involving various types of simple inquiry and information transfer, such as travel of information, can be modeled in this way. Other types of dialogue that involve some form of negotiation cannot be modeled in this way. For instance, planning something may requiring the discussion of the constraints that are unknown by either the system or the user. Some negotiation and discussion of constraints that not provided in the proposed method is required in these interactions. Thus, this method is not appropriate for these applications of this type.

5. CONCLUSION

In this paper, we advocate a domain transparent design of dialogue management for a mixed initiative Chinese Spoken Dialogue System Engine. We argue that the method has many advantages concerning the portability of the system, the easiness of developing a system. However, some shortcomings still existed in this method. There are plenty of points for improvement.

This method is currently applied in two applications – a train ticket information system and a stock information system. The result confirms the claims above. In the future, we will focus on the reason ability of the system to make the system suitable for more complex application.

6. ACKNOWLEDGEMENTS

This work was partially funded by the National Key Fundamental Research Project – Image, Speech, Nature Language Understanding and Information Mining. It is a part of its sub-project G1998030505: The Research of the Platform for Chinese Spoken Dialogue System.

7. REFERENCES