A MIXED-INITIATIVE NATURAL DIALOGUE SYSTEM FOR CONFERENCE ROOM RESERVATION

Claudia Pateras, Nicolas Chapados, Remi Kwan, Dominic Lavoie, Réal Tremblay

Nortel Networks
16 Place du Commerce, Nuns' Island, Quebec, Canada, H3E 1H6

{pateras, chapados, kwanremi, lavoied, tremr}@nortelnetworks.com

http://www.nortelnetworks.com/openspeech

ABSTRACT

Telephone based spoken dialogue systems have the potential to automate many routine tasks. Yet, to become widely accepted they must be natural, easy to use, efficient, and robust. We introduce the Conference Room Reservation System (CRRS), a mixed-initiative natural dialogue system used routinely by the employees of our lab. The CRRS allows callers to reserve or cancel rooms by simply stating their constraints in a natural way. The system prompts for missing information and offers alternative solutions if the original constraints cannot be satisfied. Our system uses Nortel Networks' OpenSpeech continuous speech recognizer with barge-in, a robust natural language parser, and a mixed-initiative discourse manager. We describe the overall system with an emphasis on the discourse manager. We present an analysis of real data collected over several months. We then discuss the issue of defining dialogue performance metrics that are independent of the room occupancy rate.

1. INTRODUCTION

In recent years there has been a strong trend in the speech recognition community to create applications with the capability to interact intelligently with humans. That is, it is not sufficient for a system to simply *recognize* natural speech, but it must try to *understand* it, carry out required actions and communicate the results to the human. Early attempts relied on the idea of fixed dialogues wherein the system followed a script, which accounted for all possible actions at each step of the dialogue. Users were expected to be cooperative and adhere to the script. This paradigm quickly reached its limitations and research into more flexible and robust architectures was begun. In this paper we present one such architecture which is at the heart of our Conference Room Reservation System (CRRS) used by employees in our lab. Following a brief description of the entire system, we will focus on the Discourse Manager (DM) component which controls all interactions between user and system. There will also be an emphasis on the performance evaluation of the CRRS and the metrics used therein.

2. RELATED WORK

As Spoken Language Processing applications evolved and moved away from the fixed-dialogue approach, more flexible, robust and general architectures were devised. Although there is no consensus on the correct paradigm, there have been recurring trends and common functionality among current systems. One common aspect is the ability to give more control over the dialogue flow to the user. This idea of mixed-initiative dialogue is more natural and user-friendly but requires more intelligence on the part of the system.

There seems to be a general agreement on many of the tasks that conversational systems must carry out. Typically, the tasks include: an internal representation of the meaning extracted from an utterance; interpretation of the surface meaning within the dialogue context; problem-solving (which includes constraint verification and relaxation); database interaction for information retrieval; response generation. Two representative systems include the AT&T AMICA system [1] and the MIT dialogue manager [2].

Another trend in DM architectures is the use of the theory of plan recognition. In this paradigm, the system attempts to recognize the user's intention based on individual utterances, which are part of the plan of action the user is assumed to be following. By recognizing this plan, the system should be able to attain the user's goals more efficiently by anticipating related sub-goals and needs. The MCC architecture is a good example of this model. For a description of plan recognition and pointers to the original works in the field, refer to [3].

3. SYSTEM OVERVIEW

The Conference Room Reservation System (CRRS) is a telephone-based, spoken language application that allows callers to reserve conference rooms, cancel reservations and request certain kinds of information such as room location, capacity and availability. The overall system consists of 3 main components. The first is the Nortel OpenSpeech Continuous Speech Recognizer (CSR) which produces a word graph tagged with confidence measures. The second is the Natural Language Understanding (NLU) unit, which comprises a robust, bottom-up chart parser. The NLU extracts the meaning from the parse of the top choice embedded in the CSR word graph and returns this meaning to the Discourse Manager (DM) which interprets it and takes the required action(s). The DM also interacts with a database of conference rooms and interfaces with a text to speech synthesizer to produce speech output.

Using natural speech, callers can specify their room requirements. If these are incomplete, the system will prompt for missing information; if they are inconsistent, the system will initiate a clarification sub-dialogue. Context and world knowledge are used to disambiguate responses and to resolve anaphoric references. Once enough information has been gathered and verified, the system accesses the database to find a room satisfying the caller's request. If the initial access fails to return any rooms, constraints are relaxed until a room is found or no more constraints can be relaxed. Database output is organized and presented to the caller in a concise format.

Either the user or the system can initiate error recovery. Recognition confidence measures and illegal constraint values are examples of cues to the system to initiate error-handling sub-dialogues. In addition, a help functionality is available to the user throughout the conversation.

4. SAMPLE INTERACTION

To illustrate some of the functionality, we provide a sample dialogue between a user and the CRRS in Table 1. Although brief, this example does demonstrate some of the key features, namely: anaphora resolution resulting from the second user response and exemplified in the following system prompt; explicit confirmation in system outputs 3, 4, and 5; and time expression handling based on the third user response. Note that the user is not obliged to answer the system's questions. Users can also supply as much or as little information at every turn as they desire.

5. DISCOURSE MANAGER DESIGN

The DM architecture consists of a set of modules, each responsible for part of the basic functionality (Figure 1).

System	Welcome to the CRRS. How may I help you?
User	I'd like to reserve a small room next Tuesday.
System	Many rooms are available next Tuesday, including Bugatti from 8 to 12 and Gauss from 2 to 4.
User	I'll take the second one.
System	Do you want Gauss for the whole time from 2 to 4?
User	No, just for an hour at 2.
System	Please say yes to confirm that you would like to reserve Gauss next Tuesday from 2 to 3.
User	Yes, that's right.
System	Gauss has been reserved next Tuesday from 2 to 3. Do you have any more requests?

Table 1: Sample CRRS dialogue.

Control is passed from the NLU to the DM along with the meaning representation of the top choice extracted from the CSR word graph. The meaning is packaged in a format called the logical form (LF), which is based on first order predicate calculus. We now describe each DM module.

LF Internalization. This module interfaces with the NLU to obtain the LF, which is parsed and stored internally in a frame-based representation. Other modules can more easily manipulate this internal representation.

Expectation Management. This module handles anaphora resolution and other context-dependent interpretation. For example, if the system prompts for a room capacity, the simple answer "ten" will be converted internally to a capacity feature with value 10. These and other context-dependent mapping rules are determined dynamically at the Prompt Generation stage as will be described shortly.

LF Interpretation. This component integrates the new data into the current knowledge of the system and updates the focus of the dialogue. The focus represents the goal at hand and determines what actions the remaining DM modules must take. An embedded Finite State Machine (FSM) handles the focus tracking. Each state represents a goal or sub-goal in the conversation. Transitions are taken as a result of interpreting pieces of the LF or as a result of interpreting database output. Three kinds of rules determine the transitions that can be taken in any given state. Priority rules are state-independent and are always evaluated first. General rules are state-dependent and are part of the static rule base. Dynamic rules are state-dependent and are created at the Prompt Generation stage; they only exist for the following conversation turn. To clarify the role of the FSM, consider Figure 2. In the diagram, rounded rectangles represent states, arrows represent transitions, and the labels on the arrows represent conditions for transitions. The small table depicts the private slots belonging to the "reserve" state. Although only one such table is drawn, each state has its own set of private memory locations to store relevant data. The caption in Figure 2 describes the user sentence that results in the shown snapshot of the FSM.

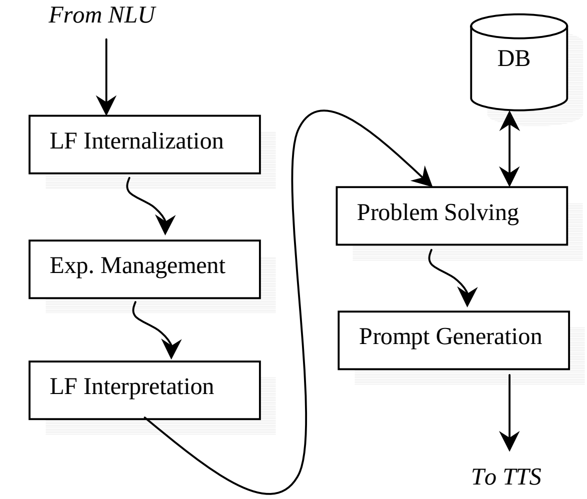

Figure 1: DM main execution loop.

The intention to reserve causes a transition from the initial null state to the “reserve” state. The remaining data extracted from the LF causes self-loops on the “reserve” state while the information is stored in the state’s private data slots. The dotted transition would be taken as a result of a database access returning a single room. When no more rules can fire, the last state transitioned to becomes the focus of the dialogue. This state helps determine the actions of the remaining two modules.

Problem Solving. This module is responsible for all the actions the DM might be required to take. In the CRRS, the main activities comprise building a database query echoing the user’s constraints, relaxing these constraints until the database results are non empty, and making or canceling reservations in the database. The required actions depend on the state in focus at the end of the LF Interpretation stage. Some states have no required actions; in these cases, this module simply updates the knowledge about the current status of the dialogue.

Prompt Generation. This component performs several activities, which are also dependent on the state in focus at the end of the LF Interpretation stage. First, there is the synthesis of the prompt to be provided to the TTS. The phrasing is partially based on templates and partially based on heuristic methods to select the most concise and informative prompt. Second, there is the generation of the context-dependent mapping rules that will be active during the next turn at the Expectation Management phase. Third, there is the generation of the dynamic FSM transition rules that will remain active solely for the duration of the next turn. These dynamic rules allow transitions to be taken based on the user’s response without augmenting the static rule base for the entire dialogue. Finally, the prompt text is sent to the TTS and the CRRS waits for the next user utterance.

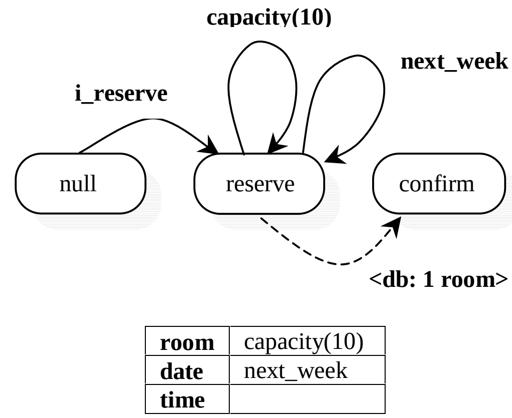

Figure 2: FSM status for “I want to reserve a room for ten people sometime next week.”

6. PERFORMANCE EVALUATION

We collected performance results for employees using the system over the course of several months (Table 2). Note that the “cancellation” and “information request” features came into service a few months after the basic reservation capabilities, and hence the reported usage figures are not representative of the true usage distribution. Since users can perform more than one task (reservation, cancellation, or query) during a single call, the task is our basic unit. A task may fail to complete for several reasons, including a user request to “start over”, a user hang up (most frequent reason), or database inaccessibility. The *task success rate* is simply the ratio of the number of successfully completed tasks to attempted tasks.

CRRS Usage Period	**Sept. 21, 1998 – March 22, 1999**
Number of calls	560
Number of unique callers	132
Number of tasks attempted	881
Number of tasks completed	578
Number of tasks not completed	
Ending with “start over”	80
Ending for other reasons	223
Task success rate	66%
Number of reservations completed	545
Number of cancellations completed	20
Number of information requests	13
Total number of dialogue turns (including both system and user)	10348
Dialogue turns accomplishing tasks	8345

Table 2: Basic Performance Results.

Overall, the CRRS is quite effective in enabling users to reach their goals. In cases where users did not complete a task, an analysis revealed that many users just hung up in an otherwise well-progressing conversation. Hypotheses for these anomalies include user experimentation and external interruption during a call.

Figure 3 shows a histogram of the number of dialogue turns (including both system and user turns) required to complete a reservation task. Figure 4 shows the probability of the user successfully completing the *n*th task, given that s/he has attempted *n*–1 prior tasks. The graph clearly illustrates learning effects: with experience, users become more effective in using the system.

Finally, in order to assess the true performance of a natural dialogue system, we need to address the problem that not all tasks—even in the same domain—are of equal difficulty. Success rate, by itself, is not sufficient to gauge the relative merits of dialogue systems; some measure of intrinsic task complexity must also be included to account for the *a priori* probability that a task could not be completed at all. For conference room reservation, we consider database effects to be most significant. The task success rate will correlate inversely with the number of reservations logged in the database in a date range surrounding the user's target date.

Although insufficient data prevents us from presenting quantitative results at this point, we are pursuing our investigation along the following lines: we start by setting a room occupancy level that determines the difficulty of the task. Next, we model the statistical distribution of the database requests that can be issued by users, along with the conditional probability of the user accepting a room given that some constraint relaxation has been performed. Through Monte Carlo simulation using these estimated distributions, we can determine, for a given room occupancy level, the theoretical maximum performance that the dialogue system can achieve. Finally, we compare the observed system performance to the estimated maximum.

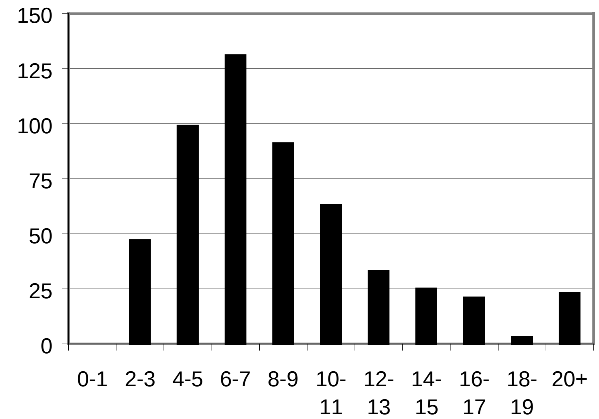

Figure 3: Histogram of the number of dialogue turns required to successfully complete a reservation task.

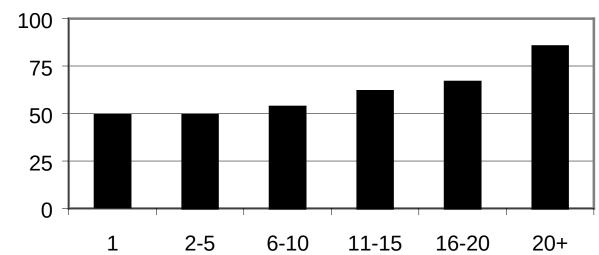

Figure 4: Probability (in percent) of successfully completing the *n*th task if a given user has attempted *n*–1 previous tasks.

7. ACKNOWLEDGEMENTS

We would like to thank Rafah Hosn for her work in the NLU component, and Philippe Carrier for his contribution to the performance analysis.

8. CONCLUSION

This paper introduced the Nortel Networks OpenSpeech Conference Room Reservation System (CRRS), a mixed-initiative, natural spoken dialogue system. We emphasized the Discourse Manager (DM) component which handles all interactions between the user and the other components of the system. Unlike fixed-dialogue systems, the CRRS gives users the freedom to guide the conversation and to provide information in the order they choose. The system acquires missing data and verifies all information. In addition, the CRRS can modify user constraints with the goal of offering alternative solutions when original constraints cannot be satisfied. Performance results reveal that employees in our lab effectively use the system for their conference room needs.

9. REFERENCES

[1] Pieraccini, R., Levin, E., Eckert, W., "AMICA: the AT&T Mixed Initiative Conversational Architecture," *Proc. Of Eurospeech 97,* Rhodes, Greece, Sept. 1997.

[2] Goddeau, D., Meng, H., Polifroni, J., Seneff, S., Busayapongchai, S., "A Form-Based Dialogue Manager for Spoken Language Applications," *Proc. of ICSLP 96*, Philadelphia, Pennsylvania, Nov. 1996.

[3] Miller, B., "Interaction Plans and their Recognition," *Interactive and Collaborative Planning Workshop, in conjunction with The Fourth International Conference on Artificial Intelligence Planning Systems (AIPS-98)*, June 1998, Carnegie Mellon University.