MINOS-II: A Prototype Car Navigation System with Mixed Initiative Turn Taking Dialogue

Munehiko SASAJIMA, Takebhide YANO, Taishi SHIMOMORI, and Tatsuya UEHARA

Multimedia Laboratory,
Corporate R&D Center, TOSHIBA Corp.
{munehiko.sasajima, takebhide.yano, taishi.shimomori, tatsuya.uehara}@toshiba.co.jp

Abstract

Spoken dialogue systems are classified into three types from the viewpoint of turn taking. Dialogue can be led by the system (system initiative), the user (user initiative), and their mixture (mixed initiative). In this paper, EUROPA, a framework for developing spoken dialogue systems, is introduced. EUROPA is applied to prototyping a car navigation system called MINOS-II. MINOS-II deals with a car navigation task of mixed initiative dialogue. First, the system takes the initiative to lead the user to set a route for the destination. Next, while driving along the route, the user takes the initiative and retrieves information about the route freely. MINOS-II is built on a portable PC, can process over 2 million sentence patterns, and is able to respond to a user's question within a few seconds.

1. Introduction

Recently, some car navigation systems are equipped with a voice interface. However, most of them just switch push buttons to voice commands. For example, a voice input "magnify" magnifies a map on the display of the navigation system and "bank" shows icons corresponding to the banks around the car.

The car navigation task consists of two different sub-tasks. One is "set route" and the other is "retrieve route guide." In the former task, a user inputs keys to seek for the destination through a dialogue with the system. The keys for the destination are, for example, name, phone number and address.

If a location is solved by a key, the system calculates a route to the destination. This task is difficult for users since one location can have several keys. For example, the same address can have several names such as "Corporate R&D Center" or "Komukai Operations". As this task is fundamental in car navigation, support for users is required to be sufficient to prevent users from making mistakes in inputting the keys. In the case that users can input keys in several ways, for example, the system should show possible alternatives to users.

On the other hand, in a "retrieve route guide" task, the navigation system should not distract users from concentrating on driving. To force drivers to obey a strict grammar rules or to state a very long command may cause a traffic accident. The driver should decide the timing of the stating of a command to a navigation system, since just requesting a driver to state something may cause an accident. In a "retrieve route guide" task, therefore, the driver should take the initiative in the dialogue. Also, grammar rules should be the less restrictive ones such as those applicable to spoken dialogue.

We have designed EUROPA to be a generic framework for spoken dialogue systems [1]. This paper describes EUROPA briefly and the MINOS-II (Mobile Interactive NavigatiOn System II) application of EUROPA to the car navigation system with spoken dialogue interface. In a "set route" task, MINOS-II leads users to input keys such as nickname, phone number, or address to resolve a destination uniquely. When the destination is resolved successfully, MINOS-II starts a "retrieve route guide" task. In this task, users can ask about the route to the destination freely in spoken language. When the user wishes to change the destination, he/she can move into the "set route" mode, as he/she wants.

2. EUROPA: a framework for building spoken dialogue systems

Figure 1 shows the overall process of man-machine dialogue in EUROPA. It consists of four modules. They are (1) the word-spotting engine which recognizes the user’s utterance and generates a keyword lattice as a recognition result, (2) the BTH (Bun, meaning "sentence" in Japanese, Template Hash) parser which parses the keyword lattice and extracts plausible word-sequences, (3) the dialogue controlling module which resolves a user's question by referring to the knowledge base, and generates an answer to the user's query, and (4) the Text To Speech (TTS) module that plays back the reply to the user’s question by synthesized voice.

To accept spoken language, our voice recognition module does keyword spotting and outputs a keyword lattice. Parsing the keyword lattice, the parser extracts plausible word-sequences. Each of the word-sequences represents the user’s intention. In the case of spoken Japanese, misuse or loss of particles often occurs. Keyword spotting does not deal with them, which simplifies acceptable grammar rules. This characteristic also solves the problems of dealing with unnecessary terms such as "aah" or "well." Just by excluding them from the keyword set, we can accept sentences with these words. Furthermore, change of word order that also occurs in Japanese spoken dialogue can be easily dealt with. The BTH parser is employed for efficiently parsing the keyword lattice, which is obtained by keyword spotting, and it is transformed into a set of possible keyword-sequences. The details of the BTH parser are described in [2].

In the case of the task, it is common for over 100 spotted words to be notified from the recognition engine, and consequently over 1 million possible word sequences can be generated by unfolding the corresponding lattice even if word-
class bi-gram is applied to the lattice. BTH, however, is able to parse such a large lattice within a practical time. Obtaining the set of possible keyword-sequences from the BTH parser, the dialogue-controlling module processes it to generate a reply. The overall configuration of the dialogue-controlling module is depicted in Figure 2.

2.1. Intention Translator Module
It transforms the given plausible word-sequences set into a set of representations of input intentions. Each of the representations is called a user intention.

2.2. Script to control dialogue
In the prototyping and testing cycle, the need for modifications, such as enhancement of domain knowledge or change of task frequently arises, even if the domain and the task do not change. As shown in Figure 2, EUROPA separates modules into two groups. One consists of the domain-dependent modules/data such as a word dictionary, grammar rules, rules for translating user intentions, rules for generating sentences as an answer to the user input, and domain-specific problem solvers, such as Location-solver and Route-solver. The other consists of domain-independent modules/data such as a lattice parser, an interpreter for translating user intentions, and an interpreter for generating answer sentences. To run a dialogue, the dialogue system must control a set of modules, which belong to one of the two groups. For example, a car navigation task that solves a location specified by a user utterance requires not only a generic parser but also a domain-specific problem solver that resolves the location the user intended. Embedding such control codes including domain-specific parts reduces portability of the framework. EUROPA solves this problem by adopting an interpreter for such codes that manage dialogue. We call this interpreter the “USHI Interpreter” which stands for “Unification-based Script Handling Instruction set Interpreter.” A system developer describes the process of meaning analysis, problem solving, and response generation in an USHI script, and the module interprets the script. USHI script language is a Pascal-like language and has the following three features: (1) Subset of statements for expressing selection and looping, (2) unary and binary operators for calculation and comparison, and (3) ability to invoke a function defined in another part of the script or a system-embedded one written in C++. For dealing with feature structures and the knowledge base, the script has two more features.

(1) Unification between feature structures
(2) Access to the knowledge base
Furthermore, to gain enough speed for the response, the USHI script for managing problem solving is compiled beforehand into another form to be interpreted faster, as shown in Figure 2.

2.3. State Manager Module
Dialogue systems based on EUROPA deal with speech models of state transitions. Each of the states is represented by three attributes. They are (1) a user intention that is a trigger of the state transition, (2) a grammar set that will be acceptable in the next state, and (3) answer sentences.
Each of the states accepts a different scale of grammar. The scale of the grammar becomes larger in the free dialogue mode in which a user takes the initiative. On the contrary, when the system takes the initiative, it becomes smaller. In the case of MINOS-II, in the state of a “retrieve route guide” task, acceptable sentence patterns number over 2 million, whereas they number less than 10 in the case of a “set route” task. As the definition of dialogue states depends on each application domain and task, it is loaded from the definition file when the entire system is initialized.

2.4. Response Generator Module
Domain-dependent templates of response are combined to generate an answer sentence for the user. USHI Interpreter controls how to generate it by referring to the definition of the current state and the result of the problem solving.

3. MINOS-II
The authors analyzed car navigation tasks, and implemented MINOS-II, a prototype based on the EUROPA framework.
3.1. Task Model

The task model of MINOS-II consists of a "set route" task and a "retrieve route guide" task. First, a user starts a car navigation system with a destination in mind. The car navigation system moves into the mode of a "set route" task. Following the system guidance, the user inputs keys representing the destination. The keys are (1) nickname, (2) telephone number and (3) address. Using the keys, the system resolves a location uniquely, calculates and sets a route to the destination. After the route is set, the user's car begins to move along the route. Now, the system moves into the "information retrieve task" mode. In this mode, the system accepts the users' queries about the route.

Figure 3 shows a block diagram of MINOS-II. The driving simulator module, the graphical user interface of MINOS-II, simulates movement of the car along a route set by the user. The dialogue control module notifies necessary information to the simulator such as start/stop of the simulation and request for the route set by the user. The simulator also notifies the global position of the car to the dialogue control module.

3.2. Domain knowledge

The dialogue control module of MINOS-II is shown in Figure 2. The USHI interpreter interprets USHI code, implemented by the dialogue system designer, and controls the Intention Translator, the State Manager, the Knowledge Base Manager, and the Response Generator. These modules controlled by the USHI interpreter are domain-independent, and do not change. Only domain knowledge that is used to initialize these modules differs according to the domain and task.

Words dictionary of MINOS-II contains 1002 words and they are classified into 302 word classes. As for the sentences, we analyzed car navigation tasks and selected about 240 phrases for referring to a location, such as "next service area" and "the service area just before the exit". We call the phrase "place expression." The number of sentence patterns is 92, and the number of the combinations of the sentence patterns and the place expressions is over 2 million. The scale of the acceptable sentence patterns also differs according to the state, from 4 to over 2 million.

As for the geographical knowledge, we classified about 40 concepts including road facility such as service area, road and its sub-sections.

When the user of MINOS-II sets a route, these concept models are instantiated and models of facilities along the route are initiated so that they can be referred to by the problem-solving module. A sample route of about 40 kilometers in an urban area in Japan consists of 150 specific road facilities, shops, gas stations, and so on. These domain models are represented in the form of the feature structure.

The dialogue model of MINOS-II consists of 15 dialogue states. The states consists of one initial state, 12 states in which the system takes the initiative to lead the user to input the destination, and 2 states in which the user can ask about the route freely in spoken dialogue. Figure 4 shows a part of the dialogue state definition. When the system asks a user to input the telephone number of the destination, the state of the dialogue moves into this state. An example of a dialogue corresponding to this state is shown in Figure 5. After the system's utterance S1, the state of the dialogue moves into the state 5 in Figure 4.

In this state a user may say, for example,
- "045," an area code of the phone number
- "I do not know," expressing that the user does not know the area code
- "Cancel," expressing that the user found some mistakes in his/her operation and wants to cancel the past operations.

Among them Exp_of_I_DONT_KNOW in Figure 4 corresponds to the intention input by the second one, "I do not know".

3.3. Dialogue example

Figure 5 shows an example of the dialogue. MINOS-II accepts only Japanese, and the figure shows dialogue translated into English. The upper part of the figure, from S1 to S6, is an
example of a "set route" task dialogue. The lower part, from U6 to S8 is an example of a "retrieve route guide" dialogue. Each of the states, especially in the "set route" task, has a different scale of the grammar to enhance recognition ratio. For example, after the system utterance S2, the system must accept the names of all the prefectures in Japan (and their abbreviations), the scale of the vocabulary becomes about 100. On the other hand, in the states corresponding to the utterance U6 to S8, the vocabulary is 300-700 words and the number of sentence patterns is over 2 million.

As shown in Figure 6, MINOS-II runs on a single notebook PC (CPU: Pentium-II, 266Mhz). Among the car navigation tasks, the "retrieve route guide" task has the highest computational cost. In the normal environment such as an exhibition hall, MINOS-II answers almost all questions within 2 seconds, except in the case that an excessively large-scale word lattice is generated because of the heavy noise.

4. Conclusions

We proposed EUROPA, a framework for building spoken dialogue systems, and MINOS-II, an application of EUROPA to the car navigation task model that is mixed initiative. EUROPA, a task and domain-independent framework, is applied to a train ticket reservation task in which the system takes all the initiative [3]. Correction of the dialogue examples

and evaluation of MINOS-II from various viewpoints are subjects for future work.

5. References

