Parallel Computing-Based Architecture for Mixed-Initiative Spoken Dialogue

Ryuta Taguma, Tatsuhiro Moriyama, Koji Iwano, and Sadaoki Furui
Department of Computer Science, Tokyo Institute of Technology
{rtag, oui, iwano, furui}@ furui.cs.titech.ac.jp

Abstract

This paper describes a new method of implementing mixed-initiative spoken dialogue systems based on parallel computing architecture. In a mixed-initiative dialogue, the user as well as the system needs to be capable of controlling the dialogue sequence. In our implementation, various language models corresponding to different dialogue contents, such as requests for information or replies to the system, are built and multiple recognizers using these language models are driven under a parallel computing architecture. The dialogue content of the user is automatically detected based on likelihood scores given by the recognizers, and the content is used to build the dialogue. A transitional probability from one dialogue state uttering a kind of content to another state uttering a different content is incorporated into the likelihood score. A flexible dialogue structure that gives users the initiative to control the dialogue is implemented by this architecture. Real-time dialogue systems for retrieving information about restaurants and food stores are built and evaluated in terms of dialogue content identification rate and keyword accuracy. The proposed architecture has the advantage that the dialogue system can be easily modified without remaking the whole language model.

1. Introduction

Providing spoken language interaction capability as a part of multimedia user interface is believed to add naturalness and efficiency to human-computer interactions. Numerous commercial spoken dialogue systems are currently being deployed, primarily for access to information over the telephone. There are, however, major open research issues that challenge the deployment of completely natural and unconstrained spoken language interactions even for limited task domains. Most of the conventional dialogue systems are implemented by a system-initiative structure imposing constraints on the range and scope of allowed user inputs at any point during an interaction. Since such systems are very troublesome for the users, mixed-initiative systems have also been investigated, in which the course of the dialogue can be changed by both the user and the system at any point [e.g. 1]. These systems need to be able to accept and understand unrestricted utterances at any dialogue state. Such expansion automatically degrades not only the processing speed but also the performance of the system. To alleviate these problems, this paper proposes to implement mixed-initiative systems on a parallel computing architecture, in which multiple recognition systems, separately designed according to dialogue contents (sub-tasks), are run in parallel.

2. System outline

2.1. System components

This paper addresses the problem of the design and implementation of mixed-initiative spoken dialogue systems for information retrieval. We assume that all the inputs of the system are given by speech and that the system uses a display to present information to the user. A basic system structure is shown in Figure 1. The system consists of the following three elements.

- *System controller*

The system controller passes input speech to the multiple recognizers simultaneously. Recognizers return recognition results with scores to the system controller. The system controller selects a recognition result returned by a recognizer that has the maximum likelihood. The system controller thus identifies what kind of dialogue content (sub-task), such as a request or a reply, is spoken. If necessary, the system controller passes keywords detected in the recognition result to the database retriever to get information. The system controller outputs the retrieved information on the display, and waits for the next speech input. The controller also detects the user's request to restart the system or go back to one of the previous stages.

- *Multiple Recognizers*

Multiple recognizers are designed to accept various dialogue contents (sub-tasks). Each recognizer uses a language model that accepts speech for each dialogue content (described in 3.2). These recognizers are driven in parallel under a parallel computing architecture. Each recognizer returns a recognition result including acoustic likelihood and linguistic likelihood to the system controller. In our implementation, speech recognition decoders are run on network-connected Linux PCs.

The *Julius* speech recognition engine [2] distributed by the "Continuous Speech Recognition Consortium" is used in each recognizer. Tied-mixture triphone HMMs with 2,000 states and 16 Gaussian mixtures in each state are used as acoustic models. Speeches from 338 presentations in the "Spontaneous Speech Corpus"[3] uttered by male speakers (approximately 59 hours) are used for training.

- *Database retriever*

Receiving information request from the system controller, the database retriever extracts data matched to the request and returns them to the system controller.

The proposed system architecture has the following advantages.

♦ *Flexible system design*

For accepting new dialogue contents (sub-tasks) or changing the whole task, only language models in the related recognizers need to be rebuilt (added, changed or deleted). This achieves a very flexible system design.

♦ *Real-time dialogue*

Driving multiple recognizers using multiple computers, in which each computer processes only one recognizer, the system controller can get all recognition results at the same time, irrespective of the number of dialogue contents, and process the dialogue immediately. Therefore, real-time dialogue can be easily implemented.

2.2. Task

The task of the system is retrieving information about restaurants and food stores. A user utters a kind of food, a station name and conditions for narrowing down retrieving candidates. The database of restaurants and food stores, which is available in the Internet, is provided by NTT Directory Services Co. The database consists of 80 business categories and data about 4,091 food stores and restaurants. The data includes store names, telephone numbers, average prices, business hours, services, facilities and comments.

2.3. System states

The four system states are defined as follows. The system state transition is illustrated in Figure 2.

System State **A**

In this state, instructions for the user to utter a place and a kind of food are displayed. The state shifts to System State B for verifying the recognized keywords including a place and a kind of food.

System State **B**

In this state, for verifying the place and the kind of food uttered in System State A, "1. Both are correct," "2. The place is incorrect," "3. The food is incorrect," and "4. Both are incorrect," are displayed. The user utters one of these numbers to verify the recognition results. If the user chooses 1, the system shifts to System State C. If the user chooses 2-4, the system shifts to System State A for requesting the user to utter a place or/and a kind of food again.

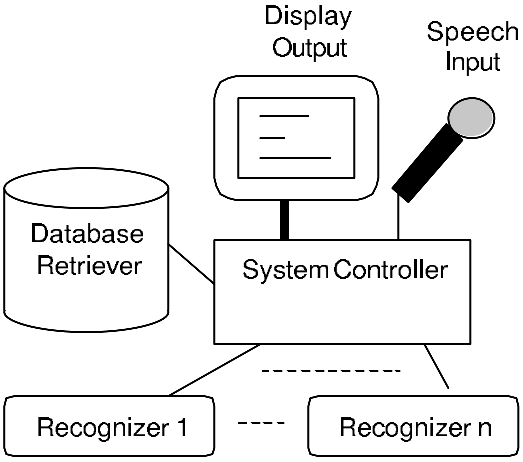

Figure 1. System components

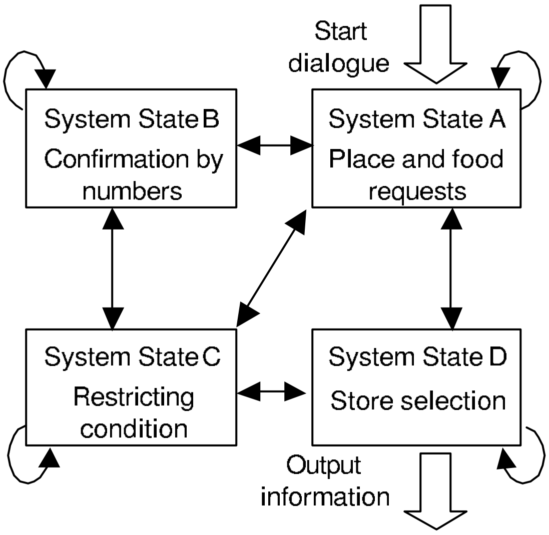

Figure 2. System state transition

System State **C**

In this state the number of the stores that match the user's request and an instruction for the user to utter conditions to narrow down the retrieving candidates are displayed. If the number of the store candidates is less than a displayable limit (10 in this paper), the system shifts to System State D. Otherwise, the system state makes a self-loop.

System State **D**

In this state, the candidate stores are listed up on the display. The user can add more constraints to the store query or utter a store name to see more details about the store, such as its phone number, business hours and available services. The system remains this state until the user stops the system or requests to go back to a previous state.

The user can always utter a command to control the system state transition and finish the system at any system state. If the system controller judges that keywords extracted from the user utterance are unacceptable at a system state, the system controller requests the user to rephrase his last utterance and remains in the same state.

3. Dialogue contents and language models

3.1. Expected dialogue contents

User utterances are categorized into the following five dialogue contents (sub-tasks).

P&F: Places and kinds of food

The user utters a place and a kind of food at the beginning of the dialogue. In this category, station names and words such as "bar" and "Japanese restaurant" are treated as keywords.

e.g. *"I'd like to eat Sushi near the Meguro station."*

RES: Restrictions to narrow down the constraints

The user utters this content to narrow down too many matching stores or restaurants. Conditions, such as business hours, available services, prices and closeness to the station, are treated as keywords.

e.g. *"Lower than 3,000 yen."*

STR: Store names

The user chooses a store from a list of matching food stores or restaurants and utters the store name. All the store names are recognized as keywords.

e.g. *"Edo-sushi please."*

COM: Commands

Irrespective of a system state, the user can utter comments to control the system state, such that returning to a previous state, restarting or stopping the system.

e.g. *"Go back." "I'd like to correct the station name please."*

NUM: Numbers

The user can utter a number to make a choice. The numbers are used for verifying the recognition results of the place and the kind of food and for choosing a store.

e.g. *"Um, number one." "Three."*

3.2. Language models

Language models consisting of class bigrams and reverse class trigrams with backing-off are used. The models are trained using text corpora that are prepared separately for each dialogue content. Some training texts are transcribed from real dialogue speeches, and others are manually written by several human subjects on the assumption that they are actually using the dialogue system. The first set is called real transcriptions, and the latter is called artificial transcriptions. Several sets of words, such as numbers, store names, fillers and prices, are respectively grouped and class language models are made. Words belonging to each class are given an equal word occurrence probability.

Five language models (LMs) are made according to dialogue contents as follows.

LM-P: LM for places and kinds of food (P&F)

This model accepts utterances including keywords of places and kinds of food. Training texts are accurately transcribed from 539 dialogue utterances of 35 speakers. Station and food names are combined to STATION and FOOD classes, respectively. The STATION class has 1609 names of Tokyo area stations. The FOOD class has keywords related to 80 kinds of food.

LM-R: LM to restrict retrieving conditions (RES)

This model accepts utterances that restrict retrieving conditions, such as "Near the station," and "I want to use a parking lot." Training texts are 564 artificial transcriptions by 14 subjects. Various words and phrases indicating business hours and prices are classified as TIME and PRICE.

LM-S: LM for store names (STR)

This model accepts utterances including store names. 18 artificial transcriptions made by one subject are used as training text. Store names are combined to STORE class. The STORE class has 4,491 store names.

LM-C: LM for commands (COM)

This model accepts commands for controlling system states. Training texts are 68 artificial transcriptions.

LM-N: LM for numbers (NUM)

This model accepts utterances including numbers. All acceptable numbers are combined to NUMBER class. All words included in the NUMBER class are recognized as keywords. Training text consists of 17 artificial transcriptions.

4. Mixed-initiative dialogue systems

The two following mixed-initiative online real-time dialogue systems have been built and evaluated. Both of them are using the common acoustic model and language models.

SYSTEM-1

This system allows users to use numbers for selection. At the System State B, the system requests the user to confirm the user utterance (recognition results) madeat the System State A. The user utters one of the numbers assigned to the suggested choices. At the System State D, the system displays a store list with numbers. The user can choose one of the suggested store names by saying either the store name itself or the labeled number, and then the system controller displays complete store information. The system returns to a previous state or restarts whenever requested by the user.

SYSTEM-2

This system does not have the System State B. The system shifts to the System State Cdirectly from the System State A. At the System State C, the system accepts utterances to correct kinds of food or/and station names. If the user utters a command such as "I want to correct the kind of food" and "The station name is wrong", the system shifts back to the System State A and requests the user to utter again a kind of food or/and a station name. At the System State D, the system doesn't display selection numbers. Instead the user is requested to utter a store name itself. SYSTEM-2 is simpler and more natural than SYSTEM-1, but tougher from the recognition point of view.

5. Dialogue content detection

The system controller detects a dialogue content (subtask), that is, understands the meaning of the input speech, by selecting a language model having the maximum likelihood score. The probability $P(w,c|x)$ is calculated as follows.

$$P(w,c \mid x) = \frac{P(x \mid w,c)P(w,c)}{P(x)}$$
$$= \frac{P(x \mid w,c)P(w \mid c)P(c)}{P(x)}$$

Here, c is a dialogue content, w is a word sequence, and x is an input speech. Therefore the process of detecting a dialogue content c and deciding a word sequence w is equivalent to maximizing the P defined below.

$$P = P(x \mid w,c)P(w \mid c)P(c)$$

where $P(x|w,c)$ is the acoustic likelihood and $P(w|c)$ is the linguistic likelihood, both given by the recognizer that uses a language model for the content c. $P(c)$ is the likelihood that the system makes a transition to that state. In this paper, $P(c)$ for each system state is determined by calculating the number of occurrences of the dialogue content.

Table 1. Dialogue contents of test set

	SYSTEM-1	SYSTEM-2
P&F: Places, Food	83	117
RES: Restrictions	119	177
STR: Store names	96	90
COM: Commands	175	253
NUM: Numbers	99	0
ALL	572	637

Table 2. Specification of content-dependent language models

	Vocabulary size	Perplexity	Unknown words
LM-P	1617	49.9	53
LM-R	721	12.0	114
LM-S	4587	21.2	4
LM-C	108	6.9	70
LM-N	120	6.0	5

For the actual dialogue content detection, the following weighted score P is used.

$$P \approx P(x \mid w,c)P(w \mid c)^a \, P(c)^b$$

Here, a is a linguistic score weight and b is a content score weight.

6. Experiments

Input utterances automatically segmented into sentence utterances are digitized with 16kHz sampling and 16bit quantization. Feature vectors have 25 elements consisting of 12 MFCC, their delta, and delta log energy. The CMS (cepstral mean subtraction) is applied to each utterance.

For evaluating the SYSTEM-1 and the SYSTEM-2, 1,209 dialogue speeches are used. These speeches are categorized by their contents as shown in Table 1. Vocabulary sizes, test-set perplexities, and numbers of unknown words inthe test set are listed in Table 2.

Three dialogue content detection methods using the following likelihood scores are investigated.

A+L: Acoustic + Linguistic scores

This method assumes all dialogue contents occur with the same probability at any system state. This means that $P(c)$ is constant, and b =0. The optimum value of the weight a is determined experimentally.

A+L+C: Acoustic + Linguistic + Content scores

This method incorporates the content score to the previous method A+L. Content occurrence probability $P_s(c)$ of a system state s is defined as follows.

$$P(c)=P_s(c)=\frac{N_s(c)}{\sum_c N_s(c)}$$

$N_s(c)$ is the number of sentences with a dialogue content c. The optimum values of the weight a and b are determined experimentally.

KNOWN: Dialogue content is known

This is the case in which the system controller knows the content c of the input speech at any system state. Therefore, the system controller does not need to compare the recognition results and scores that are returned from the recognizers. Instead, the recognition result and the score returned by the recognizer corresponding to the known dialogue content c is used. This method is equivalent to the case where the dialogue content identification rate is 100%.

The SYSTEM-1 and the SYSTEM-2 are evaluated using these three dialog content detection methods. In Figure 3, the lines indicate dialogue content identification rates and bars indicate keyword recognition accuracies. As described above, the identification rates of the KNOWN condition are 100%. It is observed that by adding the content score (A+L+C), the content identification rate is improved by 5.2% (averaged over the SYSTEM-1 and the SYSTEM-2) compared to the A+L condition. Similarly, the keyword accuracy of A+L+C is 2.7% higher than A+L. The keyword accuracy of KNOWN, that is the condition in which the dialogue content is known, is 1.9% higher than A+L+C. This means that if the dialog content detection method is improved, the keyword accuracy will be further improved.

The keyword recognition accuracy for each dialogue content is shown separately for the two systems in Figures 4 and 5. It is observed that the content score always increases the keyword accuracy. Since the dialog content NUM (Numbers) is not uttered in the SYSTEM-2, the keyword accuracy of NUM is not available in Figure 5. These results show that keyword accuracies with the SYSTEM-1 are higher than that with the SYSTEM-2 especially for the dialogue content c. This is because the SYSTEM-2 does not confirm the recognition results with numbers but only using natural speech, thus increasing recognition difficulty. In addition, the keyword accuracy of the speech content NUM is very high. The difference of the keyword accuracies are decreased if utterances with the dialogue content NUM are removed. However, the preliminary results of our subjective evaluation experiments indicate that most of the users prefer the SYSTEM-2 even though its keyword accuracy is lower.

In these experiments, the content scores are calculated using the evaluation data for the method A+L+C. To make

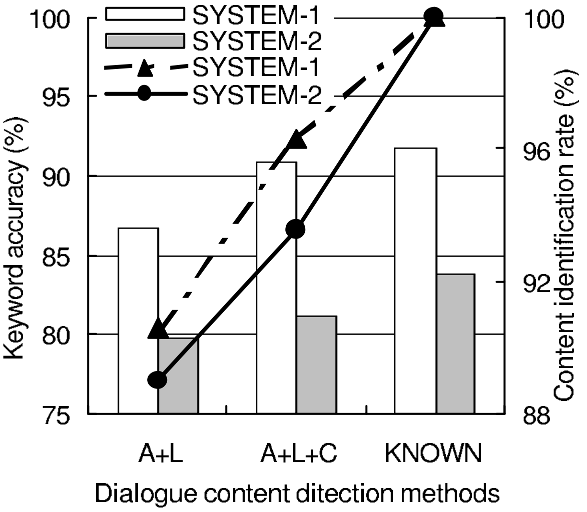

Figure 3. Results of proposed method

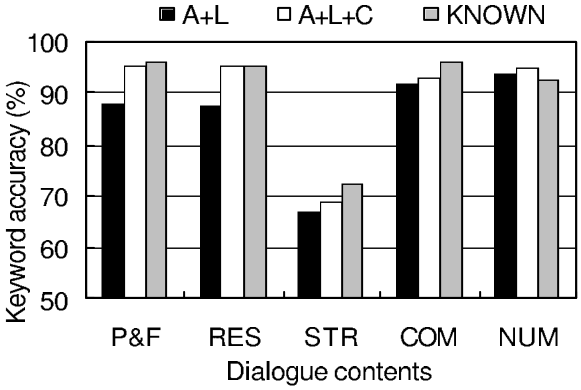

Figure 4. Results of **SYSTEM-1**

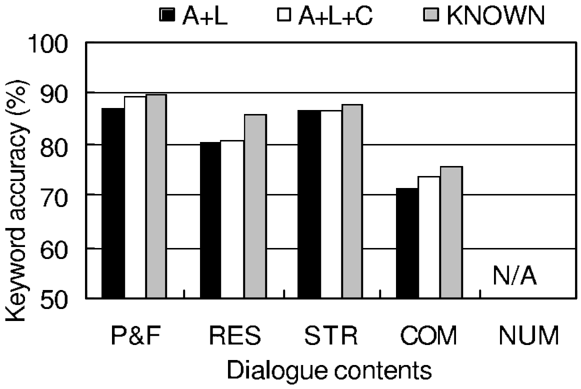

Figure 5. Results of **SYSTEM-2**

a fair evaluation, another dialogue content decision method named A+L+C(O) in which the scores are determined by rules independent of the evaluation data is introduced as follows.

At any system state, messages on the display suggest the users what kinds of utterances can be accepted by the system. Users generally follow these suggestions. Considering this fact, content scores are decided by the following rules:

> *At each system state, all the dialogue contents of user utterances are assumed to be included in the contents listed in Table 3. That is, the users never say other content utterances. In addition, all the contents are assumed to have the same occurrence probability at each system state. Scores of contents not shown in are set at zero.*

Figure 6 shows the dialogue content detection rate and the keyword accuracies for all the content detection methods. In this figure, the scores of SYSTEM-1 and SYSTEM-2 are averaged. The keyword accuracy of the closed-method A+L+C is 0.14% higher than the rule-based open-method A+L+C(O). Closeness of the keyword accuracies is due to the fact that the occurrence probability distribution of the evaluation set is very similar to the rule-based values at each system state. With the dialogue systems which accept more unrestricted utterances, it is expected that the keyword accuracy of the dialogue content detection methods A+L+C, which estimates content occurrence distributions, will become much higher than the rule-based method A+L+C(O).

Table 3. Acceptable dialogue contents at each System State in the open-method

	SYSTEM-1	SYSTEM-2
System State A	P&F, COM	P&F, COM
System State B	NUM, COM	
System State C	RES, COM, NUM	RES, COM
System State D	STR, RES, COM, NUM	STR, RES, COM

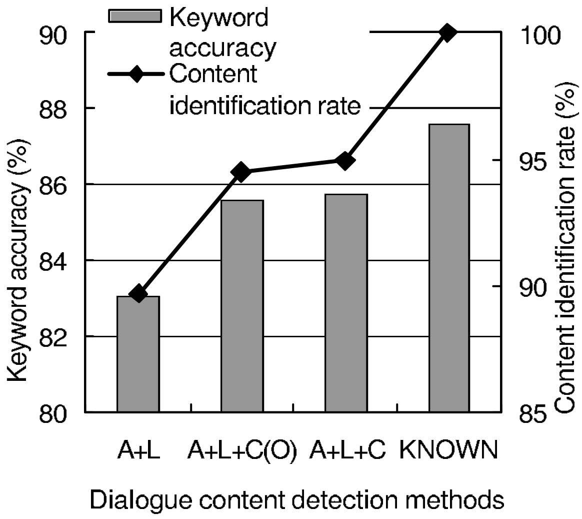

Figure 6. Results including the additional experiment using the rule-based content scores

7. Conclusions

Two on-line mixed-initiative spoken dialogue systems have been implemented and evaluated using a restaurant and food store information retrieval task. For accepting users' unrestricted utterances, multiple language models according to dialogue contents have been built. The language models are used in multiple recognizers driven under a parallel computing architecture. Likelihood scores, incorporating state likelihood scores, given by the recognizers are used to detect a dialogue content. The proposed architecture has the advantage that the dialogue system can be easily modified. Two dialogue systems with different dialogue flows have been built without re-building the whole language models. The two systems have been evaluated in terms of dialogue content identification rate and keyword accuracy, and the effectiveness of the proposed methods has been confirmed.

Although such architecture offers the advantage that real-time systems can be easily implemented, the cost of the systems infrastructure can be relatively high. However, since the cost of computers is getting smaller, it is expected that the merit of such systems using multiple computers will become significant and that such systems will become popular in the ubiquitous computing era [4].

8. References

[1] E. Levin, et al., "The AT&T-DARPA COMMUNICATOR mixed-initiative spoken dialog system", *Proc. ICSLP2000*, vol. 2, Beijing, China, Sep. 2000, pp.122-125.

[2] A. Lee, T. Kawahara, and K. Shikano, "Julius — an open source real-time large vocabulary recognition engine", *Proc. EUROSPEECH2001*, vol.3, Aalborg, Denmark, Sep. 2001, pp.1691-1694.

[3] T. Shinozaki, C. Hori, and S. Furui, "Towards automatic transcription of spontaneous presentations", *Proc. EUROSPEECH2001*, vol.1, Aalborg, Denmark, Sep. 2001, pp.491-494.

[4] S. Furui, K. Iwano, C. Hori, T. Shinozaki, Y. Saito, and S. Tamura, "Ubiquitous speech processing", *Proc. ICASSP2001*, vol.1, Salt Lake City, U.S.A., May 2001, pp.13-16.