

論文 / 著書情報
Article / Book Information

Title	Multimodal Telephone Directory Assistance System and Its Evaluation
Author	Yasuhiro Minami, Osamu Yoshioka, Kyohiro Shikano, Sadaoki Furui
Journal/Book name	International Workshop on Human Interface Technology 94, Vol. , No. , pp. 7-14
Issue date	1994, 9

Multimodal telephone directory assistance system and its evaluation

Yasuhiro MINAMI, Osamu YOSHIOKA, Kiyohiro SHIKANO*, and Sadaoki FURUI

NTT Human Interface Laboratories
Musashino-shi, Tokyo 180, Japan

ABSTRACT

To evaluate our algorithm for large-vocabulary speaker-independent continuous speech recognition systems and to collect actual dialog data between a computer and a user, we have implemented a multimodal dialog system for telephone directory assistance. To evaluate the dialog system, we used twenty subjects using twenty telephone number retrieval tasks per subject. In our experiments, the task completion rate was 99%. About 75% of the users said they preferred this system to looking up telephone numbers in a telephone directory.

1. INTRODUCTION

One of the most useful and successful applications of speech recognition technology is database inquiry systems. For these systems to be useful, they must be able to recognize utterances spoken in a spontaneous manner. They must also have a large vocabulary because if they have only a small vocabulary, their performance will be no better than that of a menu-input system.

Several approaches to large-vocabulary continuous speech recognition having more than 10,000 words have recently been investigated (Kenny et al., 1992, Ney et al., 1992). Kenny et al. (1992) reported recognition results for a "books on tape" task with a 60,000 word dictionary. Ney et al. (1992) reported an algorithm for a 10,000-word vocabulary continuous speech recognition system with application to a database query task. In spontaneous utterances, users tend to add interjections and incorporate various expressions. Although the ARPA project in the U.S. has been paying a lot of attention to spontaneous speech recognition (Hirschman et al., 1993), recognition of spontaneous speech having a vocabulary of more than 10,000 words has not been tried yet.

This paper reports on the design and evaluation of a large vocabulary spontaneous speech recognition system. To achieve efficient parsing of sentences that include various expressions, the HMM-LR algorithm (Kita et al., 1989), which can analyze context-free grammar, was employed (Minami et al., 1992). Applying

this algorithm to large-vocabulary continuous speech recognition requires (1) accurate scoring for phoneme sequences, (2) reduction of trellis calculation, and (3) efficient pruning of phoneme sequence candidates (Minami et al., 1992).

To investigate the user's satisfaction with speech recognition and to improve its performance, it was necessary to implement an actual dialog system using speech recognition and to have users evaluate it. To improve the performance of the system, it was also necessary to collect dialog data during the evaluation. There are several approaches in this field. In the air travel information service (ATIS) domain, Shriberg investigated user satisfaction and system performance [1]. Takebayashi implemented a multi-modal dialog system for a hamburger shop task. However, Shriberg's approach does not use multi-modal input devices. In a multi-modal dialog system, the system can select the appropriate input device according to its features. When the user must select a keyword from a small selection, a mouse is useful. On the other hand, when the user must select a keyword from a large selection, speech recognition is useful. Takebayashi's approach used small-vocabulary speech recognition. This approach did not make the best use of the advantages of speech recognition. We implemented a multi-modal dialog system for telephone directory assistance using a large-vocabulary speech recognition system.

With this kind of system, background noise is a big problem in recognizing speech. One simple solution is to make HMMs from noisy speech, but this needs a lot of calculation. Several new noise adaptation techniques, called HMM decomposition or HMM compensation, that create HMMs for noisy conditions from speech HMMs and a noise HMM have been proposed (Varga and Moore, 1990; Gales and Young, 1992). We showed that the HMM composition technique performed very well for continuous speech recognition; this technique is almost the same as HMM compensation (Martin et al., 1993). We use this technique for adapting the system to noisy conditions.

In Section 2, we describe our large-vocabulary speech recognition algorithm for telephone directory

* He is now with Nara Institute of Science and Technology,
Nara-shi, Nara 630-10, Japan.

assistance. In Section 3, we discuss a multi-modal dialog system that uses our algorithm.

2. SPEECH RECOGNITION ALGORITHM

2.1 TELEPHONE DIRECTORY ASSISTANCE TASK

Since a database inquiry task is a good application for large-vocabulary speech recognition, we adopted a telephone directory assistance task to test our algorithm. It covers two cities and contains more than 70,000 subscriber names. While the grammar needed for this task may appear to be simpler than general dialog, we found that the various expressions used for simulated telephone directory assistance inquiries allowed us to investigate and collect a variety of dialog data. It is worthwhile studying this task. In this task, only the subscriber name is indispensable in each utterance. There are no constraints on the combination of addresses and subscriber names because users may sometimes say the wrong address. For example, the user may say the name of the train station nearest to the subscriber's house instead of the real address.

2.2 TWO-STAGE LR PARSER

Our algorithm is based on the HMM-LR. The generalized predictive LR parser is used as a language model for phoneme prediction. It uses an LR table, which is pre-compiled from context-free grammar rules, to predict phonemes. The flow of the recognition process is as follows:

(1) The LR parser predicts all possible phonemes using the LR table. It then transfers them to an HMM phoneme verifier.

(2) It next calculates their trellis likelihood scores by using their HMMs.

(3) The state of the candidate then moves to the next state in the LR table according to *shift* and *reduce* operations.

(4) During steps (1)-(3), all possible phoneme sequence candidates are constructed in parallel. These phoneme sequence candidates are pruned by using their HMM likelihood. The likelihood calculation is based on a trellis algorithm (not the Viterbi one) in order to match the likelihood calculation method with the training method. The log probabilities are used as the likelihood scores.

(5) Steps (1) to (4) continue until the parser does not predict any more phonemes. It then outputs the recognition candidates that have been accepted by the grammar and that have the highest likelihood score.

We have developed a two-stage LR parser that uses two classes of LR tables: a main grammar table and several sub-grammar tables (Figure 1) that are separately compiled from a context-free grammar. The sub-grammar tables contain such keywords as city name, town name, block number, and subscriber name. The

main grammar table controls the relationships between the keywords. The main parser also analyzes meaningless words: interjections, pauses, and several expressions commonly used in directory assistance dialogs. Dividing the grammar into two classes this way has two advantages. Since each grammar can be compiled separately, the time needed for compiling the LR table is reduced; and the system can easily extract the meaning of the utterance by looking at keywords in sub-tables without considering meaningless words, such as interjections.

2.3 ACCURATE AND EFFICIENT SEARCH ALGORITHM

(1) ACCURATE SCORING

Our algorithm uses not only a forward trellis but also a backward trellis to accurately calculate the score of a phoneme sequence candidate. The backward trellis probabilities, $b(i)$, $1 \leq i \leq \text{end frame}$, are used as the estimated probabilities of potential succeeding phoneme sequences (heuristics).

The phoneme verification process is as follows. First, first-level trellis probabilities from the end of speech are calculated for all phonemes. Then, the best phoneme that obtains the maximum probability normalized by the number of frames is chosen. The trellis probabilities of the phoneme are used as initial values of the next level trellis calculation. This process continues up to a sufficient number of levels. Consequently backward trellis probabilities for each level are obtained. Finally $b(i)$ is obtained by choosing the maximum probability among all levels. In the forward trellis calculation, the LR parser predicts the next phonemes according to the LR tables. Their forward trellis probabilities ($a(i)$, $1 \leq i \leq \text{end frame}$) are calculated by verifying the predicted phoneme HMMs against the input speech. The phoneme sequence scores are calculated by taking the trellis convolution of the forward and backward trellises as follows:

$$\text{SCORE} = \sum_{i=1}^{\text{end frame}} a(i)b(i) \quad (1)$$

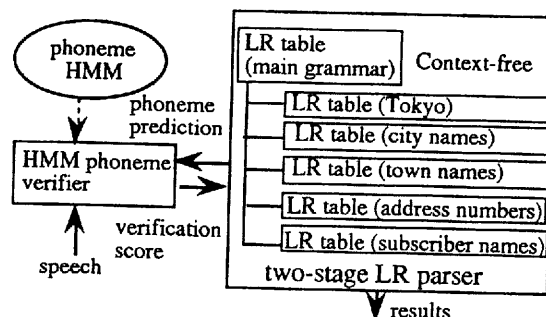


Figure 1. Structure of the continuous speech recognition system.

After all the phoneme sequence scores have been calculated, the candidates are sorted according to their score. Candidates below a certain rank (beam width) are then pruned (beam search). This beam search procedure is iterated from the beginning to the end of the input speech according to the LR tables.

(2) ADJUSTING WINDOW

Since the conventional HMM-LR algorithm calculates the trellis probabilities against all input frames of the predicted phonemes, it takes a lot of calculation. If the probable location of a predicted phoneme is known, the algorithm need only calculate the trellis probabilities for that location. We therefore propose an algorithm for determining an adjusting window that restricts trellis calculation to the probable part of the input frame for each predicted phoneme (Figure 2). The adjusting window (shaded rectangle in Figure 2) has a length of 50 frames (400 ms). The initial area of the trellis calculation for the predicted phoneme is restricted to the forward trellis probabilities within this window. The ending frame area of the predicted phoneme is restricted to 120 frames (the bigger rectangle at the end of the predicted phoneme); this frame size is selected because it is sufficiently large to contain the ending frame of the predicted phoneme. Trellis calculation is thus restricted to the region between the initial and ending frame areas, significantly reducing trellis calculation from the first to the end frame. To calculate the phoneme sequence score, the convolution of the forward and backward trellis probabilities is calculated as follows, instead of using equation (1):

$$\text{SCORE} = \sum_{i=j}^{j+49} a(i)b(i), \quad (2)$$

where j is the first frame of the adjusting window. In this procedure, the probabilities in the backward trellis are multiplied by $(1-\epsilon)$, where ϵ is a small positive value. To determine the best position for the adjusting window, the scores of each frame are calculated one by one using equation (2), varying the value of j . The best position is where the maximum score is achieved. By using ϵ , the adjusting window is usually correctly assigned to the phoneme boundary between the predicted phoneme and the next phoneme in the input speech frame. After determining the adjusting window position, the forward trellis probabilities within the adjusting window, plus an additional 20 frames of trellis probabilities (to cover the area of the phoneme boundary), are stored as the initial values of the next trellis calculation.

Determining the position of the adjusting window requires additional calculation. To reduce the amount of computation needed for the logarithmic calculation of equation (2), logarithmic function tables were used. To

avoid under-flow errors, we used logarithmic calculation for equation (2).

(3) MERGING PHONEME SEQUENCES AND GRAMMATICAL STATES

The LR tables need multiple pronunciation rules to cover allophonic phonemes, such as devoicing and long vowels in Japanese pronunciation. However, these multiple rules cause an explosion of the search space. To make the search space smaller, we merge phoneme sequence candidates as well as grammatical states when the following conditions hold:

- (a) the grammar stacks of the candidates are the same,
- (b) the candidates are in the same state in the LR tables, and
- (c) the phoneme sequences of the candidates are the same, disregarding devoicing and long vowels.

To further reduce the amount of calculation, we do not recalculate the forward trellis probabilities of a phoneme sequence candidate when the same phoneme sequence already exists, even if the grammar stacks of the candidates are different.

(4) MERGING CANDIDATES AT THE MEANING LEVEL

For spontaneous utterances, the top N recognition results tend to have the same meaning. Occupation of the top N candidates by candidates with the same meaning often result in the correct candidate being pruned. We therefore need to merge candidates with the same meaning into a best-score candidate. In the two-stage LR parser, the main grammar does not include any key-

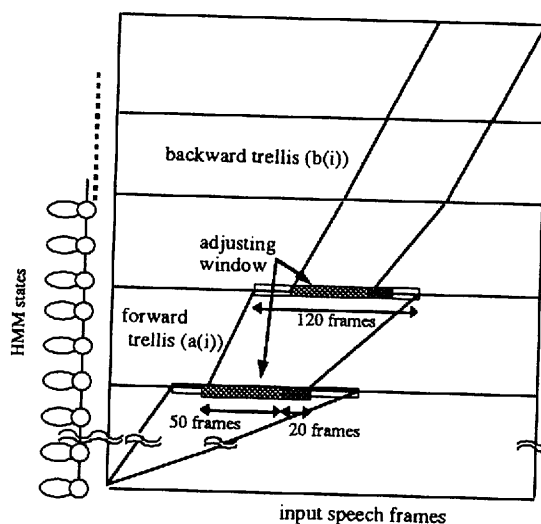


Figure 2. Search algorithm for the continuous speech recognition system.

words; only the sub-grammars include keywords. The meaning of a phoneme sequence candidate is extracted from keywords in the sub-grammars, with no consideration of the main grammar phonemes. The conventional HMM-LR algorithm works phoneme-synchronously. When using this process without modification, we cannot merge candidates that have different phoneme lengths, because candidates should always have the same phoneme length in this process. We therefore change the phoneme-synchronous search into a quasi-frame-synchronous one by making those candidates that have progressed farther wait for the others to catch up. This change keeps the adjusting window region of each candidate in nearly the same frame. Thus the candidates can be merged at almost the same frame. We change merging condition (c) (Sec. 2.3 (3)) in the search process to a new condition: merge candidates that have the same meaning, that is, those having the same keywords (address and name), ignoring the meaningless words (pauses, interjections, etc.). To merge candidates that are in nearly the same frame, we add new condition (d): the adjusting window positions of candidates are almost the same. These modifications raise the merge probability and improve recognition performance. Merging candidates at the meaning level is only summarized briefly here. We will describe this process in more detail in another paper.

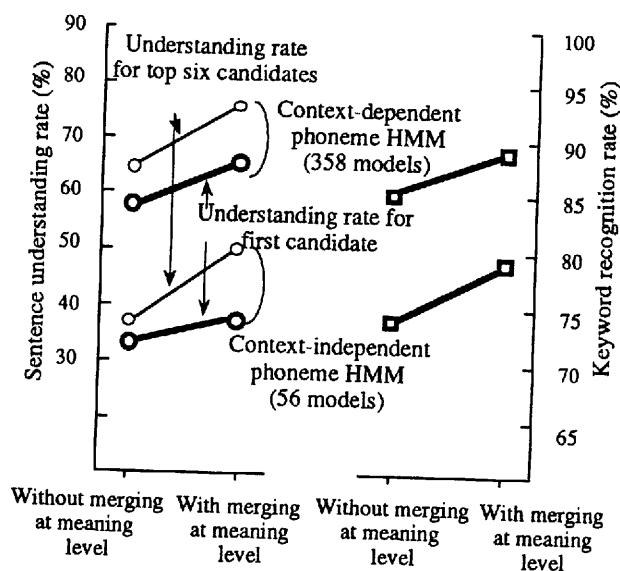


Figure 3. Sentence understanding and keyword recognition rates

2.4 RECOGNITION PERFORMANCE

We evaluated our algorithm by using spontaneous utterance mode. To handle spontaneous speech expressions, the grammar used here has various rules for interjections, verb phrases, particles, etc. It was created by analyzing 300 sentences in a simulated telephone directory assistance dialog. The test-set phoneme perplexity of this task was 4.9.

We performed experiments using context-independent HMMs and context-dependent HMMs for these utterances. We used 56 context-independent phoneme HMMs (15 vowel models, 28 consonant models, 10 diphthong models, and 3 silence/pause models) and 302 context-dependent phoneme HMMs. They had four states and three loops, and their feature vectors consisted of 16 cepstral coefficients, 16 delta cepstral coefficients, and delta power (33 dimensions in total). They were continuous Gaussian-mixture-density phoneme models with a diagonal covariance and with four distributions for each state. These phoneme HMMs were trained by concatenated training using 9600 phonetically balanced sentences uttered by 34 male and 30 female speakers. The beam width, the number of phoneme sequence candidates, was fixed at 2500. The weighting value $(1-\epsilon)$ for the backward trellis was 0.99.

We evaluated the proposed algorithm by using 51 spontaneous sentences containing 184 keywords. However they did not contain out-of-vocabulary words. Each sentence was uttered by eight different speakers.

The recognition experiments were done under four different conditions: (1) with a context-independent phoneme HMM, without merging at the meaning level; (2) with a context-dependent phoneme HMM, without merging at the meaning level; (3) with a context-independent phoneme HMM, with merging at the meaning level; and (4) with a context-dependent phoneme HMM, with merging at the meaning level.

The average sentence understanding and keyword recognition rates are shown in Figure 3.

The sentence understanding rate was 32% with a context-independent HMM and without merging at the meaning level. Using a context-dependent HMM without merging improved the average sentence understanding rate from 32% to 58%. Using merging, the average sentence understanding rate with a context-independent HMM was improved from 32% to 38%. These results confirm the effectiveness of a context-dependent HMM and merging at the meaning level. Finally, the recognition experiment was done using both a context-dependent HMM and merging. The average sentence understanding rate was 65% and the average keyword recognition rate was 89%.

3. MULTI-MODAL DIALOG SYSTEM

We designed our multi-modal speech dialog system for telephone directory assistance (Figure 4) with three input devices (microphone, keyboard, and mouse) and two output devices (loudspeaker and display). Figure 5 shows an example of the output window for the system. The numbers down the left side of the window show the rank of each candidate. For each of the five potential subscriber candidates, the system displays five slots: city, town, block number, subscriber name, and telephone number. A simple example of how this dialog system is used is as follows:

- (1) After clicking the speech button, the user states the address and subscriber name.
- (2) The system recognizes the input speech and displays the top five candidates.
- (3) If one of the candidates is correct, the user obtains the telephone number by clicking the telephone number slot of the correct candidate.

3.1 SPEECH RECOGNITION SYSTEM

Since response speed is an important factor in evalu-

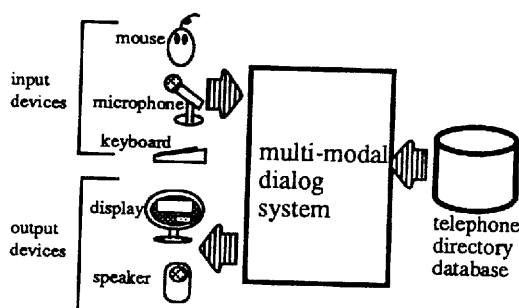


Figure 4. Basic structure of the multi-modal dialog system for telephone directory assistance.

	city	town	block number	subscriber name	telephone number
1	Mitakashi	Inogashira	4-22	Takahashi	
2	Mitakashi			Takahashi	
3	Mitakashi			Tanahashi	
4		Inogashira		Tanahashi	
5	Mitakashi			Tanashi	

reset keyboard speech cancel

message

recognition result: Mitakashi Inogashira yoN no nijuuni Takahashi saN

Figure 5. Sample of the output on a graphical display.

ating performance, we reduced the number of subscribers in our test system. The vocabulary contained 4000 words, and the beam width (the number of candidates kept in beam search) was reduced from 2500 to 200. We also changed the frame shift from 8 to 16 ms to improve response speed. Context-independent HMMs were used and there was no merging at the meaning level.

3.2 NOISE ADAPTATION

In a preliminary evaluation of our system, we could not obtain good performance because of environment noise, e.g., from computers and cars. We found that the HMM composition (compensation) method greatly improved the continuous speech recognition for a noisy environment (Martin et al., 1993). Under 18dB computer noise and for speaker dependent speech recognition, the HMM composition method showed almost the same performance as HMMs trained from noisy speech, while the HMMs trained from clean speech could hardly recognize sentences. Therefore, we used the HMM composition method for noise adaptation in the multi-modal dialog system.

In this section, we describe this method simply (See, in particular Gales and Young, 1992; Martin et al., 1993). The basic concept is to make HMMs for noisy conditions from speech HMMs and a noise HMM (Figure 6). Speech HMMs are modeled from noise-free speech and the noise HMM is modeled from environment noise. The main difficulty in doing this is calculating the output Gaussian distributions of output probabilities from the two source HMM distributions: the noise HMM and the speech HMM. Since each HMM is defined in the cepstrum domain, and speech and noise are additive in the linear spectrum domain, the Gaussian distributions defined in the cepstrum domain are transformed into log normal distributions in the linear spec-

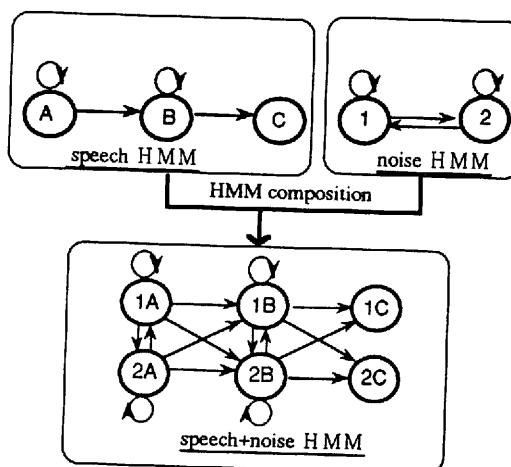


Figure 6. Example of the HMM composition process.

trum domain and convoluted and re-transformed. This method is formulated as follows.

Suppose that there is no correlation between the noise and speech signals. Since the noise signal and speech signal are additive in the linear domain, we get the equation:

$$R_{cp} = \Gamma^{-1} \log(e^{\Gamma S_{cp}} + e^{\Gamma N_{cp}}), \quad (3)$$

where R_{cp} is the noisy speech cepstrum, Γ is the cosine transform, S_{cp} is the speech cepstrum, N_{cp} is the noise cepstrum, e is the exponential transform, and $\log$ is the logarithmic transform. Now, we want to find the distribution of R_{cp} , when distributions of S_{cp} and N_{cp} are defined as Gaussian distributions. Since the cosine transform is a linear transform, the distribution after transformation is still a Gaussian distribution; distributions of $S_{lg} = \Gamma S_{cp}$ and $N_{lg} = \Gamma N_{cp}$ become Gaussian distributions. Thus the mean and covariance values of the distributions after cosine transformation are easily calculated. Since after the exponential transformation, the Gaussian distribution becomes a log normal distribution, the distributions of $N_{ln} = e^{\Gamma S_{cp}}$ and $N_{ln} = e^{\Gamma N_{cp}}$ are log normal distributions. Since $(e^{\Gamma S_{cp}} + e^{\Gamma N_{cp}})$ is the sum of $e^{\Gamma S_{cp}}$ and $e^{\Gamma N_{cp}}$, its distribution is obtained by convoluting the distributions of $e^{\Gamma S_{cp}}$ and $e^{\Gamma N_{cp}}$.

It is difficult, however, to obtain a real distribution for $(e^{\Gamma S_{cp}} + e^{\Gamma N_{cp}})$. If we assume that $(e^{\Gamma S_{cp}} + e^{\Gamma N_{cp}})$ can be approximated to a log normal distribution, the distribution of $\Gamma R_{cp} = \log(e^{\Gamma S_{cp}} + e^{\Gamma N_{cp}})$ is a Gaussian distribution and its mean and covariance can easily be obtained from the means and covariances of the distributions of $(e^{\Gamma S_{cp}}$ and $e^{\Gamma N_{cp}})$ (See Gales and Young, 1992; Martin et al., 1993). Finally, since Γ^{-1} is also a linear transform, the distribution of R_{cp} can be obtained as a Gaussian distribution. These process are shown in Figure 7.

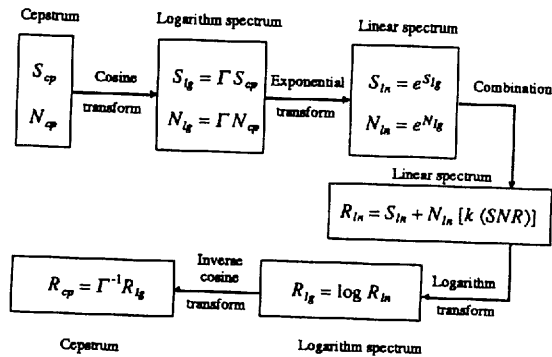


Figure 7. HMM composition process for output probability.

Our previous experiments showed that the computation needed for HMM composition was significantly smaller than for training speaker-independent HMMs using noisy speech, while recognition accuracy was comparable (Martin et al., 1993).

3.3 DIALOG CONTROLLER

The main functions of the dialog controller are as follows.

(1) DISPLAYING CANDIDATES

After speech recognition, four potential candidates are displayed in order of their likelihood scores. We generally do not use the directory assistance database constraint to select these candidates, as described in 2.1.. As the fifth candidate, however, we use the candidate which satisfies the constraint in the telephone directory assistance database, because there is a high possibility that the candidate that satisfies the constraint is correct, even if it has a low likelihood score.

(2) CORRECTING ERRORS

If there is no correct candidate among the five candidates, the user corrects the input error by choosing a candidate close to the correct subscriber address and name, clicking the slot with the wrong keyword, and uttering a sentence with that specific semantic item. In error correction, the speech recognition system uses a grammar adjusted for each slot. For example, if the user selects the subscriber name slot, the system switches to the grammar for utterances containing a subscriber name.

4. EVALUATION OF DIALOG SYSTEM

We asked 20 users in our laboratories to try to use this system. Each user was given a randomly selected sheet of paper indicating the task using an annotated map. Each user completed 20 tasks, in total.

Dialog experiments were performed to evaluate the following issues.

1. System performance,
2. User evaluation of the system,
3. Content and manner of user utterances, and
4. Problems encountered with the system.

Training

The users were first requested to practice operating this system by themselves using a tutorial system, which was an interactive system implemented on a workstation. The tutorial system was designed to give the same amount of information to each user. The training took about 10 minutes on average. Users were presented three example queries as a means of implicitly teaching them the general structure of a query. The examples were presented as text displayed on the workstation and the users requested to practice by uttering these examples three times. Speaking rate, style, and pauses were not controlled and any feedback on this aspect was

provided.

Experiments

After the training, the users made twenty attempts at getting telephone numbers using information indicated on twenty simplified city maps. Subscriber name and address were indicated on each map. Figure 8 shows a sample. The maps were used to reduce restrictions that would be implicitly imposed by written sentences implicitly. If instructions containing names and addresses were written out, some users might simply read them or it might influence their choice of words.

Questionnaires

After the experiments, users were requested to answer several questions on a scale from 1 to 5. They were also asked to write their overall impression of the system.

Results

Table 1 shows the experimental results. The task completion rate which means users could get the correct telephone number, was 99.0%. The average sentence understanding rate was 57.8%, while the average rate that the correct item was in the top five candidates was 77.5%. This shows that in three of four trials, the system

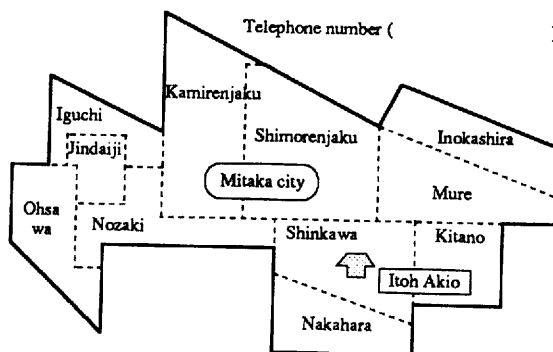


Figure 8. Example of the sheet indicating a directory inquiry task.

Table 1. Experimental results.

Task completion rate	99.0%
Sentence understanding rate	57.8%
Sentence understanding rate for top five candidates	77.5%
Number of utterances	1.40
Number of pointings with mouse	2.88
Task completion time	57.2sec.

displayed the correct telephone number after the users first utterance. The average number of utterances was 1.40, and the average number of pointing actions with the mouse was 2.88. At least, one utterance and two pointing actions were needed to get a telephone number. The results show only a small increase in the average number of utterances and pointing were needed for the users to complete the task, so indicating the multi-candidate approach was effective in reducing the number of operations.

The average sentence understanding rates for the top five candidates are plotted in Figure 9 as a function of the average number of utterances for each user. The higher the average sentence understanding rate a user had, the fewer the average number of utterances that were made. Even when the sentence understanding rate was low, roughly 60%, a user could get the correct

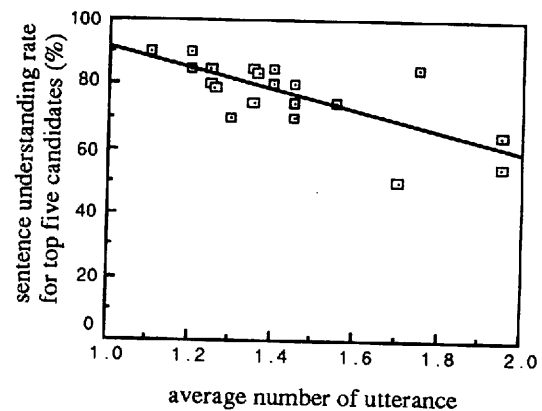


Figure 9. Sentence understanding rate for top five candidates versus the average number of utterances

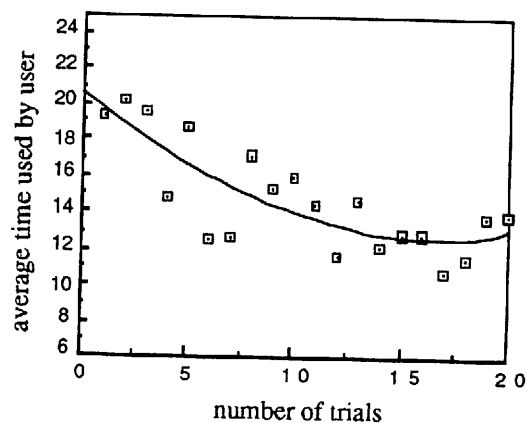


Figure 10. time used by user versus the number of trials.

telephone number in two utterances on average.

Figure 10 shows the relation between the number of trials and the average 'time used' by users, where time used means the time needed for users to make decisions or operate the system. As shown in the figure, user time decreased as the number of trials increased. This shows the users adapted well to operating the dialogue system. However, adaptation of the speaking manner to improve the speech recognition rate or speech recognition time was not found.

Preference for This System

Two of the questions were:

1) When you want to get a phone number, which would you prefer to use, this system or a telephone book?

2) Do you feel this system is easy to use?

The answer spread shows that 75% of the users preferred using the system to a telephone book and 55% of them felt it was easy to use. More than half of the users thought the system performed well, so we have come close to achieving our goal of designing a dialogue system that users feel comfortable with.

5. CONCLUSION

We proposed a very-large-vocabulary continuous speech recognition algorithm and applied it to a telephone directory assistance system. It has an accurate and efficient algorithm that uses a two-stage LR parser with phoneme HMMs. To evaluate our speech recognition algorithm, we implemented a multi-modal dialog system. We evaluated it from the human-machine-interface point of view. Although experimental results show that the smaller-scale system containing 2,300 subscribers works very well, we still need to improve the performance of the system; in particular, to speed up the processing time.

6. ACKNOWLEDGMENTS

We thank the members of Furui Research Laboratory of the NTT Human Interface Laboratories for their useful discussions. We used the Acoustical Society of Japan speaker-independent continuous speech database to train the speaker-independent HMMs.

REFERENCES

- P. Kenny, R. Hollan, G. Boulianne, H. Garudadri, Y.-M. Cheng, M. Lennig, and D. O'Shaughnessy (1992), "Experiments in continuous speech recognition with a 60,000 word vocabulary", *Proc. Internat. Conf. Spoken Language Process.*, Banff, October 1992, pp. 225-228.
- H. Ney, R. Haeb-Umbach, B.-H. Tran, M. Oerder (1992), "Improvements in beam search for 10,000-word continuous speech recognition", *Proc. Internat. Conf. Acoust. Speech Signal Process.*, San Francisco, March 1992, pp. I-9 to I-12.
- L. Hirschman, M. Bates, D. Dahl, W. Fisher, J. Garofolo, D. Pallett, K. Hunnicke-Smith, P. Price, A. Rundnick, and E. Tzoukermann (1993), "Multi-site data collection and evaluation in spoken language understanding", *ARPA Workshop on Human Language Technology*, March 1993.
- K. Kita, K. Kawabata, and H. Saito (1989), "HMM Continuous Speech Recognition Using Predictive LR Parsing," *Proc. Internat. Conf. Acoust. Speech Signal Process.*, Glasgow, May 1989, pp. 703-706.
- M. Tomita (1986), *Efficient Parsing for Natural Language: A Fast Algorithm for Practical Systems* (Kluwer Academic Publishers).
- S. Austin, R. Schwartz, and P. Placeway (1991), "The Forward-Backward Search Algorithm," *Proc. Internat. Conf. Acoust. Speech Signal Process.*, Toronto, May 1991, pp. 697-700.
- F. K. Soong and E.-F. Huang (1991), "A tree-trellis based fast search for finding N best sentence hypotheses in continuous speech recognition," *Proc. Internat. Conf. Acoust. Speech Signal Process.*, Toronto, May 1991, pp. 705-708.
- E. Shriberg, E. Wade, and P. Price (1992), "Human-machine problem solving using spoken language systems (SLS): Factors affecting performance and user satisfaction", *Proc. Speech and Natural Language Workshop*, February 1992, pp. 49-54.
- Y. Takebayashi, H. Tsuboi, Y. Sadamoto, H. Hashimoto, and H. Shintchi (1992), "A Real-Time Speech Dialog System Using Spontaneous Speech Understanding", *Proc. Internat. Conf. Spoken Language Process.*, Banff, October 1992, pp. 651-654.
- Y. Minami, T. Matsuoka, and K. Shikano (1992), "Very large vocabulary speech recognition algorithm for telephone directory assistance", *1st IEEE Workshop on Interactive Voice Technology for Telecommunications Applications*, October 1992.
- F. Martin, K. Shikano, Y. Minami (1993), "Recognition of noisy speech by composition of hidden Markov models", *Proc., Eurospeech*, Berlin, September 1993, pp. 1031-1034.
- A. P. Varga and R. K. Moore (1990), "Hidden Markov model decomposition of speech and noise", *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Albuquerque, April 1990, pp. 845-848.
- M. J. F. Gales and S. J. Young (1992), "An improved approach to the hidden Markov model decomposition of speech and noise", *Proc. Internat. Conf. Acoust. Speech Signal Process.*, San Francisco, March 1992, pp. 233-236.
- Y. Minami, K. Shikano, S. Takahashi, T. Yamada, and O. Yoshioka (1993), "Large vocabulary speech recognition system for telephone directory assistance", *Proc. International Symposium on Spoken Dialog*, Tokyo, November 1993, pp. 169-172.