A MACHINE UNDERSTANDING SYSTEM FOR SPOKEN JAPANESE SENTENCES

SEI-ICHI NAKAGAWA

DEPARTMENT OF INFORMATION SCIENCE
KYOTO UNIVERSITY

OCTOBER 1976

A MACHINE UNDERSTANDING SYSTEM FOR SPOKEN JAPANESE SENTENCES

Sei-ichi NAKAGAWA

October 1976

Department of Information Science

Kyoto University

Kyoto, Japan

Submitted to the Faculty of Engineering
of Kyoto University
in partial fulfillment of the requirements
for the degree of Doctor of Engineering

A MACHINE UNDERSTANDING SYSTEM FOR SPOKEN JAPANESE SENTENCES

Sei-ichi NAKAGAWA

Kyoto University

The more progress information science makes, the more has our daily life been benefited by computer applications, such as seat reservations for airplane or train, and computing service by pushphone. If we could operate it as easily as we drive a car, an age of information science will truly come. It is, therefore, desired that we be able to converse with a computer by speech which is the medium of human communication. Man-machine communication using speech cannot only offer increased physical mobility (omnidirectional) to the computer user, but it can also enable us to use our hands and eyes for other activities while talking or listening to the computer.

It would be one of our most cherished desires in information science to develop a speech understanding system. The final goal of this research is to develop a system capable of recognizing connected speech, from any speaker, in close to real time. Further, this system should be generalized so as to operate easily in any different task domain.

This thesis represents an attempt to build a system capable of understanding messages spoken in simple Japanese sentences. We have developed the LITHAN (LIsten-THink-ANswer) speech understanding system. By using higher linguistic information such as syntax, semantics, and pragmatics, this system can automatically recognize continuously uttered speech. LITHAN has been designed to save computer storage and amount of computation, and to correct several kinds of errors occurring at each of the linguistic levels. The included components are an Acoustic Processor, Phoneme Recognizer, Word Identifier, Word Predictor, Parsing Director and Responder.

The Acoustic Processor has a 20-channel 1/4-octave filter-bank spectral analyzer. The Phoneme Recognizer encodes a time series of the spectrum into a phoneme string. The Word Identifier verifies a predicted word in the above phoneme string. The Word Predictor predicts plausible words in the unknown portion of the input sentences from a grammatical standing point. The Parsing Director parses an input sentence while activating the Word Identifier and Word Predictor. The Responder(under development) tries to understand the meaning of the recognized sentence, and it answers a query or acts according to the input command. The low level of the system consists of a hierarchical model, the high level heterarchical one.

LITHAN uses many types of *a priori* information; statistics, word dictionary, spoken grammar with additional information, and semantic and pragmatic information. Utilizing linguistic information, it predicts words which might possibly appear at the unrecognized portion of input utterances. It then verifies these words by using the optimum matching algorithm between a recognized phoneme string and each entry in the word dictionary. LITHAN parses sentences by a new tree search method which possesses the benefits of both the best-first search and the depth-first search. It works well to compensate for inaccurate results in phoneme recognition and word verification.

The LITHAN system can be applied to a new task with a minimum effort, simply by preparing a grammar and word dictionary. The task we have applied this efficient system to is operational commands or inquiries about the status of a computer network (The size of this vocabulary is about 100.), and a typical utterance is composed of 4 to 24 words. The system has been tested on a sample of 200 sentences spoken at a normal speed by 10 male adults. 64% of the sentences and 93% of the output words have been recognized correctly.

We have also applied LITHAN to sentences from other tasks which have different linguistic information, such as 'Arithmetic Expression' and 'Calendar'. We have proved through these experiments that the contribution of linguistic information to the recognition rate is

very large.

As a subpart, this system has a spoken word recognition system with a limited vocabulary. This sub-system has correctly recognized 97% of the digits isolatedly spoken by 20 male speakers, and 98% of those by 10 adaptive speakers (system training).

I hope that this thesis creates and promotes the reader's interest in speech understanding research. I gratefully welcome any comment and suggestion in regard to this thesis.

ACKNOWLEDGEMENTS

I would like to express my grateful appreciation to Professor Toshiyuki Sakai, who has been a most supportive advisor. His ideas concerning this study and his suggestions for this thesis have been equally valuable.

I wish to thank Dr. Kenji Ohtani who guided my research in speech understanding. I am also grateful to Associate Professors Shigeharu Sugita (National Museum of Ethnology) and Takeo Kanade for their encouragement and discussions during the course of this study. Also, I am indebted to Associate Professor Yasuhisa Niimi (Kyoto Technical University) for having initially motivated this area of research and for his continuing advice.

Special acknowledgement and thanks belong to Mr. Hiroshi Sakimura and Naoharu Yoshitani for their close co-operative help in designing and implementing the LITHAN system. Thanks also to Mr. Kimio Maegawa, a strong supporter who constructed the hardware and software for speech processing.

I also wish to thank various people who helped me during the course of this work, including Mr. Teruhiko Ukita, Syukichi Hayashi, and my colleagues in Prof. Sakai's research group.

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION

Why would one want to talk with a computer in natural language? Why should one continue at the work of speech recognition in spite of Pierce's critics (Pierce[69]) ?

In order to answer these questions, I should explain the advantages of speech communication and the significance of speech recognition research.

Of all communication media between man and machine, speech is the best one that can presently be thought of, when viewed from the following considerations:

(1) Man is capable of making any abrupt interactions to a machine by his own speech at any time under any circumstances, i.e., in the dark or even when eyes, hands and feet are occupied for other activities.

(2) For speech communication, man does not need any tools or specialized training, it is rather the most natural channel of human communication.

(3) Speech has probably the most universal output modality of communication and one of the man's fastest modalities.

From the viewpoint of practical applications, I might add the following: An economical and rapid transmission of information between machines and a large number of people would be achieved very effectively by voice transmission over a popular telephone network.

From the standpoint of researchers, there are two primary reasons for a strong interest in machine recognition of speech (Otten [67]):

(1) Interest in translating human speech into a machine code as basis for man-machine communication (practical needs).

(2) Interest in an understanding of the mechanism of human speech production and perception process (scientific curiosity).

Furthermore, for conversational speech, (3) there is a strong interest in an understanding of natural languages as communication code (scientific curiosity). Such understanding will be also valuable for practical need. In this thesis, I describe the studies motivated by mainly the first two interests.

I-1 Speech Understanding System - Philosophies

If possible, man-machine communication by speech would be very efficient and convenient, as described before. Many researchers have studied automatic speech recognition by machine (see a survey by Hyde [29]). From the results of their works, we have found that, it is very difficult even to recognize ten spoken digits in isolation, as long as it is desired to work for unspecific speakers. The task becomes even more difficult for continuous or conversational speech.

Attempts to overcome this difficulty have led us to the idea that linguistic information or knowledge as well as acoustic information is required to construct an automatic speech recognizer. Denes([14], 1959) was the first to work on this idea. His recognizer used information about all possible diagram of sequential probabilities relevant to the speech material to be tackled by the recognizer. A similar attempt was made by Sakai and Doshita([76],1963). Flanagan ([18],1965) suggested that automatic speech recognition would probably be possible only through the proper analysis and application of grammatical, contextual, and semantic constraints. Alter([2],1968) introduced the use of word dictionary and contextual constraints in speech recognition for 'Spoken Fortran' language. Vicens and Reddy (Vicens[95],1969) organized a system which could recognize continuous speech in a highly constrained natural language; this was the first

system that took into account information from all levels.

In 1970, Fant[16] proposed a new model of a powerful speech recognition system which contained acoustical, phonological, syntactic, and also semantic speech processing stages. The remarkable feature of this model was that it introduced feedback paths for reexamination from a stage of higher level to a lower one.

Through such examinations, we met the ARPA Speech Understanding Project (Newell, et al.[62],1971). The Study Group has grasped that the research of speech understanding is to understand the message, not to recognize phonemes, words or sentences.

This project was based on the progress made in isolated word recognition and in other aspects of artificial intelligence. The final goal of speech understanding research is to develop a system capable of recognizing connected speech, from any speaker, in close to real time. Further, this system should be easily generalizable to operate in any different task domain.

If we could indeed construct a successful speech understanding system (SUS), this research would give us the scientific significance as well as practical applications.

An SUS uses as many various kinds of information as possible: acoustic, parametric, phonemic, syntactic, semantic, and pragmatic. For the purpose of efficient utilization of linguistic information, the input should be sentences which describe only a restricted world. This world is called a 'task'. In machine processing of natural language, a specialized world, processed by a system, allows us to easily create a model of a concept of that world.

To construct a successful SUS, we should solve the seven major problems which arise in the following levels:

1. *Acoustic Parametric level* ..... Extraction of suitable acoustic features.
2. *Phonemic level* ..... Normalization of coarticulation and differences among speakers.

3. *Lexical level* ..... Translation of a recognized phoneme string into a word in the lexicon, and establishment of phonological rules.
4. *Syntactic level* ..... Establishment of a spoken sentence grammar.
5. *Semantic level* ..... Representation of meaning for each word or sentence.
6. *Sentence level* ..... Understanding of natural language.
7. *System level* ..... System organization which includes control of errors.

I-2 Aspects of Speech Understanding Research

The two main aspects of speech understanding research include what model we employ on a system design, and how matching and searching procedures are used. First let us have some views on models or procedures used in other SUS's.

Klatt and Stevens[38] attempted to recognize a set of unknown sentences by visual examination of spectrograms and by machine-aided lexical searching. When human experts attempted to decipher spectrograms, the initial transcriptions in terms of phonetic features contained many errors and omissions. However, 96% were successful in identifying the words of the utterances when permitted to make use of their knowledge of the vocabulary, syntactic and semantic constraints.

The HEARSAY I system(Reddy et al.[74], Neely[59], Erman[15]) employed a hypothesis-and-test paradigm for speech recognition. In this system, each type of knowledge source (acoustic, syntactic, semantic) was represented as a model operating somewhat independently from others. This independence brought a clean structure that allowed the contributions of each source to be evaluated. The search strategy was basically a best-first tree search. For word prediction, words were hypothesized proceeding left-to-right in the utterance.

The HEARSAY II system(Lesser et al.[47]) have been designed to

handle more complex speech task domains, e.g., larger vocabularies, less restricted grammars, and a more complete set of knowledge sources including prosodics, user models, etc. The two main features of this system are first the representation of a partial analysis in a generalized 3-dimensional network ("information levels", "speech time", and "alternative hypotheses"), and second a convenient modular structure for incorporating new knowledge into the system at any level. But I feel that this plan might be beyond the current arts in speech recognition and artificial intelligence.

The DRAGON system(Baker[3],[4]) models knowledge sources as probability functions of the Markov process. The system finds a well-formed sequence on related observations which maximizes the *a posteriori* probability. A matching procedure is needed to evaluate each particular hypothesis, and a searching procedure is needed to explore the space of possible hypotheses. However, DRAGON consumes much computational time and also cannot easily treat semantic knowledge.

Miller[51] developed an approach different from traditional parsing methods in that it has no inherent left-to-right or right-to-left, but allows syntactic structures to be recognized locally in any part of the sentence. This system(LPARS) was, however, implemented in "scrambled input", which simulated inputs obtained from a temporary phoneme recognizer.

The CASPERS system(Klovstad and Mondshein[40]) performs a match between lexical items and a noisy phoneme sequence by using multiple dictionary entries, therefore phonological rules are embedded in the dictionary. The search is controlled by an augmented context-free grammar which performs a left-to-right, bottom-to-up parse.

In the SPEECHLIS system(Woods[100], Bates[8], Nash-Webber[58]), the most remarkable feature is that all alternatives at any point in time are shown by lattice structures (i.e., segment lattice and word lattice). However, our experience tells us that with a word lattice it is very difficult to consider the contextual influence between words.

The SRI speech understanding system(Walker[97], Becker and Poza

[9], Paxton[68]) uses a 'word function' to make a detailed examination of acoustic data. Just as in HEARSAY I, the parser employs a top-down, best-first strategy. In this strategy, the priority value of a word is affected by several sources, such as the grammar, world model, and results of the word function test. It seems to me that the only features the word function searches in detail are the positive ones on the acoustic parameters, further, they must be prepared after a detailed examination of acoustic data.

The VDMS(Barnett[6],[7], Ritea[75]) developed at SDC employs a strategy of predictive linguistic constraints. The main feature of word identification is that very loose matching criteria are used for locating the predicted words. First, special attention is given to the word segments experimentally determined to be most invariant in the word. Finally, this matching procedure is carried out on a syllable-by-syllable basis, applying phonological rules. This linguistic processor is capable of handling both left-to-right and right-to-left parsing.

At the IBM Watson Research Center(Jelinek, Bahl, and Mercer[35]), a linguistic sequential decoder is employed for recognition of continuous speech. As in the DRAGON system, the search procedure is a stack decoding algorithm which seeks a word sequence having the maximum value of *a posteriori* probability. Statistical matching is done between a phonemic lexicon of a hypothesized word and a noisy phoneme sequence obtained by acoustical analyses.

Sperry Univac(Lea[45], Lea, Medress, and Skinner[46]) is continuing its implementation and testing of a strategy of speech recognition. In this strategy, prosodic features are both to segment the speech into grammatical phrases and to identify the stressed syllables in the sentence structure.

In American speech understanding research, ARPA network has been powerfully used for resource sharing of programs or data bases, e.g., CMU-Xerox(Neely and White[60]), Univac-BBN, Univac-SDC(Lea[45]).

In Italy, Mori et al.[52],[53] have investigated the problem of

emitting and verifying hypotheses by using a stochastic grammar for speech; this grammar has a set of productions for each coarticulation segment at the syllabic level. Particularly interesting to me is the automatic extraction and description of acoustic data on the basis of distinctive features, prosodic features, and so on. However, I doubt if such a description could be given.

In France, Haton and Pierrel[26] have developed a system which is organized around a syntactic analyzer and incorporates semantics in an interactive dialog with the user in order to avoid some of the errors and ambiguities. The system was applied to a task with a limited grammar and a small vocabulary.

In Japan, some notable speech recognition (understanding) systems have been developed.

Shirai and Fujisawa(Waseda Univ.[89]) developed the first recognition system of Japanese, spoken separately. The sentence structures were defined according to the ordering of the parts of speech and the problem of sentence recognition was formulated as an optimization with the constraints of the sentence structure. This approach, I think, would be applied only to very limited sentences.

The process of the speech recognition system developed at Musashino Electrical Communication Laboratory(Kohda et al.[41]) consists of two stages: acoustic and linguistic processing. In the former, the syllables of input speech (inserting a pause at every phrase) are analyzed one by one against the acoustic phonetic data on a VCV syllable-by-VCV syllable basis. In the latter, the main features are the use of phoneme transcribing rules and depth-first search. The final goal is to construct a question-answering system by speech.

Niimi et al.(Kyoto Technical Univ.[64]) have developed a voice-input programming system, 'SPOKEN-BASIC I'. In the system evaluation, they examined the effects of two typical tree search methods: the depth-first method and the breadth-first method.

Takeya and Tamachi(Kyushu Univ.[93]) have proposed a unique

parsing method which uses a top-down and breadth-first strategy. The syntactic structures are described by a dependency grammar. This parser can treat unknown words but accepts some meaningless sentences. The performance of phoneme and word recognition is rather poor.

Sekiguchi et al.(Yamanashi Univ.[87]) have developed a speech recognition system of 'Fortran' uttered with a long pause between successive words. They have aimed to develop a powerful phoneme recognizer.

The LITHAN speech understanding system developed by the authors (Sakai and Nakagawa[77],[78]) recognizes continuously uttered speech in simple Japanese sentences. This system will be outlined in the following section.

I-3 LITHAN - System Overview -

This section gives an overview of the LITHAN system, in particular, philosophy of the system design and the criterion of task selection.

I-3-1 System Specifications

Is it possible to understand speech by computer? To answer this question, detailed specifications of the speech understanding system are required. These specifications also determine the type, difficulty, and techniques for solving problems involved in implementing the system. An example of such specifications has been drawn up in a final report by the study group of ARPA(Newell et al.[62]); hopefully, this will provide a reasonable chance of success for the system in five years. However, I fear that a period of five years (whereby a speech understanding system should be demonstrable in 1976;) is not long enough to achieve such a goal.

We have been studying speech understanding for the past three and a half years (i.e. from the spring of 1973). In comparison with that of the ARPA Projects, our starting time is behind by about one and a half years. But on the other hand, the works done by ARPA in this period have benefited us and led us to some discussions:
(1) Just as Fant[17] cited as one of the aspects of American study, almost all groups of speech understanding researchers have gone ahead and particularly developed syntactical and semantic analysis. I want to affirm the fact that we could not make a successful speech understanding system without first constructing a powerful phoneme or spoken word recognizer.
(2) We learned that we could avoid the same mis-strategies as they did, by examining the strong and weak points of those foregoing speech understanding systems, for example, in word identification, tree searching, and parsing flow.

From these points of view, we set up the following specifications for building our speech understanding system. Of course, we considered the aims of practical and scientific research.

1. *Multiple speakers* ..... If an SUS accepts only a single speaker, it means that in practical use this speaker has the system to himself. Thus, input speech may be uttered in a sequence of isolated words or meaningful symbols, since by training he is able to memorize an order of utterance. In this case, excluding scientific interests, speech understanding research offers little practical advantage to us.

2. *Continuous speech* ..... If it is necessary to speak separately word-by-word, we cannot make use of the two advantages of natural language and voice input. By first advantage we can immediately constitute a message in the brain, and through the second we can represent this message fluently and speedily.

3. *Phonemes as a recognition unit* ..... As the task vocabulary size becomes larger, it becomes necessary that the system be able to identify more words. If each word has to be matched one by one

against the acoustic phonetic data, this process will consume a large amount of computational time and memory storage. Therefore, as a unit of recognition, we have to consider smaller units such as phonemes, syllables, and VCV syllables. In order to normalize differences among speakers, we need some normalizing or learning procedure. Furthermore, this procedure must be a fast learning algorithm, because the users of an SUS may often be taking turns in a short period. Speech recognition on the basis of the phonemic unit has the following merits: (a) quick learning, (b) small computer memory, (c) short processing time and (d) easy introduction of various linguistic information. This approach is the scientific process, and it also presents other aspects, such as an interesting difference between observed acoustic and pre-registrated data.

4. *Word identification* ..... To develop a successful SUS, we need to construct powerful subsystems: an acoustic feature extractor, a phoneme recognizer, and a word identifier. The performance of this word identifier depends on that of the two other subsystems, and has also drastic influence on the performance of the total system. Therefore, we desire a powerful word identifier such that when recognizing ten digits in isolation the error rate will be less than 5% for unknown speakers.

5. *Tree searching* ..... Because tree searching techniques developed in artificial intelligence research depend on procedures for problem solving, they are not suitable for SUS's. A search procedure must seek a word string which has the maximum likelihood among all plausible strings. For this logical reason, we must consider a new tree searching method suitable for speech input, as this is quite different from text input.

6. *Flexible system* ..... As long as our intent is fundamental research of an SUS for future development, the system must be flexible one which enables us to evaluate the contribution of syntax, semantics, pragmatics, and other sources of knowledge towards recognition by

accepting various tasks. Such an SUS can give us an estimation as to what kind of knowledge and how much each kind of knowledge contributes towards improving the preformance of the system.

7. *Speech material and system evaluation* ..... In almost all systems developed so far, performance evaluations have been done on total systems, but not on individual components except HEARSAY I(Erman[15]). And these evaluations have only been based on small speech material for one or a few speakers. I feel that little interesting scientific knowledge can be obtained from such results. Therefore, we should evaluate the performance of individual components as well as that of a total system with much speech material.

I-3-2 Task Selection

When we construct an SUS, we should first select a task and define its domain. *What criteria should prospective tasks satisfy?* The criteria, of course, depend on the state of the current art of automatic speech recognition and the aim of the work. We have specified the following criteria for a task selection:

1. The input is continuous speech in simple Japanese sentences.
2. The vocabulary size is in the range of 100 to 200 words.
3. The system accepts natural language with slightly restricted syntax.
4. The system can use semantic and pragmatic information.

Considering the advantages and disadvantages of speech input to computer, the following should be added to the criteria:

5. We can use the system by speech input while interacting with a computer.....Even if a successful SUS is constructed, we must always realize that the system may make a mistake.
6. The system should be used by time sharing.....An SUS cannot occupy a large computer by itself, from the view point of cost.
7. We can utter an input sentence without making any previous notes. ....This input should not include any random sequence of numerals or

symbols which man could not memorize quickly.

According to these criteria, we selected the task concerned with 'Computer Network', of which the vocabulary size is about 100 words used to give an operational command and inquire about the status of a computer network. In defining the task, we considered the following:

1. The types of input sentences are either affirmative, imperative or interrogative.
2. An input sentence may contain adjectives.
3. The vocabulary should contain digits.....Ten digits compose a set of the most fundamental words for man-machine communications.

We selected two other tasks, 'Arithmetic Expression' (such as 11+2×3=) and 'Calendar'(such as Wednesday, January 1). Although these tasks do not satisfy the criteria mentioned above, they do have very definite syntax ('Calendar' also has distinct pragmatic information), and they give a fundamental knowledge in regard to the contributions of individual components (see Chapter VI).

We further selected ten digits as a task, where the performance of the lower levels i.e., Acoustic Processor, Phoneme Recognizer and Word Identifier, was tested on isolated word recognition.

I-3-3 System Organization

Fig. I-1 indicates a block diagram of the LITHAN (LIsten-THink-ANswer) speech understanding system. LITHAN is composed of three parts: LISTEN(LImited Spoken Text ENcoder), THINK(THought by INternal Knowledge) and ANSWER(ANnouncement by Saying/Writing to receivER). Viewed from a different standpoint the system consists of six components: Acoustic Processor, Phoneme Recognizer, Word Identifier, Word Predictor, Parsing Director, and Responder.

Fig. I-2 illustrates the knowledge and processing flows in LITHAN. Knowledge flow is based upon the data base (or blackboard, Reddy and

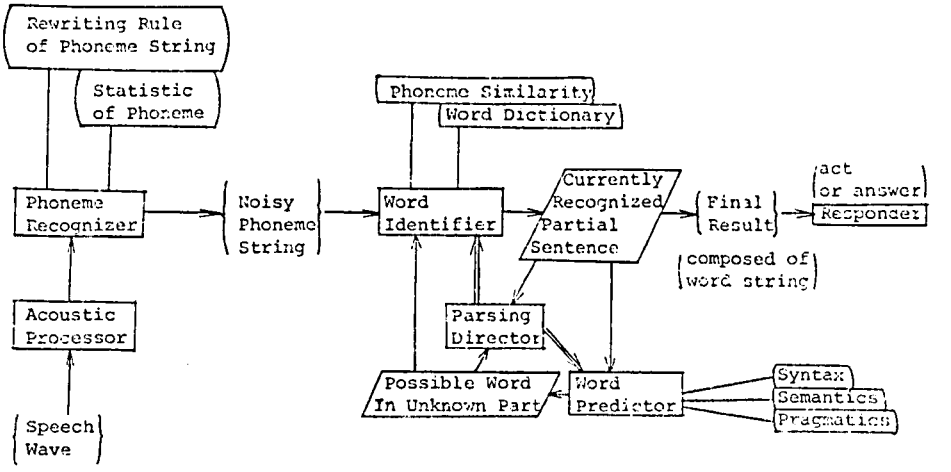


Fig. I-1 Configuration of Speech Understanding System LITHAN

$\longrightarrow$ data flow, $\Longrightarrow$ control flow, —— knowledge flow

Erman[73]) model, and processing flow upon the hierarchical model. These models are adopted by the following:

1. One of the aims of speech understanding research is to combine pattern recognition research for phoneme recognition with the research on artificial intelligence for natural language processing. The interaction between them will be performed by the similarity matrix $S(i,j)$ in the Word Identifier, where 'i' is a phoneme in a language level and 'j' a phoneme in an acoustic level. This matrix is used for evaluating the translation from a recognized phoneme string into a word. Thus, an SUS is organized by a hierarchical model where an acoustic (physical) level and a language level are linked by the Word Identifier. In the language level, then, the Word Identifier and the Word Predictor can be constructed independently under control of the Parsing Director, which directs the parsing procedure.

2. Speech understanding research has many difficult problems. It is necessary that it be tackled by a number of cooperating researchers. Thus, it is desired that all components be so independent of each other that this system may be constructed in parallel by different researchers.

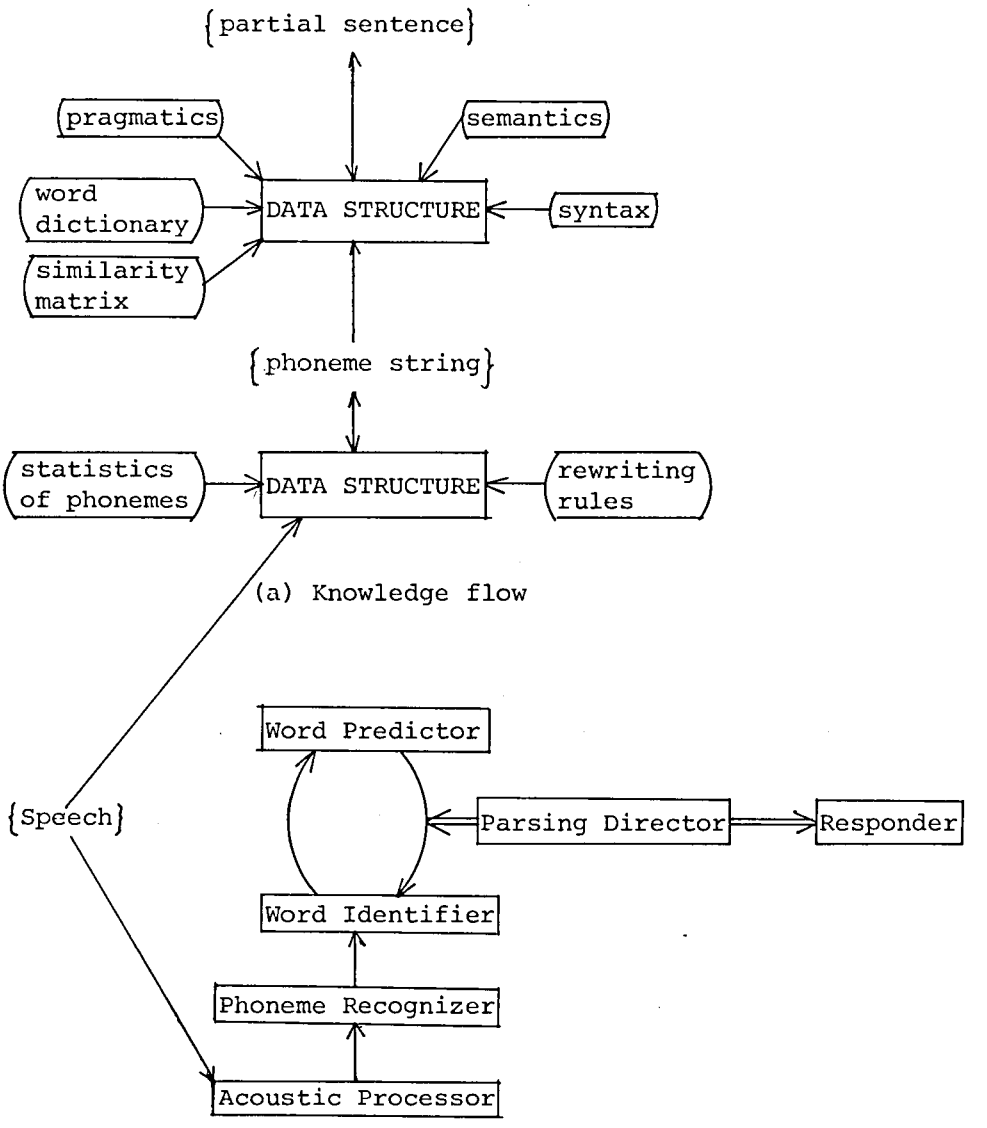


Fig. I-2 Macro-representation of the LITHAN system

3. If we accept the published reports that speech perception occurs at the level of the phoneme (Lieberman et al.,[48]) or that the CV syllable is a perceptual unit in speech (Massaro[50]), we should prefer the bottom-up procedure at the lower levels.

4. In the course of developing of a fundamental SUS, the contribution of each component to speech recognition should be evaluated for various tasks.

The Acoustic Processor (described in Chapter II) consists mainly of a 20-channel filter-bank spectral analyzer. A speech signal is passed into a pre-emphasis circuit with a slope of 6-dB per octave below 1600Hz, and then fed into the filter-bank. The outputs are rectified, smoothed, and sampled at every 10ms interval, thus yielding a series of a short time spectrum of 20-dimensional spectra. This series is then converted into a phoneme string, utilizing statistics of phonemes and rewriting rules. Each segment consists of a 4-tuple: the phonemic symbol for the first and second candidates, the degree of confidence (reliability) for the first candidate, and the segment duration.

The Word Identifier (described in Chapter III) verifies predicted words from a given part of the phoneme string and calculates their likelihoods. It has a phoneme similarity matrix and a word dictionary as given knowledge. The output for each word is a likelihood and an identified location.

The Word Predictor (described in Chapter IV) first predicts plausible words in an unknown portion of a recognized phoneme string by using syntactic, semantic and pragmatic information; it then sends these words to the Word Identifier. The Word Predictor has a spoken grammar with additional information which is represented by an augmented context-free grammar.

The Parsing Director (described in Chapter V) directs the Word Identifier, passing to it words to be identified and a portion of a phoneme string to be matched with them; it then builds up word strings

based on the identified results. It also directs the Word Predictor, passing to it some word strings to be parsed. In this process, on behalf of reducing the number of word strings, the director uses a new method for pruning the parsing tree.

The Responder (under development, not described in this thesis) answers questions or acts according to the result of speech understanding.

We have applied this system with great flexibility to restricted utterances which have a 101-word vocabulary. We call this application 'Computer Network' (shown in Chapter VII). LITHAN can be applied to new tasks with a minimal effort by changing only the grammar and word dictionary. We have applied LITHAN to other tasks which are called 'Arithmetic Expression' and 'Calendar' (shown in Chapter VI). The results of these experiments have verified that the contribution of linguistic information to recognition rate is closely dependent on the contents of the tasks (discussed in Chapter VIII).

In Appendix A, I will describe the on-line, real-time spoken word recognition system which is a by-product of the LITHAN system.

CHAPTER II

PHONEME RECOGNIZER

II-1 Introduction

There are two main problems which make it difficult to recognize phonemes in continuous speech: coarticulation and speaker difference. Various methods have been investigated to solve the former difficulty. They might be divided into three types as follows:

1. *Choice of a larger unit for recognition* (Kohda et al.[42]).....In an automatic speech recognition system, it is a very important problem as to what we choose as the minimal unit to be recognized *(phoneme?, syllable?, word?)*. If we choose a larger unit such as a syllable or word, we can escape the difficulty caused by coarticulation. This method may not be scientific, but practical for a single speaker.

2. *Application of phonological rules or rewriting rules* (Barnett[7], Cohen and Mercer[13], Mori et al.[52], Itahashi et al.[32]).....In this method, a phoneme is used as a unit, and some recognized phonemes are replaced with others using phonological rules or rewriting rules. However, I imagine it is to be very difficult to construct rules so flexible that they work well for many speakers and in various kinds of contexts.

3. *Approach by speech production model* (Fujisaki et al.[22], Kasuya et al.[37]).....This approach is further divided into generative and analytic approaches. Using a model of coarticulation in the formant frequency domain, Fujisaki et al. attempted to convert measured formant patterns into stepwise patterns representing the target values of the model. Although this approach is very theoretical, it consumes a large amount of processing time.

Kasuya et al. tried to normalize the second formant by using formants observed at the initial, middle and final positions of a phonemic segment.

Two approaches have been investigated in order to solve the problem of speaker differences: normalization and learning.

1. *Normalization approach*(Fujisaki and Kawashima[21], Gerstman[24], Scarr[86], Sambur and Rabiner[85]).....The fundamental idea of this approach is based upon the assumption that there exist certain invariant relationships in given acoustic features across different speakers or contexts. The idea is also based on the fact that vowel identification is dependent on listening to several different vowels spoken by the same person(Ladefoged and Broadbent[44]).

2. *Learning approach*(Kohda et al.[41]).....Although the method mentioned above is very theoretical, the current study still has many problems to be solved in application to speech recognition. On the other hand, the learning approach is a reliable and practical method, in which a standard pattern for an individual speaker is learned by that speaker's sample patterns. This learning approach will be divided into supervised and non-supervised learning approaches.

For coarticulation, we have adopted rewriting rules in phoneme recognition and word identification stages. For speaker differences, we have adopted the learning approach.

There are other difficult problems in phoneme recognition, such as the variations in acoustic signals caused by the rate of speech, or the speaker's emotions and dialect. And yet the most important one might be that of extracting the best acoustic features from a speech wave. We employed spectral analysis by a filter bank. Ichikawa et al. [30] and White and Neely[98] reported that a short time spectrum is one of the best acoustic features in the current art of speech analysis.

II-2 Outlines of Phoneme Recognizer

II-2-1 Speech Research System

Fig. II-1 shows the hardware structure of the speech research system(Sakai and Ohtani[80]). It is a sub-system of the inhouse computer network(KUIPNET; Kyoto University Information Processing NETwork, Sakai et al.[81]). The central host computer is NEAC 2200/250 with several mass storage devices. MELCOM 70 particularly shares the Input/Output part of speech processing: it is combined with an A/D and D/A converter, a graphic display, a serial printer, two cassette tape units, and so on.

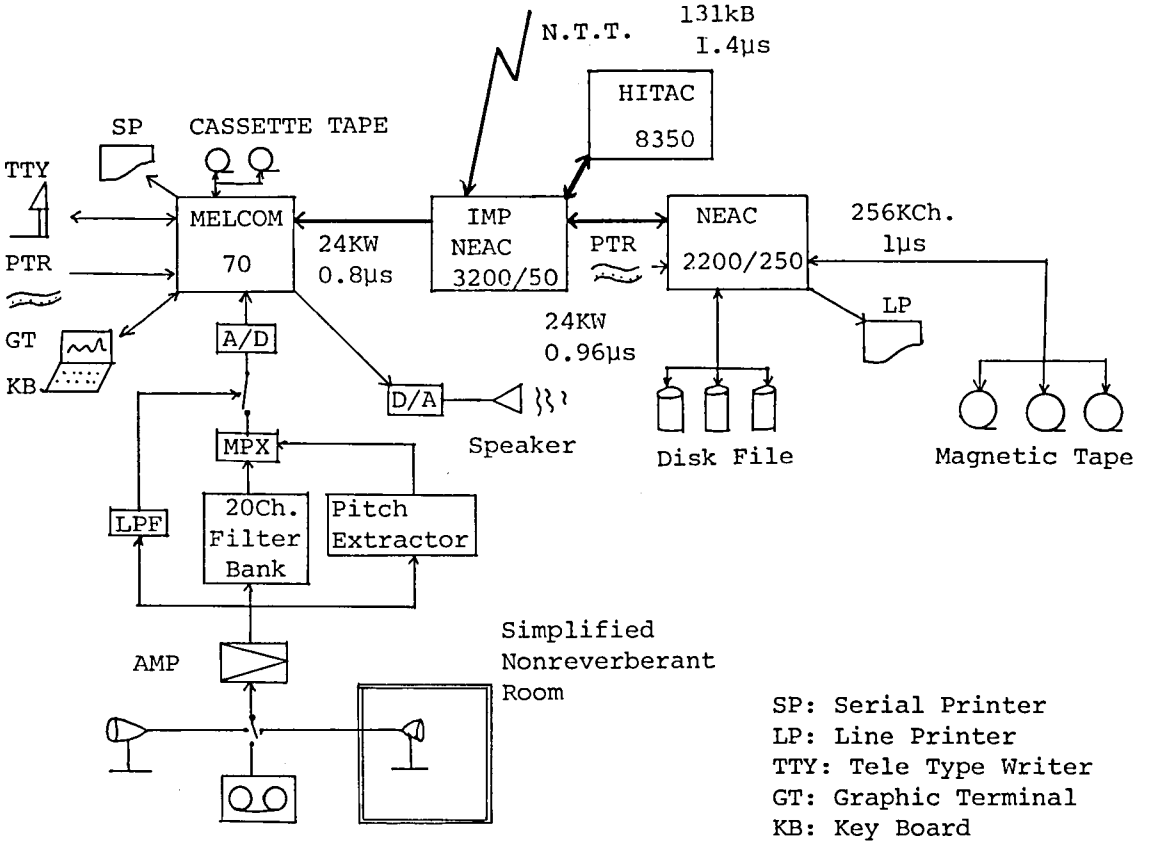


Fig. II-1 Hardware Structure of Speech Research System

NEAC 2200/250 is a medium-sized character-oriented machine. The current configuration contains 256K (6-bit) characters (1μsec cycle time), a line printer, three disk units, three magnetic tape units, a paper tape reader, and a paper tape puncher. Not only an assembly language, but also FORTRAN and LISP language are available.

MELCOM 70 is a minicomputer with 24K words memory (16 bits/word and 0.8μsec cycle time). The cassette tapes attached are used to store various program packages, so that these tapes can be loaded and run when necessary. The capability of high-speed calculation enables the minicomputer to process speech data while controlling the I/O devices in real time(Sakai, Nakagawa, and Maegawa[79]).

In addition to these, the system is equipped with a 20-channel filter-bank, pitch extractor, tape recorder, microphone, sound spectrograph and simplified nonreverberant room. The tape recorder used here is made by TEAC Co. Ltd.(Model R-341c), and it is usually used at the broadcasting station. The microphone used is made by SONY Co. Ltd.(Model ECM-52). This is a 'electret' condencer microphone. The simplified nonreverberant room is made by RION Co. Ltd.(2 cubic meters). As the result of rough measurement, we found that the sound-insulation-effect was more than 30dB above 100Hz(Tabata[91]).

II-2-2 Acoustic Processing

In this section, I describe the preprocessing phase of speech recognition: obtaining a spectrum representation of speech signals.

The first step in speech recognition by computer is preliminary processing of the acoustic input, traditionally called 'preprocessing'. This preprocessing should extract meaningful features from speech signals for speech recognition, and then provide speech data reduction. The kinds of preprocessing that have been used in speech analysis and recognition include Fourier transform, Cepstrum, linear predictive

coding, inverse filtering, auto-correlation, partial correlation, zero-crossing, filter banks (analog and digital), and so on.

In designing the preprocessing, we mainly considered two factors: performance for the recognition task and the processing time. The consideration led us to spectrum analysis by a filter bank.

Short time spectrum has the following features:

1. A distinct correspondence is proved to exist between a short time spectrum and physiological speech production theory (We can extract the formants even from a short time spectrum analyzed by a filter bank (Suzuki et al.[90])).
2. Its performance for speech perception task has been guaranteed by Channel Vocoder(Gold and Rader[25]).
3. Spectrum analysis by a filter bank is approximately the same as the acousto-mechanical operation of ears(Plomp[70]).
4. Spectrum analysis is easily performed in real time by a filter bank.

A speech signal is first passed into a pre-emphasis circuit with a slope of 6-dB per octave below 1600Hz because of improving to signal-to-noise ratio at high frequencies, and then fed into the 20-channel filter-bank. After they are full-wave-rectified and smoothed by the low-pass filter (cut-off frequency: 40Hz), the output waves are sampled at every 10ms interval and digitized with an accuracy of 10 bits. The center frequencies of the 20 channels used increase in order by a factor $2^{1/4}$. These frequencies are shown in Table II-1.

Table II-1 Center frequency(Hz) of bandpass filter

channel No.	center freq.	channel No.	center freq.	channel No.	center freq.	channel No.	center freq.
1	210	6	500	11	1190	16	2830
2	250	7	595	12	1410	17	3360
3	297	8	707	13	1680	18	4000
4	354	9	841	14	2000	19	4760
5	420	10	1000	15	2380	20	5660

Thus, the Acoustic Processor produces 10×20 bits/10ms, namely, 20,000 bits/sec, less in comparison with the about 80,000 bits/sec for telephone speech.

II-2-3 Outline of Phoneme Processing

The Phoneme Recognizer divides input speech into segments of utterances which are hypothesized to consist of a continum of a single phoneme; it then assigns one phoneme category to each segment. Fig. II-2 shows a block diagram of phonemic processing, called LISTEN (LImited Spoken Text ENcoder).

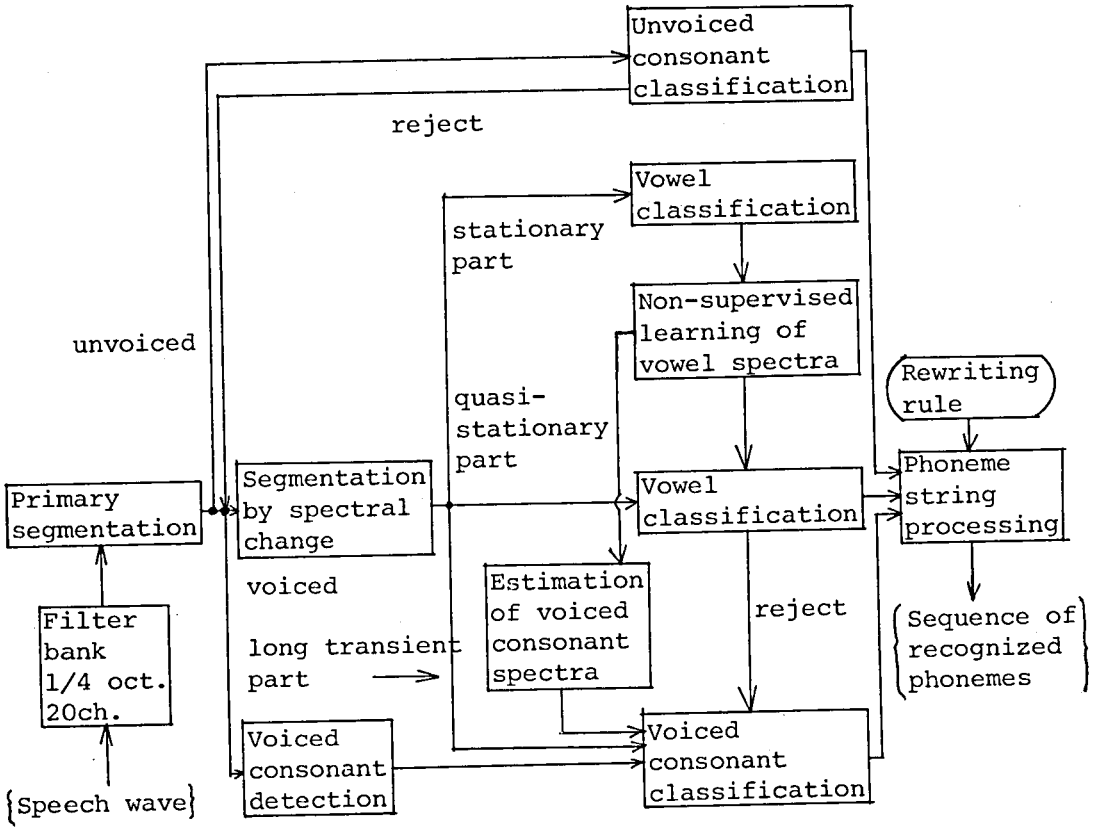


Fig. II-2 Block diagram of phonemic processing.

The phoneme recognizer first analyzes input speech by the filter-bank. Primary segmentation is performed on such analyzed speech, now represented by a sequence of short time spectra (I will call each spectrum a 'frame'). Each frame is classified into one of four groups: silence, voiceless-nonfricative, voiceless-nonplosive or voiced; based on the energy and deviation around the low or high frequency of (20-dimensional) spectrum. Each classified segment is recognized as one of phonemic categories. If a part of a sequence of recognized segments is composed of the same successive phonemic categories, these are combined. On the other hand, if a part is irregular, it is smoothed by using rewriting or phonological rules. The output of this algorithm is a sequence of continuous and non-overlapping segments.

A segment which has been regarded as included in a voiceless group is further classified into one of phonemes. This more detailed classification is based on the segment duration, the presence of silence preceding the segment, spectral change, etc.

For a segment classified into a voiced group, it is next determined whether each frame included in this segment is stationary, quasi-stationary or transient. Such determination is based on the degree of spectral change between adjacent frames (I call this the secondary segmentation).

The decided stationary and quasi-stationary parts are regarded as a presenting portion for vowels. The most stationary frames are used for partially non-supervised learning of the spectrum patterns of vowels. A portion of voiced consonants is detected as one of the following: 1) rejected by the vowel recognition process; 2) long transient; 3) having weak energy with concave speech level (or dip). The spectral patterns of voiced consonants are gradually trained (or estimated) by using those learned vowels.

Vowels and voiced consonants are recognized by Bayes' discriminant functions, which are obtained from the renewed standard patterns of each phoneme. Semi-vowels are recognized by applying rewriting rules to a recognized noisy phoneme string.

To reduce the influence of missegmentation on the system performance, the segmentation process is designed so that a voiced part may be divided into a segment finer than a phoneme. This division will be recovered in word identification process, for example, a vowel can be allowed to match with up to three segments and a consonant with up to two segments.

To the segment thus processed are given the first candidate phoneme, the second one, the degree of confidence(reliability) of the first candidate, and the segment duration. Also the string of segments would be corrected by phonological rules.

II-3 Representation of Phoneme

We use the following phoneme categories in our system.

1. vowel /a/,/i/,/u/,/e/,/o/
2. semi-vowel /y(j)/,/w/
3. nasal /m/,/n/,/$\tilde{g}(\eta)$/
4. voiced plosive /b/,/d/,/g/
5. liquid-like /r/
6. voiced fricative /z/,/dz/
7. voiceless plosive /p/,/t/,/k/
8. voiceless fricative /s/,/ʃ/
9. affricate /c/
10. aspirated /h/
11. syllabic nasal /N/
12. silence or pause /./,/-/,///*

* in the order of increasing duration, where /./,/-/,/// may correspond to a closure of voiceless plosive consonants, a choked sound and a pause between phrase boundaries (non-speech), respectively.

Without distinguishing, the system treats /z/ and /dz/ as the same phoneme, and /s/ and /ʃ/ as well. The system also does not distinguish among the voiceless plosive group; therefore, I may simply denote this group as /p/ or /p,t,k/. If a speech recognizer uses linguistic information such as a word dictionary, the classification becomes easier. Thus, a group with the same manner of articulation does not have to be broken down into a smaller unit, i.e., a phoneme, in the phoneme recognition level(Itahashi et al.[33]).

Japanese has three explosive nasals: /m/, /n/ and /$\tilde{g}$/. However, there is no opposition of nasals in the syllable-final position (syllabic nasal). Thus /m/ appears only before /p/, /b/, /m/; /n/ only before /t/, /d/, /n/; /$\tilde{g}$/ only before /k/, /g/, /$\tilde{g}$/; /N/ only before a pause, etc. Hattori[27] interprets that these implosive nasals (and nasalized vowels) correspond to the fourth (implosive) nasal phoneme /N/. We always treat these as /N/ for convenience (see section III-3-2).

In Japanese, /g/ appears generally in the initial position of words, while /$\tilde{g}$/ appears most frequency in the intervocalic position (implying Kyoto dialect).

II-4 Speech Analysis

In this section, we shall investigate some properties of speech which give us the fundamental ideas for speech recognition.

II-4-1 Statistical Property of Speech Spectrum

Statistical analyses of speech spectra have been made by many researchers. For example, techniques of multidimentional scaling and principal component analysis have been devised by a group of Dutch investigators(Pols, Kamp and Plomp[71], Klein, Plomp and Pols[39]). These men examined the correlation between the physical and perceptual dimensions, using a multidimensional scaling of speech spectra obtained by an 18-channel 1/3-octave filter bank. Also investigating vowel configuration by use of a principal component analysis, they found a high correlation between the configuration of the average vowels in the factor space and their configuration in the F_1-F_2 formant plane. Other statistical analysis was made by Tabata[91], who performed multivariate statistics by four factors (speaker, initial vowel, consonant and final vowel) on a set of VCV utterances (vowel-consonant-vowel) spoken by five male adults.

From different points we investigated the statistical properties of speech spectra. To provide the speech material, 245 meaningful words of VCV-types were spoken by ten male adults. In these words, V was selected from the five vowels /a/,/i/,/u/,/e/, and /o/, and C from the consonants /N/,/y/,/w/,/m/,/n/,/g̃/,/b/,/d/,/g/,/r/, and /z/. I will hereafter call this 'fundamental speech material'. By visual observation of the spectral patterns of these words, one spectral frame was extracted from each vowel and three from each consonant.

Since a spectrum is the output of a 20-channel filter bank, it can be considered to be a 20-dimensional vector. Hereafter we will use the following notations for representation of spectrum.

$Z(t)$: the amplitude outputs of the 20-channel 1/4-octave filters at time t. where $Z(t)=z_1(t), z_2(t), \ldots z_{20}(t)$ are the output of a bandpass filter. $Z(t)$ may considered an approximate representation of the output of the sonagraph.

$y(t)$: the normalized $Z(t)$, that is, $y(t)=Z(t)/|Z(t)|$, where $|Z(t)|=(z_1^2(t)+z_2^2(t)+\ldots+z_{20}^2(t))^{1/2}$.

$X(t)$: the logarithmic transformation of $y(t)$, that is, $x_i(t)=\log y_i(t)$.

In speech recognition, we may prefer the relative intensity between frequency components of speech sound spectrum to the instantaneous amplitude $Z(t)$. In other words, we normalize the square sum of the spectrum components to 1, yielding $y(t)$. Meanwhile, the auditory sense for the intensity of speech sound is said to be proportional to the logarithm of the intensity itself (Weber-Fechner law). Considering these two points, we use mainly the spectrum $X(t)$ unless noticed.

Now we assume that the spectra of each phoneme are distributed in a 20-dimensional vector space according to the multivariate normal distribution $p(X/i)$:

$$p(X/i)=\frac{1}{(2\pi)^{n/2}|\Sigma_i|^{1/2}}\exp\left\{-\frac{1}{2}(X-m_i)\Sigma_i^{-1}(X-m_i)^t\right\}, \qquad \text{(II-1)}$$

where, X is a spectrum, m_i and Σ_i are the mean vector and the covariance matrix of phoneme i, respectively, n is equal to 20 and t denotes the transposition. For every

phoneme, we obtained the Σ_i, $|\Sigma_i|$ and m_i for all the speakers and the individual speaker.

From these distributions, we can calculate the distance between two phonemes i and j in the spectral space. Various concepts of distance have been defined for such a purpose, e.g., Minkovski distance, Mahalanobis generalized distance, Kullback distance, Chernoff distance and Bhattacharyya distance, etc. We use Bhattacharyya distance, because it has a same order relation to the misrecognition rate in the recognition scheme based on Bayes' rule (Ichino and Hiramatsu[31]).

Bhattacharyya distance $B(i,j)$ is defined as following (Bhattacharyya [10]).

$$B(i,j)=\frac{1}{8}(m_i-m_j)\cdot\left\{(\Sigma_i+\Sigma_j)/2\right\}^{-1}\cdot(m_i-m_j)^t+\frac{1}{2}\log\left(\frac{|(\Sigma_i+\Sigma_j)/2|}{|\Sigma_i|^{\frac{1}{2}}\,|\Sigma_j|^{\frac{1}{2}}}\right) \qquad \text{(II-2)}$$

Table II-2 Bhattacharyya distance between spectral distributions of two phonemes to all speakers.

	a	i	u	e	o	N	y	w	m	n	g̃	b	d	g	r	z
a	0	8.0	6.1	3.9	3.1	4.4	3.7	2.3	5.7	5.0	4.6	5.9	5.5	6.8	4.0	7.4
i	8.0	0	2.1	3.4	5.5	1.9	2.3	7.5	2.9	2.6	1.3	2.0	2.4	2.2	2.3	3.3
u	6.1	2.1	0	3.3	2.5	1.4	2.8	3.7	1.7	1.6	0.8	0.9	1.7	2.1	1.7	2.8
e	3.9	3.4	3.3	0	3.9	3.3	1.0	4.9	4.5	3.2	2.6	3.5	2.9	4.0	2.1	4.1
o	3.1	5.5	2.5	3.9	0	2.6	3.8	1.1	3.9	3.9	2.5	2.6	3.6	4.4	2.7	5.6
N	4.4	1.9	1.4	3.3	2.6	0	3.2	3.5	1.3	1.3	1.4	1.8	2.7	3.3	2.1	4.5
y	3.7	2.3	2.8	1.0	3.8	3.2	0	4.2	4.0	3.4	2.3	3.0	2.8	4.2	1.6	4.1
w	2.3	7.5	3.7	4.9	1.1	3.5	4.2	0	4.8	4.9	3.7	3.8	4.9	6.3	3.2	7.7
m	5.7	2.9	1.7	4.5	3.9	1.3	4.0	4.8	0	1.0	1.7	1.9	3.1	4.0	2.2	5.6
n	5.0	2.6	1.6	3.2	3.9	1.3	3.4	4.9	1.0	0	1.5	1.7	2.0	3.7	1.6	3.9
g̃	4.6	1.3	0.8	2.6	2.5	1.4	2.3	3.7	1.7	1.5	0	0.9	1.3	1.6	1.4	2.6
b	5.9	2.0	0.9	3.5	2.6	1.8	3.0	3.8	1.9	1.7	0.9	0	1.2	2.0	1.3	3.5
d	5.5	2.4	1.7	2.9	3.6	2.7	2.8	4.9	3.1	2.0	1.3	1.2	0	1.9	1.5	1.7
g	6.8	2.2	2.1	4.0	4.4	3.3	4.2	6.3	4.0	3.7	1.6	2.0	1.9	0	3.1	2.7
r	4.0	2.3	1.7	2.1	2.7	2.1	1.6	3.2	2.2	1.6	1.4	1.3	1.5	3.1	0	3.5
z	7.4	3.3	2.8	4.1	5.6	4.5	4.1	7.7	5.6	3.9	2.6	3.5	1.7	2.7	3.5	0

Table II-2 shows the Bhattacharyya distance, which is calculated by using the spectrum of each phoneme averaged over all the speakers and phoneme environments. From this table, we find that the distance between two vowels is larger than that between two voiced consonants. The distance between two voiced consonants having the same manner of articulation (/m,n,$\tilde{g}$/ or /b,d,g/) is particularly small. This fact suggests that classification of these phonemes is more difficult than classification of those having the same place of articulation (/m,b/, /n,d/ or /$\tilde{g}$,g/). Note that the distance B(a,w) between the vowel /a/ and semi-vowel /w/ is comparatively small in spite of the difference in articulation. This is caused by the special fact that the semi-vowel /w/ is always followed by the vowel /a/ in Japanese. Conversely, the distance between /g/ and another phoneme is comparatively large, because /g/ always appears at the initial position of words.

Table II-3 Bhattacharyya distance between spectral distributions of two phonemes for each speaker, averaged over all speakers.

	a	i	u	e	o	N	y	w	m	n	$\tilde{g}$	b	d	g	r	z
a	0	15.9	11.3	8.4	6.2	12.6	10.4	8.0	13.2	11.5	11.8	10.2	11.0	15.5	7.4	15.9
i	15.9	0	5.4	7.5	13.3	7.3	8.1	21.5	9.9	9.2	5.5	6.0	7.5	7.3	5.8	8.6
u	11.3	5.4	0	6.8	5.6	6.4	8.8	10.6	6.8	6.2	4.5	3.7	5.9	6.0	4.7	7.2
e	8.4	7.5	6.8	0	7.8	10.0	5.7	15.6	12.5	10.0	7.4	8.0	7.9	9.9	5.1	10.8
o	6.2	13.3	5.6	7.8	0	9.4	11.1	6.0	11.3	10.4	8.4	6.5	9.4	10.8	6.4	13.1
N	12.6	7.3	6.4	10.0	9.4	0	17.2	16.8	8.1	7.6	8.1	9.1	12.5	15.5	9.6	17.2
y	10.4	8.1	8.8	5.7	11.1	17.2	0	16.5	18.1	16.0	12.2	11.0	12.1	15.7	8.0	16.1
w	8.0	21.5	10.6	15.6	6.0	16.8	16.5	0	19.2	18.1	14.7	12.1	20.6	20.9	11.6	28.4
m	13.2	9.9	6.8	12.5	11.3	8.1	18.1	19.2	0	6.6	9.0	8.5	13.4	15.1	9.8	20.5
n	11.5	9.2	6.2	10.0	10.4	7.6	16.0	18.1	6.6	0	7.8	7.9	9.5	14.1	8.6	14.3
$\tilde{g}$	11.8	5.5	4.5	7.4	8.4	8.1	12.2	14.7	9.0	7.8	0	5.6	6.9	9.4	6.7	10.5
b	10.2	6.0	3.7	8.0	6.5	9.1	11.0	12.1	8.5	7.9	5.6	0	5.4	7.1	4.9	9.6
d	11.0	7.5	5.9	7.9	9.4	12.5	12.1	20.6	13.4	9.5	6.9	5.4	0	8.4	6.3	6.8
g	15.5	7.3	6.0	9.9	10.8	15.5	15.7	20.9	15.1	14.1	9.4	7.1	8.4	0	9.4	8.8
r	7.4	5.8	4.7	5.1	6.4	9.6	8.0	11.6	9.8	8.6	6.7	4.9	6.3	9.4	0	9.3
z	15.9	8.6	7.2	10.8	13.1	17.2	16.1	28.4	20.5	14.3	10.5	9.6	6.8	8.8	9.3	0

Table II-3 shows the Bhattacharyya distance obtained by averaging the distances calculated for each speaker. Note that it is generally larger than the former in Table II-2; however it must be remembered that this is calculated from a smaller number of phoneme samples. If a speaker is fixed, in short, the recognition becomes easier than that for unspecific speakers. This conclusion is consistent with that of Tabata[91], that at a stationary part of the nasal consonants, the effect of the speaker factor is larger than that of the consonant factor.

Table II-4 shows the variance of spectral distribution for each phoneme in the spectral space. We use three measures as follows:
(a) The first is a logarithmic transform of the determinant of covariance for each phonemic spectral distribution over all speakers, that is, $log[\det\Sigma_i]$.
(b) The second is an average over all speakers of the logarithmic transform of the determinant of each phonemic distribution for each speaker (inter-speaker), that is, average $log[\det\Sigma_i^s]$.

These measures are approximately based on the linear relation of the determinant of volume (or variance) of distribution in the space. For example, we

Table II-4 Variance of each phoneme in the spectral space.

(a) logarithmic transform of determinant of spectral distribution to all speakers.

(b) average over all speakers of logarithmic transform of determinant of spectral distribution for each speaker.

(c) average of inter-speaker for Bhattacharyya distance.

	a	b	c	No. of samples
a	-73.4	-84.9	3.9	1088
i	-58.4	-73.0	6.1	838
u	-54.2	-67.4	4.8	880
e	-67.3	-80.9	5.4	885
o	-66.7	-76.6	3.9	921
N	-57.5	-88.3	20.5	750
y	-67.7	-97.3	21.5	450
w	-74.4	-103.2	15.4	450
m	-62.0	-88.8	17.8	750
n	-61.1	-88.4	17.9	750
$\widetilde{g}$	-55.6	-77.5	11.9	750
b	-56.9	-73.9	7.8	750
d	-60.0	-82.1	10.5	450
g	-53.3	-71.1	7.2	750
r	-63.5	-80.4	7.1	750
z	-60.1	-80.4	8.8	750

can presume that /a/ is the most unstable vowel in the spectral space. (c) The third measure is an average of the Bhattacharyya distance over all speakers (inter-speaker), that is, average $B(i_j, i_k)$, where i denotes a phoneme, and j and k denote speakers.

From the third measure, we find that /N/,/y/,/w/,/m/,/n/ and /$\tilde{g}$/ are more influenced by the speaker factor than by other phonemes (or the phoneme factor).

II-5 Primary Segmentation

If we use the phoneme as the smallest unit in speech recognition, a speech recognizer must be able to segment a continuous speech utterance into discrete parts, each corresponding to a single phoneme. However, such a segmentation is as difficult as the recognition of phonemes, because acoustic characteristics are never invariant within a given phoneme. So we divide the segmentation process into two stages: primary and secondary. In the latter stage, we will adopt feed back from phoneme recognition to segmentation.

In the primary segmentation stage, input speech (a time-series of spectra) is segmented into four groups: silence (non-speech), unvoiced (or voiceless) non-fricative, unvoiced non-plosive, and voiced. Here we introduce three new parameters which are comparatively stable against time within a segment.

AMP_n = the power of the n-th frame, defined as

$$(z_1^2 + z_2^2 + \ldots z_{20}^2)^{1/2}.$$

CNL_n = the minimum channel number j which satisfies the next condition:

$$\sum_{i=1}^{j} y_i \geqq 0.5.$$

This parameter gives us an estimate for the deviation of energy in a lower frequency.

$$MOM_n = y_{18} + 2 \times y_{19} + 3 \times y_{20}.$$

This parameter indicates the spectral deviation (or moment) in a higher frequency.

Further, the average of each parameter within a segment is simply denoted by AMP, CNL, and MOM, respectively. These notations will be used in the following section. The steps for primary segmentation are as follows:

(a) Processing of a frame

We assign to the n-th frame:

(1) silence if $AMP_n < 15$ and $CNL_n \leq 7$, or $AMP_n < 10$.

(2) voiced if $AMP_n \geq 15$ and $CNL_n \leq 5$, or $AMP_n \geq 15$ and $MOM_n < 1.0$.

(3) and unvoiced otherwise.

(b) Segment processing

(4) If the n-th and following frames belong to the same group, they are combined into a single segment and their parameters are averaged.

(5) In a segment, if there exists an unvoiced frame which satisfies the condition $CNL \geq 15$, then we assign it a group of unvoiced non-plosive consonants, otherwise we assign a group of unvoiced non-fricative consonants.

(6) go to (1).

In these steps, we explicitly use the phonological rule in Japanese that two consonants never occur successively, exception the

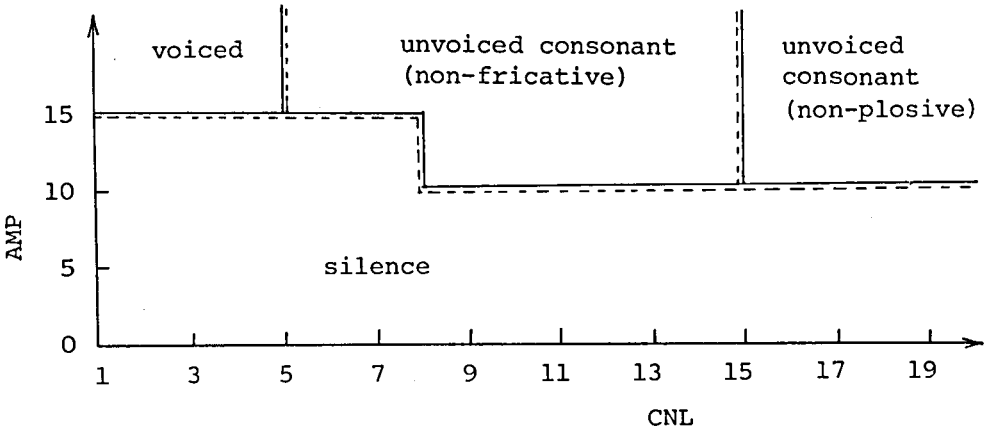


Fig. II-3 Primary segmentation on the CNL-AMP plane

case of vowel devocalization. (This system regards the syllabic nasal /N/ as a vowel.)

Fig. II-3 illustrates the region in the AMP-CNL plane where each group is contained. This procedure yields a sequence of segments consisting of four groups.

II-6 Classification of Unvoiced Consonants

A segment classified by primary segmentation as unvoiced is further classified into one of the detailed groups, each of which corresponds to either a phoneme such as /s/, /c/, /h/, or /p,t,k/, or to a voiced sound which may have been previously mis-segmented into an unvoiced section. Now I describe this procedure. We will first define some necessary parameters.

DUR = the number of frames within the segment to be classified. This corresponds to the duration of the segment.

PRAMP = the average power of the three successive frames which precede the segment.

SL = min(1.0, 1.3 - PRAMP/50). This measure gives the likelihood of silence or closure for plosive consonants on the basis of speech power. Min is an operator which selects the smallest value among the several values. onents.

AMPMAX = the maximum value of power in the segment.

AMPMIN = the minimum power between the frame with maximum power and the last frame of the segment.

These parameters are illustrated in Fig. II-4.

Using them, we calculate the likelihood of the voiced group and of each of the unvoiced consonants. This likelihood is normalized so that the value becomes larger than 1.0 for a typical segment corresponding to a phoneme. As this likelihood value becomes larger than the criterion (1.0), the degree of reliability for phoneme classification becomes higher. This procedure easily lead us to select

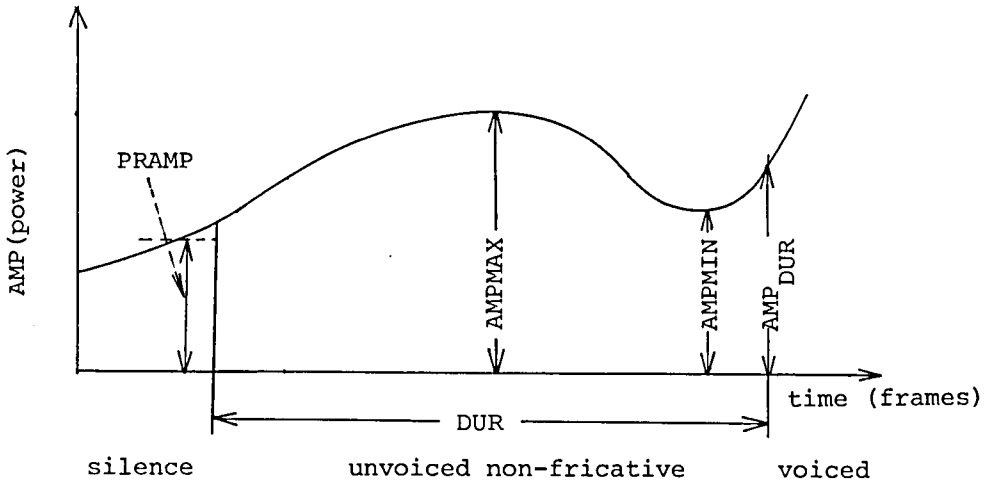


Fig.II-4 Illustration of some parameters

the best threshold for each parameter, because the parameters are regarded as independent ones to one another. We classify segments into phonemes which have the highest likelihood. If the likelihood of "voiced" becomes highest, the segment is regarded as "voiced" and will be subjected to that classification process. The following is the procedure to calculate the likelihood of each phoneme category. When the attached condition is satisfied, the operation in brackets is carried out, otherwise, the value is regarded as 1.0.

(1) Classification of Unvoiced Non-Fricative Segment

Likelihood of "voiced"

$$= \frac{AMP}{200} \times \frac{1}{MOM} \times \{\frac{DUR}{12} \text{ ; if } DUR \geqq 15\} \times \{\frac{8}{CNL} \text{ ; if } CNL \geqq 8\}$$

$$\times \{1.2 \text{ ; if } (AMP_1 = AMPMAX) \vee (AMP_{DUR} = AMPMAX)\}$$

The calculation is quite dependent on the voiced group (/a/ in most cases) which tends to be misclassified into an unvoiced non-fricative group by primary segmentation. The condition in the fifth term shows that of monotonous power change.

Likelihood of the unvoiced plosive /p,t,k/

$$= SL \times \{1.5 \text{ ; if } DUR \leqq 4\} \times \{(\frac{AMPMAX}{AMP} + \frac{AMPMAX}{AMPMIN} + \frac{AMP_{DUR+1}}{AMPMIN})$$

$\times$ 1/4.5 ; if DUR$\geq$4} $\times$ {$\frac{8}{DUR}$; if 4<DUR$\leq$7}

The third term represents the condition of convex power change, corresponding to unvoiced release-aspiration.

Likelihood of the aspirated /h/

= 1.0 $\times$ {$\frac{DUR}{8}$; if (DUR>8) $\wedge$ (AMP$\leq$150)} $\times$ {$\frac{DUR}{4}$; if DUR$\leq$4}

$\times$ {$\frac{CNL}{9}$; if CNL>9} $\times$ {$\frac{50}{AMP+10}$; if (AMP$\leq$40) $\wedge$ (AMPMAX< AMP$\times$1.3)}

(2) Classification of an Unvoiced Non-Plosive Segment

Likelihood of the aspirated /h/

= (3.7 - MOM) $\times$ {$\frac{60}{AMP+10}$; if AMP$\leq$40}

Likelihood of the affricate /c/

= min (SL $\times$ 1.2, 1.0) $\times$ {1.2 ; if 2.5<MOM<3.4} $\times$

{1.2 ; if DUR$\geq$7} $\times$ {($\frac{AMPMAX}{AMP_{DUR/2-2}}$ + $\frac{AMPMAX}{AMP_{DUR/2+2}}$) $\times$ $\frac{1}{2.4}$;

if DUR$\geq$8}

The first term represents that the silence preceding an affricate is not as clear as that of an unvoiced plosive. The fourth term shows the condition of convex change of speech power.

Likelihood of the fricative /s/

= (MOM - 2.3) $\times$ {$\frac{DUR}{8}$; if DUR<8} $\times$ {$\frac{DUR}{12}$; DUR>12}

The relation of major positive feature parameters for unvoiced segments is shown in Table II-5. The right-hand value illustrates a criterion for each feature which becomes positive for its corresponding to a phoneme. Although each phoneme has various negative features, we mainly use only the positive ones. From the above procedure, we can obtain the first two phonemes with highest likelihood. These are regarded as the first and second candidates, respectively.

Table II-5 Relation of main positive feature parameters for unvoiced consonants.
(the value of right side denotes a criterion)

feature parameter	non-fricative			non-plosive		
	voiced	p,t,k	h	h	c	s
power AMP	$\geq$200		<40	<40		
duration DUR (frames)	$\geq$12	$\leq$7	$\geq$8		>7	>12
high frequency MOM	$\leq$1.0			$\leq$2.7	>2.5 <3.4	$\geq$3.3
low frequency CNL	$\leq$8		$\geq$9			
* SL		=1.0			$\geq$0.875	
change of power	monotone	convex	non-decreasing		convex	

* the likelihood of silence at the preceding segment

As soon as the classification of unvoiced segments has been finished, rewriting rules are applied to the new classified sequence. For the first candidate, let F_n be the label (phoneme) for the n-th segment, and let DUR_n be the duration. Then the following rules apply:

Rule 1 If F_n = /h/ (or /p,t,k/), F_{n+1} = silence and $DUR_n \leq$20ms, then these two segments are combined into one segment, and it is labeled /h/ (/p,t,k/).

Rule 2 If F_n = silence, F_{n+1} = unvoiced plosive, $DUR_n \leq$20ms and F_{n+2} = silence, then F_{n+1} is rewritten as "silence", and these segments are merged into one segment, and it is labeled "silence".

Rule 3 If F_n = voiced and $DUR_n \leq$20ms, then this segment is absorbed by the preceding segment.

Rule 4 If F_n = voiced, F_{n+1} = /p,t,k/ and $DUR_{n+1} \leq$30ms, then the

degree of reliability of /p,t,k/ is slightly decreased.

II-7 Secondary Segmentation

(a) Detection of stationary parts

A segment, which has been classified as "voiced" by either primary segmentation or the classification process of an unvoiced segment, is further segmented into stationary or transient part. The procedure is based on a statistical test which examines the change between two adjacent spectra. This test can decide whether any frame is stationary or transient. We can calculate the degree of stationary for a given frame from the four following probability densities:

1. P(S) and P(T) : *a priori* probability density functions of the occurrence of stationary and transient frames, respectively, where P(S) + P(T) = 1.0.
2. P(d/S) and P(d/T) : *a posteriori* probability density functions of the degree of spectral change (d) in stationary and transient parts, respectively.

The degree of spectral change (d) at time t is obtained from the following equation, using certain transformed spectra U(t-1), U(t), and U(t+1).

$$d = \{|U(t-1) - U(t)| + |U(t) - U(t+1)|\} / 2 \qquad \text{(II-3)}$$

If these four probability densities P(S), P(T), P(d/S) and P(d/T) are known, we can calculate the conditional probabilities, P(S/d) and P(T/d) for a given d. The statistical test employs the likelihood ratio;

$$\lambda = \frac{P(S/d)}{P(T/d)} = \frac{P(S)\cdot P(d/S)}{P(T)\cdot P(d/T)} \qquad \text{(II-4)}$$

If λ is greater than or equal to 1, the given frame is decided to be stationary; otherwise, it is transient. Assuming that both P(d/S) and P(d/T) are normal distribution density functions, we calculated their

means and variances from many speech materials. In this case, by visual observation of the spectral pattern, we decided whether a frame is stationary or transient. If we adequately change the value of P(S), we can extract only the desirable rate for the stationary parts. In LITHAN, we use two different values of P(S) and spectra u(t), with the following aims:

(1) Learning process : In order to adapt the system to different speakers, we need to automatically collect some spectra of vowels and syllabic nasal of the current speaker out of frames which are decided to be stationary with the high reliability. At this stage, we use the logarithmic transform of z(t) as u(t) in Eq.(II-3), and assume P(S) as 0.1.

(2) Recognition process : This is the stage after the system has been adapted to a speaker. It is now desired to extract as stationary and quasi-stationary parts the frames corresponding to the vowels and syllabic nasal. In order to extract them, we use y(t) as u(t), and assume P(S) as 0.5. The probability densities in such cases are shown in Table II-6.

Table II-6 Parameters of P(d/S) and P(d/T)

	u(t)=log z(t)		u(t)=log y(t)	
	P(d/S)	P(d/T)	P(d/S)	P(d/T)
mean	0.386	1.05	0.136	0.271
variance	0.0361	0.281	0.00495	0.0135

(b) Detection of voiced consonants

The portion of voiced consonants is detected as one of the following:

(1) portion rejected by the vowel recognition process.

(2) long transient portion.

(3) portion of small sound power concave form.

In Japanese, a voiced consonant does not exist at the final

position of a word. We make good use of this phonological property, as it can be rephrased in a convenient form for our purpose, that is, a voiced consonant cannot exist at the end of a series of voiced segments. However, at the initial position of a voiced segment there may exist a voiced consonant to be preceded by a voiceless consonant. This is because a vowel (/i/ or /u/) which a voiceless consonant follows is sometimes devocalized or missed. Owing to the above consideration, the initial position of a voiced segment which has transient parts longer than 50ms can be regarded as a voiced consonant.

Since an inter-vocalic voiced consonant always has smaller energy than the surrounding vowels, an energy contour in such portions forms a concave, or dip. Therefore we can detect these portions by approximating five successive points on the contour by a secondary degree curve.

II-8 Classification of Vowels

As described in section II-1, the most difficult problem in the automatic speech recognition is due to variations in spectrum patterns caused by coarticulation and speakers. Compared with the coarticulation problem involved in recognition, the speaker problem is essential to the recognition of vowels (see section II-8-3). While most systems which have succeeded have been able to store training data for every possible user, they still cannot adapt or train themselves to fit to new speakers very rapidly. Whereas the turn-around time required for new users is often a major factor limiting the use of speaker-dependent systems. In this section, I describe both methods of a partially non-supervised learning of vowel spectra and rapid supervised learning.

II-8-1 Partially Non - Supervised Learning of Vowel Spectra

In non-supervised learning, the system must be able to treat training data very carefully in order to avoid mislearning. From our preliminary experiments (which will be described in section II-8-3), we found that vowels can be recognized very accurately in the stationary parts extracted from the learning process mentioned in the previous section. Therefore, the spectrum of a stationary part where vowels are recognized with high reliability is a suitable sample for (partially) non-supervised learning. Let us consider the way to learn the average spectrum m_i^j of the vowel i for the speaker j. Given N samples, x_1, x_2, ... x_N, uttered by the speaker j and recognized as the vowel i, we can use the method for learning the mean vector (Nilsson[65]).

The value of m_i^j at the (N-1)th stage of learning is denoted by $m_{i,N-1}^j$. This is renewed by the N-th sample x_N as follows:

$$m_{i,N}^j = \frac{1}{\alpha+N}(\alpha \cdot m_i + N \cdot x_{1,N}) = \frac{1}{\alpha+N}\{(\alpha+N-1)m_{i,N-1}^j + x_N\} \qquad \text{(II-5)}$$

$$x_{1,N} = \frac{1}{N}\sum_{k=1}^{N} x_k \qquad \text{(II-6)}$$

$$m_{i,0}^j = m_i, \qquad \text{(II-7)}$$

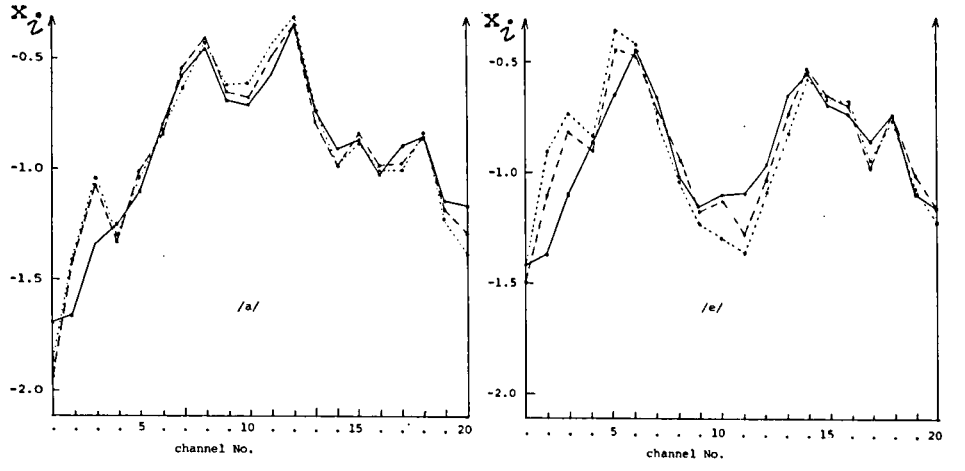


Fig. II-5 Non-supervised learning of mean vector of vowel spectrum
—— initial vector; - - - final vector; ····· target vector.

where m_i is a mean vector for all speakers, and α is a consonant value representing the degree of ambiguity of m_i^j. We substituted 10 for α in this experiment.

The Phoneme Recognizer selects training samples automatically, and recognize vowels and syllabic nasal. Fig. II-5 illustrates examples of the results of non-supervised learning. Table II-7 shows how different the learned spectra are from speaker to speaker. This learning was performed by the use of two different sets of training samples, where each set consisted of 8 sentences of 'Arithmetic Expression'.(The details will be described in Chapter VI.) The column in the table corresponds to the spectra learned by the first set (suffix with 1). The difference between two speakers, S_1 and S_2, is defined as following:

Table II-7 Difference of learned mean spectra between speakers

	SK_2	OT_2	MK_2	NK_2	YS_2
SK_1	0.59	2.87	1.72	2.86	2.15
OT_1	2.19	1.27	2.67	2.10	2.38
MK_1	1.53	3.29	1.01	2.08	1.80
NK_1	2.56	2.84	2.86	1.07	1.98
YS_1	1.66	2.18	1.36	1.67	0.67

$$D(S_1,S_2) = \sum_i \sum_{j=1}^{20} |x_{j,i}^{S_1} - x_{j,i}^{S_2}| \quad ; \quad i = a,i,u,e,o,N, \qquad \text{(II-8)}$$

where j denotes a channel number.

From the table, we find that this learning method is very useful in the following ways:

(a) The values of the diagonal are smaller than the others.

(b) This value is independent of training samples.

(c) No special training sample is required for the system user.

II-8-2 Supervised Learning of Vowel Spectra

Although (partially) non-supervised learning of vowel spectra is effective for many speech materials as shown, the convergence speed of this learning is rather slow. We propose a new technique for

supervised learning, the main feature of which is speed.

Before practical recognition is carried out, and so that a proper voice can be learned, a new user of the system is requested to utter the five vowels (/a/,/i/,/u/,/e/,/o/) in isolation and the syllabic nasal (/N/) in the context /uN/. Among all the voiced consonants, the largest inter-speaker variation occurs with the syllabic nasal (see section II-4-1). Thus, the system also learns it as well as the vowels. Fig. II-6 illustrates how the learning samples are extracted from the utterances. For vowels, the system extracts the three successive frames around the one with the largest energy; for the syllabic nasal, the three successive frames in the portion preceding by 50ms from the end of the utterance

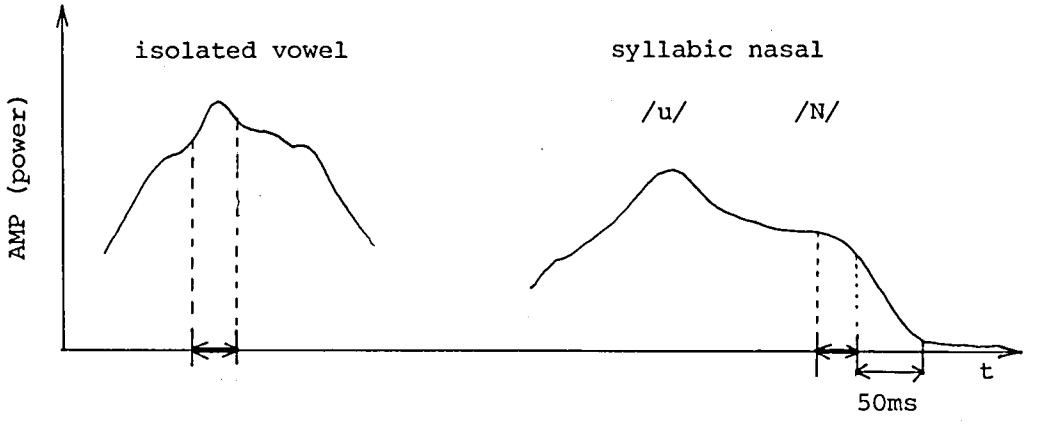


Fig. II-6 Portions of extracted samples for preliminary learning

Next, I describe in detail the learning method of both spectra and the weights of linear discriminant functions.

The spectral difference between isolated and in-word vowels can be large (Kanamori[36]). Thus it is risky to use the spectra of isolated vowels for recognition of vowels in continuous speech. Now let y_i ($=y_{1,i}, y_{2,i}, \cdots y_{20,i}$) be the spectrum common to all speakers for the isolated vowel i, and y_i^S be the spectrum of the speaker S. And also, let $\bar{y}_i$ and $\bar{y}_i^S$ be the spectra in words. We can previously

know only y_i and $\bar{y}_i$. As mentioned above, y_i^S is obtained from the isolated vowels. We want to know $\bar{y}_i^S$ by using these known spectra. The system estimates $\bar{y}_i^S$ by the following equation, where $\hat{y}_i^S$ denotes the estimator of $\bar{y}_i^S$:

$$\hat{y}_i^S = (\alpha \cdot \bar{y}_i + y_i^S) / (\alpha+1) \qquad \text{(II-9)}$$

Table II-8 Estimation error of vowel spectrum in-words

α	$\varepsilon_1=\Sigma\|\ \|$	$\varepsilon_2=\Sigma(\ )^2$	note
0	97.8	15.88	isolated vowel spectrum for each speaker
1	60.3	6.77	
2	53.7	5.43	
3	52.3	5.07	
4	52.0	4.96	
5	52.0	4.93	
6	52.1	4.93	
7	52.2	4.94	
8	52.4	4.95	
9	52.5	4.97	
10	52.9	4.97	
∞	54.5	5.28	common vowel spectrum in-words for all speakers

Table II-8 shows the estimation errors ε_1 and ε_2 which are evaluated by the following:

$$\varepsilon_1 = \sum_S \sum_i \sum_{k=1}^{20} |\bar{y}_{k,i}^S - \hat{y}_{k,i}^S| \ , \quad \varepsilon_2 = \sum_S \sum_i \sum_{k=1}^{20} (\bar{y}_{k,i}^S - \hat{y}_{k,i}^S)^2 \qquad \text{(II-10)}$$

The speech materials used are the same as analyzed in section II-4, that is, 'fundamental speech material'. In this table, $\alpha=0$ corresponds to the pre-registration of an isolated vowel for each speaker, and $\alpha=\infty$ to the usage of the vowels extracted from the connected speech common to all speakers. From these evaluations, we can find that the best value of α which results in the smallest ε_1 and ε_2 is $\alpha=4$ or 5. Table II-9 shows the estimation error for each vowel spectrum in-words. Seeing this table, it is observed that the

Table II-9 Estimation error of each vowel spectrum in-words ;

α	0		2		4		∞	
	ε_1	ε_2	ε_1	ε_2	ε_1	ε_2	ε_1	ε_2
a	12.74	1.82	6.36	0.43	5.87	0.34	5.89	0.34
i	15.37	1.88	9.51	1.03	9.70	1.06	10.08	1.24
u	18.68	3.26	9.61	1.04	9.35	0.94	9.93	1.00
e	12.51	1.43	7.27	0.53	7.24	0.51	7.83	0.57
o	16.26	2.52	7.29	0.54	6.36	0.41	6.16	0.36
N	22.28	4.48	13.69	1.83	13.51	1.70	13.87	1.77

estimation of /a/ and /o/ makes little difference in comparison with the case of no-estimation ($\alpha=\infty$). However, the estimation of these spectra is fortunately better than those of other phonemes.

The learning of linear discriminant functions, which discriminate a category in the vector space, is also considered. Let $\overline{w}_{ij}$ be the weighting vector of the linear discriminant function distinguishing the vowel i from j. Then, the estimator of $\overline{w}_{ij}$ for the speaker S is given by the following:

$$\hat{w}^S_{ij} = (\alpha \cdot \overline{w}_{ij} + y^S_i - y^S_j) \;/\; (\alpha+1) \;, \qquad \text{(II-11)}$$

The same technique as mentioned in non-supervised learning will be used for two or more training samples.

In order to evaluate the effect of the learning, we recognized, on the basis of linear discriminant functions, the vowels in digits spoken isolatedly by 10 male adults. The experimental results are shown in Table II-10. From this table, we can see that the best value of α is 2. Further, it is clear that this method has almost the same ability as in the case where $\overline{y}^S_i - \overline{y}^S_j$ is used as w^S_{ij}. In the estimation of spectra (statistical test) and in the performance, the best values of α are different. I think that this difference is caused by both the non-linear relation between the two estimations and the usage of different speech materials.

This supervised learning will be used for the on-line, real-time,

spoken words recognition system -LISTEN- (see Appendix A).

Table II-10 Recognition results of vowels in digits(number of errors). Number of samples: /a/=200, /i/=100, /u/=/e/=50, /0/=150

α	a	i	u	e	o	total	note
0	31	24	10	4	55	124	isolated vowel spectrum
2	20	6	5	1	18	50	learning by isolated vowels
4	30	10	5	2	25	72	learning by isolated vowels
∞	44	6	6	2	9	67	common vowel spectrum
\	27	4	2	1	10	44	spectrum in-words for each speaker

II-8-3 Classification Method of Vowels

There are various methods for classifying into each category observed samples in the feature parameter space (Nagy[55]). We investigated how large difference these methods yielded in the rate of vowel recognition. In order to conduct this comparison, we used cityblock distance (or the Chebyshev norm), Euclidian distance, linear discriminant functions, and quadratic discriminant functions as classification algorithms. It was assumed that the observed samples originated from multivariate normal (or Gaussian) populations with the mean m_i equal to the spectral mean for the vowel i, and the covariance matrix Σ_i equal to the spectral covariance matrix. It was also assumed that the *a priori* probability $P(i)$ is equal to the probability of the input belonging to the category of i.

The algorithm based on cityblock and Euclidian distance assigns an observed sample (spectrum) to the category of the nearest reference point. These two distances are defined by r=1 and r=2, respectively, in the following Minkovski distance.

$$d(i,j) = \left(\sum_{k=1}^{20} |x_{k,i} - x_{k,j}|^r \right)^{1/r} \qquad \text{(II-12)}$$

A discriminant function $f_{ij}(X)$ is a single-valued function to

distinguish the category i from the category j. Using this function, we assign the category i to an observed sample X if $f_{ij}(X) \geq 0$. The discriminant analysis leads to a linear discriminant function of the form

$$f_{ij}(X) = X \cdot W_{ij}^t + w_{0,ij} \quad , \tag{II-13}$$

where the weight vector and constant value are obtained by the following equations:

$$W_{ij}^t = (P(i) \cdot \Sigma_i + P(j) \cdot \Sigma_j)^{-1} \cdot (m_i - m_j)^t \tag{II-14}$$

$$w_{0,ij} = -\frac{1}{2}(m_i + m_j) \cdot (P(i) \cdot \Sigma_i + P(j) \cdot \Sigma_j)^{-1} \cdot (m_i - m_j)^t + (P(i) - P(j))/2 \cdot P(i) \cdot P(j) . \tag{II-15}$$

In order to classify n categories, ${}_nC_2 = n(n-1)/2$ linear discriminant functions are necessary.

Under our assumption, the probability $P(X/i)$ that a sample belonging to the category i is observed at the point X is given as follows:

$$P(X/i) = (2\pi)^{-n/2} \cdot |\Sigma_i|^{-1/2} \cdot \exp\{-\frac{1}{2}(X - m_i) \cdot \Sigma_i^{-1} \cdot (X - m_i)^t\} \tag{II-16}$$

The following quadratic discriminant function is derived, under the assumption that all the occurrence probabilities $P(i)$ are equal, as well as Bayes' recognition rule that the category i is assigned to an observed sample X when the probability $P(X/i)$ that the vector belongs to the category i is the largest.

$$f_{ij}(X) = (X - m_i) \cdot \Sigma_i^{-1} \cdot (X - m_i)^t + \log|\Sigma|_i - (X - m_j) \cdot \Sigma_j^{-1} \cdot (X - m_j)^t + \log|\Sigma|_j \tag{II-17}$$

We carried out recognition experiments by these four algorithms, (r=1 and 2 in Eq.(II-12), Eq.(II-13) and Eq.(II-17)) extracting training and test speech samples from 'fundamental speech material' as described in section II-4. In method 'A' (speaker independent), all necessary parameters are common to all speakers. On the other hand, in method 'B' (speaker dependent), all parameters except the covariance matrices are adjusted to each speaker. It is

Table II-11 Results of vowel recognition by various classification methods.

	method	cityblock distance	Euclidean distance	linear discriminant function	quadratic discriminant function
A	speaker-independent (common spectrum)	86.3%	87.4%	91.8%	93.0%
B	speaker dependent (individual spectrum)	91.8%	92.2%	95.0%	95.0%

seen from Table II-11 that the quadratic discriminant function has the most powerful performance in vowel recognition. Even though this function consumes computational time in comparison with other algorithms, we will use it hereafter.

A quadratic discriminant function needs a mean vector and a covariance matrix for each category, and these statistics depend on each speaker. So we investigated the following cases where:

(a) the means and covariance matrices are common to all the speakers;

(b) the means are personally adjusted but the covariance matrices are common;

(c) both the means and covariance matrices are personally adjusted.

Table II-12 shows the recognition results, where training and test speech samples spoken by five male adults were extracted from the sentences of 'Arithmetic Expression'. Although the method (c) gives the best performance, a personally adjusted covariance matrix cannot, in general, be obtained because it takes a great number of samples to

Table II-12 Results of vowel recognition by quadratic discriminant function in parts used for reference samples extraction.

method	reference spectrum		a	i	u	e	o	total
	mean	covariance						
a	common	common	100.0%	90.9%	93.2%	92.4%	95.0%	95.3%
b	individual	common	99.6	98.5	96.0	100.0	98.0	98.5
c	individual	individual	99.6	99.2	98.9	100.0	100.0	99.5

calculate it accurately. However a personally adjusted mean vector could be obtained by the learning procedure, as described in the preceding sections. Therefore our system approximately learns (b) from (c) by using the (partially) non-supervised learning procedure. On the other hand, the real time spoken words recognition system LISTEN, which will be described in Appendix A, learns (b) from (a) by using supervised learning procedure. If the conditional probability $P(X/i)$ for all i of a given sample is smaller than the pre-set thresholds (in relation to the first and second candidates), this input sample cannot be regarded as one of vowels. That is, it is rejected by the process. It is then passed to the classification process of voiced consonants.

II-9 Classification of Voiced Consonants

As described in section II-7, voiced consonants are detected as one of the following:

1) portion rejected by the vowel recognition process,
2) transient portion, whose duration is longer than 40ms,
3) portion with weak energy and a concave form of an energy contour.

Also, as described in section II-4-1, since the inter-speaker difference among spectra of a voiced consonant is larger than that among vowel spectra, the system must normalize or learn this difference.

Kohda and Saito[43] tested the possibility that a spectrum of a voiced consonant can be estimated from those of the vowels. This method, however, had the following weaknesses:

1) A spectrum was estimated only for the voiced consonants included in ten digits. Thus, the kinds of voiced consonants and contexts were limited.
2) The parameters used for estimation were obtained from utterances spoken by five male adults. These utterances were also employed as test samples. In other words, they did not try to estimate the

voiced consonants of unknown speakers.

These considerations led us to the necessity of a fundamental study on estimation for unknown speakers. Now I explain our method: The system does not need to use time variation of an energy envelop, because we found by preliminary experiment that this variation does not give us significant help in speech perception of voiced consonants (Shimamoto[88]). As in vowel classification, the classification of voiced consonants is carried out by the quadratic discriminant functions. Below is described our fast method for learning a mean vector of an unvoiced consonant.

Let $\bar{y}_l$ be a spectrum common to all speakers for the phoneme l and $\bar{y}_l^S$ be that of the speaker S. The system attempts to estimate the spectrum $\bar{y}_l^S$, by using $\bar{y}_i$ and $\bar{y}_i^S$ (i denotes a vowel) as follows:

$$\hat{y}_l^S = \bar{y}_l + K_{l,i} \cdot (\bar{y}_i^S - \bar{y}_i) \ , \qquad \text{(II-18)}$$

where $K_{l,i}$ is estimated as the (20×20) diagonal matrix expressing the relation between l and i. In practice, the system uses the learned spectrum $\hat{y}_i^S$ mentioned in section II-8-3, instead of the spectrum $\bar{y}_i^S$ because the system cannot know $\bar{y}_i^S$. $K_{l,i}$ for all pairs of phonemes was obtained from the minimum squared error criterion by using the speech materials as mentioned in section II-4, spoken by six out of ten adults. Other four male speakers were regarded as unknown speakers. Experimental results are shown in Table II-13, for every unknown speaker and voiced consonant. To compare with this estimation method, two other estimations are also shown in the table: that were obtained by the use of common spectra on all speakers and that of the personally adjusted spectra (individual spectra). We find from these results that the best estimation is derived from the spectrum of /u/ or /N/. Strictly speaking, the best estimation depends on the kind of voiced consonants. Based on Euclidean distance, Table II-14 shows the recognition results of the voiced consonants, using /u/ and /N/ for the estimation. These results interestingly suggest that the estimation of voiced consonants for unknown speakers is possible by

using the vowel spectra. Hereafter, the system uses the estimation by /N/ for convenience.

Recently, Furui[19] proposed a new method for normalization of voiced consonants. However, its performance was worse than that which uses spectra common to all speakers. I guess that this result is due to neglecting the intensive correlation between a vowel and voiced consonant.

Table II-13 Estimation error of voiced consonant spectrum for all speakers.

(a) classwise estimation error of speaker

estimation method \ speaker	YS	NK	MT	AR	total
Common spectrum for all speakers	420	454	517	505	1896
Estimation by spectrum of /a/	476	419	526	480	1901
Estimation by spectrum of /i/	393	385	518	482	1776
Estimation by spectrum of /u/	404	387	489	473	1753
Estimation by spectrum of /e/	408	589	513	569	2079
Estimation by spectrum of /o/	406	437	477	499	1819
Estimation by spectrum of /N/	383	366	496	501	1746
personally adjusted spectrum*	272	240	378	379	1269

* corresponds to the spectrum of known speaker.

(b) classwise estimation error of voiced consonant

method	m	n	$\widetilde{g}$	b	d	g	r	z
common	214	184	289	233	136	405	200	236
a	204	163	282	236	129	455	195	239
i	189	163	256	238	118	411	188	216
u	181	159	240	226	125	422	188	214
e	249	212	320	255	131	499	188	226
o	198	160	255	254	132	437	186	227
N	153	127	258	230	131	425	192	228
individual	116	92	164	173	88	306	162	168

Table II-14 Recognition results of voiced consonants by estimated spectrum (error number).

estimation method	speaker	m	n	$\widetilde{g}$	b	d	g	r	z	total
common spectrum for all speakers	known	57	89	101	101	49	77	61	20	555
	unknown	58	85	81	57	32	54	42	11	420
estimation by spectrum of /u/	known	57	84	95	85	46	63	59	22	511
	unknown	47	87	74	62	34	49	37	18	408
estimation by spectrum of /N/	known	53	74	94	90	49	67	61	20	508
	unknown	50	62	75	60	32	50	39	17	385
personally adjusted spectrum	known	36	46	74	61	46	52	51	15	381
	unknown*	21	31	48	38	22	39	32	14	245

* corresponds to known speaker.

II-10 Rewriting Rules for a Sequence of Phonemes

A recognized phoneme sequence is modified by rewriting rules. The rewriting rules which we use in our system are divided into following four types: (1) rules for recognizing semi-vowels, (2) rules for classifying silence, (3) phonological rules and (4) rules for correcting phonemes misrecognized because of coarticulation. I will describe these rules: Let F_n, S_n and DUR_n be the first candidate, second candidate and duration of the n-th recognized segment, respectively. Although the degree of reliability is also modified when some of rules are applied, such is not discussed here.

(1) rules for recognizing semi-vowels

i) $(F_n, S_n)=(i,e)$ or (e,i), $DUR_n \leq 50$ms and $F_{n+1}=/a/$, $/u/$ or $/o/$, then $F_n \rightarrow S_n$ and $/y/ \rightarrow F_n$.

ii) If $F_n=/u/$, $F_{n+1}=/o/$, $DUR_n+DUR_{n+1} \leq 60$ms and $F_{n+2}=/a/$, then $/w/ \rightarrow F_n=F_{n+1}$, and $/u/ \rightarrow S_n=S_{n+1}$, and these segments are merged.

If $F_n=/u/$, $S_n=/o/$, $DUR_n \leq 50$ms and $F_{n+1}=/a/$, then $F_n \rightarrow S_n$ and $/u/ \rightarrow F_n$.

(2) rules for classifying silence

In the primary segmentation process, a portion of silence was classified into a pause symbol denoted by /./. In this process, the system tries to divide silence into /./, /-/ and /// in the order of increasing duration. These symbols can correspond to a closure preceding an unvoiced plosive consonant, a choked sound and a pause of phrase or sentence boundary, respectively. The choked sound usually stands only after a short vocalic phoneme and before /p,t,k, s,c/. When F_{n+1} is silence, the rules are follows:

iii) If $DUR_{n+1} \geq 120$ms, then /-/$\rightarrow F$, and /./$\rightarrow S_n$.

iv) If F_n=a vowel, $DUR_n \geq 200$ms and $DUR_{n+1} \geq 200$ms, or F_n=/N/, $DUR_n \geq 150$ms and $DUR_{n+1} \geq 150$ms, then ///$\rightarrow F_{n+1}$, and /-/$\rightarrow S_{n+1}$.

(3) phonological rules

v) If F_n=/d/ and F_{n+1}=/i/ or /u/, then $S_n \rightarrow F_n$, and /d/$\rightarrow S_n$. (In Japanese, syllables such as /ti/,/di/,/tu/,/du/, etc. do not exist.)

vi) If the first candidates of three successive segments are all voiced consonants, then these segments are merged into one, to which the voiced consonant of the original middle segment is assigned. (In Japanese, voiced consonants do not appear in succession to each other.)

vii) If F_n and S_n are voiced consonants and F_{n+1}=silence, then the n-th segment is combined with the preceding one. If S_n=a vowel instead of S_n=a voiced consonant, then F_n and S_n exchange the positions. (In Japanese, a voiced consonant does not appear at the final position in a word.)

(4) rules for correcting phonemes misrecognized because of coarticulation

viii) If $F_n = S_{n+1}$ or $F_{n+2} = S_{n+1}$, and $DUR_{n+1} \leq 40$ms, then the first and second candidates of the (n+1)th segment exchange the positions.

ix) If $F_n \neq$/N/, F_{n+1}=/N/, $F_{n+2} \neq$/N/ and $DUR_{n+1} \leq 40$ms, then F_n and S_n of the (n+1)th segment is exchanged as in rule vii).

An example of the application of the rewriting rules is illustrated in Fig. II-7.

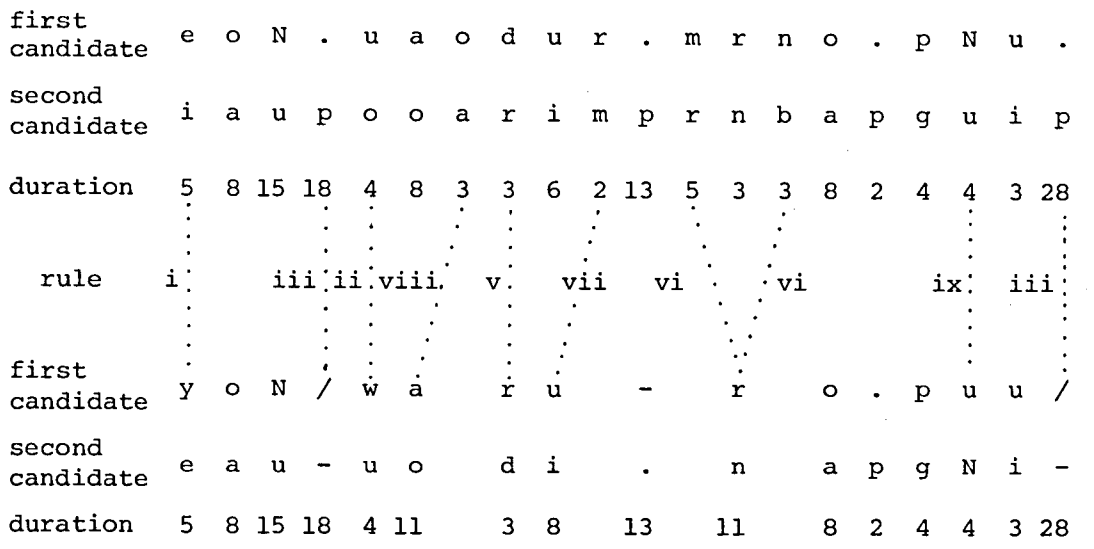


Fig. II-7 An example of the application of rewriting rules.

II-11 Experiment on Phoneme Recognition in Continuous Speech

Before we incorporated it into the total speech understanding system, we tested the phoneme recognition algorithm mentioned above. In this experiment we calculated the average vectors and covariance matrices necessary for the quadratic discriminant functions of vowels, using the 80 sentences of 'Arithmetic Expression' (see Chapter VI) uttered by five speakers. For voiced consonants, we used different materials spoken by ten male adults. The 80 utterances mentioned above were then applied to Bayes' system of discriminant functions.

The resulting confusion matrix for automatic recognition is shown in Table II-15. In the table, only first candidate phonemes are considered to be results of the recognition.

The correct rate of segmentation between voiced and unvoiced was about 97%. The rate of detection of voiced consonants was about 53%, and half of these detection errors were cases of being recognized as

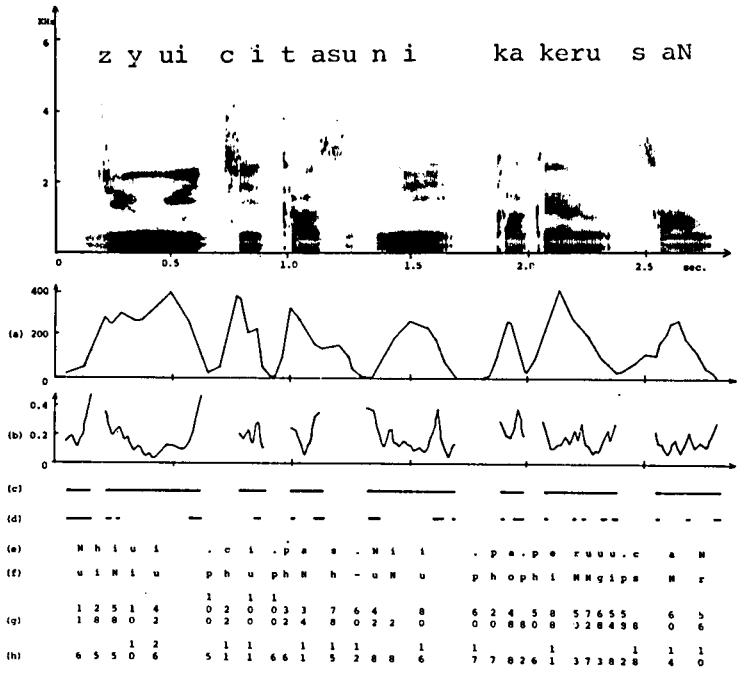
Table II-15 Recognition results of phonemes in arithmetic expressions.

in \ out	a	i	u	e	o	N	y	w	m	n	g̃	b	d	g	r	z	s	c	h	p	.
a	369	4	7	9	14	10						2		3	4				1		
i	1	209	12	9		3							1	2	2					3	
u		18	282	5	4	4						2		12	6						
e	1	1	13	202		4				1			1								
o	8	5	2		162							1			2				1		
N	3	32	15	1		91				2		1	3		1						
y		1	5	4	1		9							3	9						
w	2			4	5	7		13				4		8	5					1	
m		2	5			9			5	2				4	8						
n	3	3	5	2		59				6			3	9	9	2			1		
d		1	5	5		9							13	1	12	4			1	6	
g		3	7	1	1	3						5	1	16	1					5	
r	3	4	20	5	1	28			1			1	10	3	115	1				7	
z			1			5						3	4		2	5		2	11		
s																2	85	5	1	2	1
c																	5	38	1		4
h	1	3								1				2			1	5	69	25	3
p	2					4						5	5	5	4			2	23	272	13

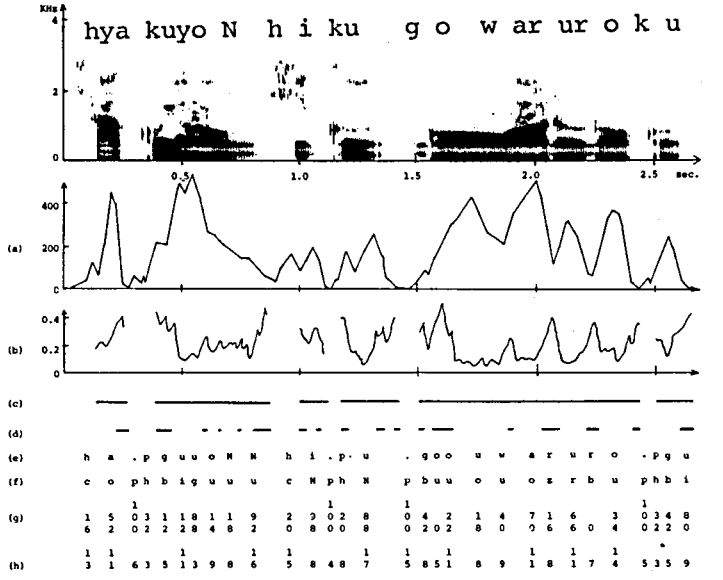
/N/. These influences were slight with respect to the total system performance, as they could easily be recovered in a following process such as word identification.

The averaged score for vowels and the syllabic nasal was about 86%. Those for semi-vowels, voiced consonants and unvoiced consonants were about 30%, 34% and 81%, respectively. Note that the voiced consonants /b/ and /g̃/ were not contained in the vocabulary of 'Arithmetic Expression'.

Examples of the recognition process of the Phoneme Recognizer are illustrated in Fig. II-8. Input sentences "11 + 2 × 3" (zyū ichi tasu ni kakeru san) and "104 - 5 / 6" (hyaku yon hiku go waru roku).



(a) Input: "zyu ichi tasu ni kakeru saN (11+2×3)"



(b) Input: "hyaku yon hiku go waru roku (104-5/6)"

Fig. II-8 Examples of phoneme recognition results
(a):power of input speech, (b):degree of spectral change,
(c):voiced part, (d):transient part, (e):first candidate,
(f):second candidate, (g):reliability(×100), (h):segment duration(×10ms)

For the first candidate, when the value of confidence degree (or reliability) is equivalent to zero, it means that there is no difference between the first candidate and second one. The word "tasu" [=plus] was recognized as "pas" in Fig. II-8(a), because its final vowel /u/ was devocalized.

Conspicuous examples of confusion are the following:

(1) Conspicuous natural confusions are consistent with both the Bhattacharyya distance in Table II-3 and the processing order in phoneme recognition. That is, the confusion occurring from vowels to voiced consonants is rarer than the inverse confusion between them.

(2) In vowel recognition, the confusions between /a/ and /o/, and between /i/ and /u/ are large.

(3) The syllabic nasal /N/ is often misrecognized as /i/ or /u/.

(4) The voiced consonants are often misrecognized as /N/, and are also confused among themselves.

(5) The confusion between the unvoiced plosive /p/ and the aspirated /h/ is large. Also, /p/ is sometimes misrecognized as silence, which implies either omitted or missed.

56 項欠

CHAPTER III

WORD IDENTIFIER

In this chapter, I will describe a new method of word identification. It is based on matching technique of a recognized phoneme sequence against a phoneme sequence given by an entry in a word dictionary. This matching uses a phoneme similarity matrix and Dynamic Programming. Before going to the description of our method, I will overview both the aspects of spoken word recognition research and various types of schemes (developed or to be developed) having their basis on DP matching.

III-1 Introduction

In automatic word recognition, if the whole input pattern is regarded as a point in the pattern space, the recognizer can avoid the problem of coarticulation. It can also use the linguistic information obtained through the word dictionary, i.e., the redundancy of natural language. Therefore, for limited speakers, it is fairly easy to recognize spoken words in a limited vocabulary, together with advances in data processing technology. Recent research in word recognition has focused on the following goals:

(1) enlargement of the vocabulary size.
(2) reduction of the amount of computation and storage.
(3) development of a general word recognition scheme using phoneme recognition.
(4) normalization or learning of speaker differences (see section II-1).

(5) extension to recognition of connected words.

(6) introduction of the error-correcting function, through man-machine communication.

In addition to these points, effective usages of the word dictionary and fast matching algorithms have also been studied as a part of the research. For matching an input pattern against a reference pattern (It is assumed that each of these patterns is expressed by a time series of feature parameters.), I believe that a matching method using DP is one of the best algorithms suited for automatic word recognition. I also believe this because all DP matching algorithms used make good use of the properties of speech sounds such as continuity and regular time order, and they allow for nonlinear warping on the matching between strings with different length.

As shown in Fig. III-1, there can be various types of DP matching algorithms in according to the levels of matched patterns. In the figure, a solid line denotes a time series of some feature parameters (spectrum, formant, LPC, cepstrum, zero crossing measurements, etc.);

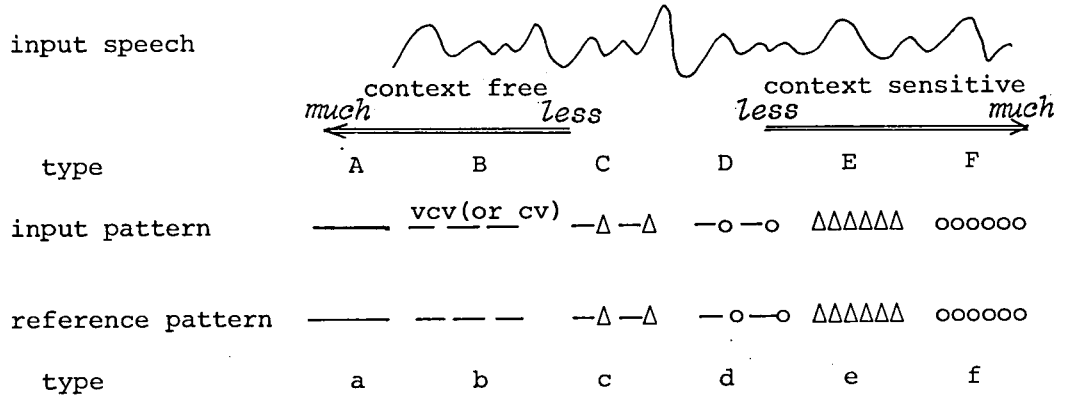


Fig. III-1 Kinds of input pattern and reference pattern on matching.

— a time series expressed by feature parameters every frame
Δ a segment expressed by feature parameters
o a phoneme corresponding to a segment
vcv denotes vowel-consonant-vowel syllable

a triangle, a segment expressed by the feature parameter associated with a phoneme; and a circle, a segment expressed by a phoneme(-like) symbol. Namely, a series of circle symbols denotes a phoneme string.

I will now consider the advantages and disadvantages of these matching algorithms.

(1) A - a type (Velichko and Zagoruyko[94], Sakoe and Chiba[83], Itakura[34])

Speaking from the point of recognition rate, this method is, at present, superior to all others for the spoken words of a single speaker. However, with reference patterns fixed, this method is not as successful for many speakers as it is for one. It is reported that this method is useful for speaker identification(Chiba and Sakoe [11]). These facts imply its dependency on the speakers. Moreover, there is a serious problem that it requires a large amount of memory and computing time. This method has a very simple recognition scheme, and may be called a 'null model'.

(2) A - b type (Nakatsu and Kohda[57])

This type accepts the merits of the *A-a type* and also tries to reduce the amount of memory for reference patterns. However, this gained advantage will be obtained only when the vocabulary size is over 200 words.

(3) B - b type (Nakatsu and Kohda[56])

In comparison with the *A-b type*, this *B-b type* is inferior in recognition rate but superior in computing time. For this type, it is necessary to do segmentation on the basis of VCV-by-VCV (or CV-by-CV) syllables, and such a way is very difficult.

When using these three types, it is very difficult for a system to learn speaker differences quickly.

(4) A - c type, C - c type and D - d type

In the transient parts of input patterns, these types take advantage over the *A-a type*. They also devise to save an amount of computer storage and computation by data-compressing in the stationary

parts of input patterns. However, these types have not been studied yet, because segmentation between the stationary and transient parts is very difficult. (Sato and Fujisaki[84] have recently proposed a method of the *D-d type*, but they did not use the DP algorithm.)

(5) A - e type (Kohda, Hashimoto and Saito[41], Makino, Suzuki and Kido[49], Vintsiuk[96]) and *E - e type* (Hirakawa and Matsuura[28])

Compared with the *A-a type*, these types try more increasingly to solve the problems of memory and computing time. However, still compression of transient parts of patterns is an open problem. Vintsiuk added the duration time of stationary parts, given by grammars, to the description of the *e* type.

(6) F - f type

The author thinks that this type is ideal (see section I-3-1), although it demands accurate recognition of phonemes, especially for continuous speech recognition.

Note that the types *A (a)* or *B (b)* are hardly sensitive to coarticulation or phoneme context (context free), while the types *E (e)* or *F (f)* are much more sensitive by it (context sensitive). As described in section I-3-1, we have used the *F-f type* in our word recognizer.

Our method has the following features:

1) fast algorithm for the identification.
2) normalization of the influence of coarticulation.
3) adoption of a phoneme similarity matrix corresponding to the confusion matrix of phoneme recognition.
4) automatic construction of an effective word dictionary.
5) adoption of dynamic programming with some matching restrictions.
6) capability of word spotting or recognition of connected word.

III-2 Phoneme Similarity Matrix

III-2-1 Similarity between Two Phonemes

Phoneme recognition is performed using statistics of the spectrum (means and covariance matrices in 20-dimensional vectors) for phonemes. Thus, if the Phoneme Recognizer makes mistakes in phoneme recognition, we should consider that such errors are caused by the fact that statistics calculated from an uttered phoneme are very similar to those of a misrecognized phoneme. The errors are generally divided into three kinds: a) substitution, b) insertion, c) omission (see Fig. III-2).

Word matching is fundamentally defined as the process which makes a one-to-one correspondence between each phoneme of a recognized phoneme string and each phoneme of an entry (a word) in the word dictionary. To evaluate a degree of matching between two phonemes, we introduce a concept of similarity between two phonemes.

In general, similarity closely depends on the recognition method used by a phoneme recognizer. It is desirable that the similarity be derived from the confusion matrix made as results of phoneme recognition. However we cannot positively use confusion matrix because a reliable one must be constructed from the results based on a lot of speech data.

As described in section II-4-1, the Bhattacharyya distance is closely related to the confusion matrix constructed from the results of phoneme recognition based on Bayes' rule. We calculate the similarity $S(i,j)$ for a pair of phonemes (i and j) by the following equation (refer to Table II-2):

$$S(i,j) = 100 - 8 \times B(i,j) \qquad \text{(III-1)}$$

If i belongs to the set (a,i,u,e,o,N) and j to the set (m,n,$\tilde{g}$,b,d,g,r, z), $S(i,j)$ calculated by Eq. (III-1) is decreased by 5. On the other hand, if i belongs to the second set or equals to /y/ or /w/ and j is in the first set, $S(i,j)$ is increased by 5. Such modifications of $S(i,j)$ are to compensate for the unsymmetric performance of the Phoneme Recognizer. If either i or j is unvoiced, $S(i,j)$ is derived from the confusion matrix (Table II-15) and referred to the

distinctive features (see Table III-4). The resulting similarity matrix is shown in Table III-1. The column "in" denotes phonemes in the lexicon, and the column "out", recognized phonemes. In this table, the pseudo phoneme /*/ is treated as an unvoiced plosive, except that it is not associated with the silence group.

Table III-1 Phoneme similarity matrix

in \ out	a	i	u	e	o	N	y	w	m	n	g~	b	d	g	r	z	s	c	h	p	t	k	.	-	/
a	100	36	51	69	75	65	70	82	51	55	58	50	51	40	63	36	5	5	70	5	5	5	5	5	5
i	36	100	83	73	56	85	82	40	72	74	85	79	76	77	75	69	5	30	70	50	50	50	5	5	5
u	51	83	100	74	80	89	78	70	81	83	89	88	81	79	82	72	5	5	5	5	5	5	5	5	5
e	69	73	74	100	69	73	92	61	59	69	74	67	72	63	78	62	5	5	5	5	5	5	5	5	5
o	75	56	80	69	100	80	70	91	64	65	75	75	66	60	74	50	5	5	5	5	5	5	5	5	5
N	65	85	89	73	80	100	75	72	84	85	84	81	74	69	78	69	5	5	5	5	5	5	5	5	5
y	75	87	83	95	75	80	100	66	68	73	82	76	77	66	87	67	5	30	50	50	50	50	5	5	5
w	87	45	75	66	96	77	66	100	62	61	71	70	61	50	75	38	5	5	5	50	50	50	5	5	5
m	61	82	91	69	74	94	68	62	100	92	86	85	75	68	83	55	5	5	70	5	5	5	5	5	5
n	65	84	93	79	74	95	73	61	92	100	88	86	84	71	87	69	5	5	70	5	5	5	5	5	5
g~	68	95	99	84	85	94	82	71	86	88	100	93	89	87	89	79	5	30	70	85	85	85	30	5	5
b	60	89	98	77	85	91	76	70	85	86	93	100	90	84	89	72	5	50	50	85	85	85	40	5	5
d	61	86	91	82	76	84	77	61	75	84	89	90	100	85	88	87	5	50	50	85	85	85	40	5	5
g	50	87	89	73	70	79	66	50	68	71	87	84	85	100	75	78	5	50	70	85	85	85	40	5	5
r	73	87	92	88	84	88	87	75	83	87	89	89	88	75	100	72	5	5	50	50	50	50	5	5	5
z	45	79	82	72	60	69	67	38	55	69	79	72	87	78	72	100	85	75	75	50	50	50	5	5	5
s	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	80	100	90	85	60	60	60	40	5	5
c	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	70	90	100	85	85	85	85	60	5	5
h	70	60	5	5	5	5	5	70	70	70	50	50	70	60	70	85	85	100	90	90	90	70	5	5	5
p	50	40	5	5	5	50	5	5	5	5	85	85	85	85	50	40	60	85	90	100	100	100	90	90	90
t	50	40	5	5	5	50	5	5	5	5	85	85	85	85	50	40	60	85	90	100	100	100	90	90	90
k	50	40	5	5	5	50	5	5	5	5	85	85	85	85	50	40	60	85	90	100	100	100	90	90	90
.	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	60	85	90	90	90	100	20	5
-	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	60	85	90	90	90	100	100	20
/	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	60	85	90	90	90	100	100	100
*	5	5	5	5	5	5	5	5	5	5	85	85	85	85	5	5	60	85	90	100	100	100	5	5	5

III-2-2 Adaptation of Phoneme Similarity to a Speaker

Strictly speaking, because of existing speaker differences and his dialect, a similarity matrix does depend on the speaker. Therefore, we adapt this matrix to a given speaker on the basis of spectra learned from the Phoneme Recognizer.

Now, let $\bar{d}(i,j)$ be the common cityblock distance between i and j to all speakers and $\hat{d}^{S}(i,j)$ between i and j to the personally adjusted speaker. The similarity $S^{S}(i,j)$ for the speaker S is modified by the following equation:

$$S^S(i,j) = S(i,j) + k\left\{1 - \frac{\hat{d}^S(i,j)}{\bar{d}(i,j)}\right\}, \qquad \text{(III-2)}$$

where k is a constant. In short, if $\hat{d}^S(i,j)$ is smaller than $\bar{d}(i,j)$, the similarity is increased. Otherwise, it is decreased. The resulting similarity is again bounded from 0 to 100.

III-3 Word Dictionary

III-3-1 Phonemic Representation in an Entry

All words which are used in a task are preregistrated in the word dictionary. In order to reduce matching time and computer storage, the dictionary describes each word as a string of phonemic symbols in only one way. Some phonemes in a word are often influenced by phoneme environments, while other phonemes are sometimes devocalized. Because of this influence, such phonemes are often omitted or misrecognized as other phonemes. By introducing the sub-phoneme '*k*' in addition to the main-phoneme '*I*', we denote these situations in the dictionary by *I*/*k*(c). This notation means that the phoneme '*I*' can be replaced by the phoneme '*k*', where c ($0 \leq c \leq 1.0$) means the weight of the sub-phoneme '*k*'. Both phonemes are equally treated if it is 1.0, and the sub-phoneme is neglected if it is 0. By this description, we can represent an optional phoneme, i.e., one that is omissible or addible. For example, if /i/ of 'rei'[=zero] is optional, the entry for "rei" can be represented by 'r e i/e(1.0)'. This representation is essentially equal to registering two entries such as 'r e' and 'r e i'.

Table III-2 shows a part of the word dictionary. The special symbols (+ and -) indicate changes of restrictions for DP matching (see section III-4-2). The maximum and minimum durations indicate the range of duration time required for uttering a word. These representations for given words are automatically constituted by the constructing rules of the word dictionary.

Table III-2 Examples of entries in the Word Dictionary

word	sym-bol	phonemic representation	duration	
			Dmax	Dmin
ichi	1	i ./c(1.0) c $\overset{+-}{i}$/c(1.0)	350ms	100ms
ni	2	n i	300	100
san	3	s a N	550	200
yon	4	y/$\tilde{g}$(0.95) o N	450	150
go	5	g o	300	100
roku	6	r o ./k(0.95) k $\overset{+-}{u}$/*(1.0)	450	100
nana	7	n a/N(0.85) n/a(0.85) $\overset{+}{a}$/N(0.85)	550	200
hachi	8	h a/N(0.85) ./c(1.0) c $\overset{+-}{i}$/c(1.0)	500	150
kyu	9	//c(0.95) k/c(0.95) y/u(0.95) u	500	200
rei	0	r/p(0.85) e i/e(0.95)	400	100

III-3-2 Rules for Constructing a Word Dictionary

We can describe a given word uniquely by writing Japanese in Roman letters. This orthographic representation of a word is transformed into a lexical representation by the following rewriting rules. These rules are divided into either phonological rules of Japanese or rewriting rules depending on our phoneme recognizer.

Let V, VC, UC, and ϕ be a vowel, voiced consonant, unvoiced consonant, and word boundary, respectively. The symbol '$\overline{\ }$' denotes an operator of the complement for a set of phonemes.

(A) Phonological Rules for Japanese

(1) affricate and unvoiced plosive

$\overline{\phi}\ c \rightarrow \overline{\phi}\ ./c(1.0)\ c$ i c i → i ./c(1.0) c i

$\overline{\phi}\begin{Bmatrix} p \\ t \\ k \end{Bmatrix} \rightarrow \overline{\phi}\ ./\begin{Bmatrix} p \\ t \\ k \end{Bmatrix}(0.95)\begin{Bmatrix} p \\ t \\ k \end{Bmatrix}$ r o k u → r o ./k(0.95) k u

$\phi \begin{bmatrix} c \\ p \\ t \\ k \end{bmatrix} \rightarrow // \begin{bmatrix} c \\ p \\ t \\ k \end{bmatrix} (0.95) \begin{bmatrix} c \\ p \\ t \\ k \end{bmatrix}$ k y u → //k(0.95) k y u

(2) vowel devocalization

$UC_1 \begin{bmatrix} i \\ u \end{bmatrix} \overline{UC}_2 \rightarrow UC_1 \begin{bmatrix} i \\ u \end{bmatrix} / UC_1 (0.95) \overline{UC}_2$; h i d a r i → h i/h(0.95) d a r i

$UC_1 \begin{bmatrix} i \\ u \end{bmatrix} \begin{bmatrix} UC_2 \\ \phi 2 \end{bmatrix} \rightarrow UC_1 \begin{bmatrix} i \\ u \end{bmatrix} / UC_1 (1.0) \begin{bmatrix} UC_2 \\ \phi 2 \end{bmatrix}$; i c i → i c i/c(1.0) i

(3) nasalization

$\begin{bmatrix} m \\ n \\ \tilde{g} \end{bmatrix} V_1 \begin{bmatrix} b \\ d \\ g \end{bmatrix} \rightarrow \begin{bmatrix} m \\ n \\ \tilde{g} \end{bmatrix} V_1 \begin{bmatrix} b \\ d \\ g \end{bmatrix} / \begin{bmatrix} m \\ n \\ \tilde{g} \end{bmatrix} (0.85)$ m a d e → m a d/n(0.85) e

(4) syllabic nasal

$V_1 N \begin{bmatrix} a \\ u \\ o \\ w \end{bmatrix} \rightarrow V_1 N/u(0.9) \begin{bmatrix} a \\ u \\ o \\ w \end{bmatrix}$ i N u → i N/u(0.9) u

$V_1 N \begin{bmatrix} i \\ e \\ y \end{bmatrix} \rightarrow V_1 N/i(0.9) \begin{bmatrix} i \\ e \\ y \end{bmatrix}$ a N e → a N/i(0.9) e

$V_1 N \begin{bmatrix} p \\ b \\ m \end{bmatrix} \rightarrow V_1 N/m(0.95) \begin{bmatrix} p \\ b \\ m \end{bmatrix}$ a N m a → a N/m(0.95) m a

$V_1 N \begin{bmatrix} t \\ d \\ n \end{bmatrix} \rightarrow V_1 N/n(0.95) \begin{bmatrix} t \\ d \\ n \end{bmatrix}$ e N d a i → e N/n(0.95) d a i

(5) velar nasal

$\overline{\phi}$ g → $\overline{\phi}$ $\tilde{g}$/g(0.95) m i g i → m i $\tilde{g}$/g(0.95) i

(6) vocalization

$\overline{\phi} \begin{bmatrix} t \\ k \end{bmatrix} V_1 \rightarrow \overline{\phi} \begin{bmatrix} t \\ k \end{bmatrix} / \begin{bmatrix} d \\ g \end{bmatrix} (0.85) V_1$ h i k u → h i k/g(0.85) u

(7) devocalization

$\phi \begin{bmatrix} b \\ d \\ g \\ r \end{bmatrix} \rightarrow \phi \begin{bmatrix} b \\ d \\ g \\ r \end{bmatrix} / \begin{bmatrix} p \\ t \\ k \\ p \end{bmatrix} (0.85)$ r e i → r/p(0.85) e i

$$\begin{bmatrix} i \\ u \end{bmatrix} \begin{bmatrix} b \\ d \\ g \end{bmatrix} \rightarrow \begin{bmatrix} i \\ u \end{bmatrix} \begin{bmatrix} b \\ d \\ g \end{bmatrix} / \begin{bmatrix} p \\ t \\ k \end{bmatrix} (0.85) \qquad h\ \underline{i\ d}\ a\ r\ i \rightarrow h\ \underline{i\ d/t(0.85)}\ a\ r\ i$$

(8) $e\ i \rightarrow e\ i/e(0.95)$ $\qquad r\ \underline{e\ i} \rightarrow r\ \underline{e\ i/e(0.95)}$

(9) $\phi\ u\ m \begin{bmatrix} u \\ e \\ o \end{bmatrix} \rightarrow \phi\ u/m(0.9)\ m \begin{bmatrix} u \\ e \\ o \end{bmatrix}$ $\qquad \underline{u\ m\ e} \rightarrow \underline{u/m(0.9)\ m\ e}$

(B) Rewriting Rules Depending on Our Phoneme Recognizer

(10) semi-vowel

$$V_1 \begin{bmatrix} y \\ w \end{bmatrix} \rightarrow V_1 \begin{bmatrix} y \\ w \end{bmatrix} / V_1 (0.9) \qquad \underline{i\ w}\ a \rightarrow \underline{i\ w/i(0.9)}\ a$$

$$\phi \begin{bmatrix} y \\ w \end{bmatrix} V_1 \rightarrow \phi \begin{bmatrix} y \\ w \end{bmatrix} / V_1 (0.9)\ V_1 \qquad \underline{y\ o}\ N \rightarrow \underline{y/o(0.9)\ o}\ N$$

(11) contracted sound (palatalization of consonants)

$y\ V_1 \rightarrow y/V_1(0.95)\ V_1$ $\qquad k\ \underline{y\ u} \rightarrow k\ \underline{y/u(0.95)\ u}$

$z\ y \rightarrow z\ h/y(0.95)$ $\qquad \underline{z\ y}\ u \rightarrow \underline{z\ h/y(0.95)}\ u$

(12) $\phi\ z \rightarrow \phi\ z/d(0.9)$ $\qquad \underline{z}\ y\ u \rightarrow \underline{z/d(0.9)}\ y\ u$

(13) $s \begin{bmatrix} i \\ y \end{bmatrix} \rightarrow s/h(0.9) \begin{bmatrix} i \\ y \end{bmatrix}$ $\qquad i\ \underline{s\ i} \rightarrow i\ \underline{s/h(0.9)\ i}$

(14) $V_1\ VC\ V_1 \rightarrow V_1 VC/V_1(0.85)\ \overset{+}{V}_1$ $\qquad n\ \underline{a\ n\ a} \rightarrow n\ \underline{a\ n/a(0.85)\ \overset{+}{a}}$

(15) $V_1\ h\ V_2 \rightarrow V_1\ h/V_2(0.9)\ V_2$ $\qquad g\ \underline{o\ h\ a}\ N \rightarrow g\ \underline{o\ h/a(0.9)\ a}\ N$

However the rules (3), (6) and (7) are already under consideration by the similarity matrix; thus in practice they are not applied to the construction of the word dictionary. The rules (10), (14) and (15) compensate for omission errors caused by the Phoneme Recognizer.

The range of duration time for a word is determined by using the duration time of each phoneme. This is shown in Table III-3, which is empirically constructed from a large amount of speech data. Minimum duration time (Dmin) is obtained by summing up the duration times for phonemes in the lexical representation of word, multiplying the resultant value by the factor 1.1, and then cutting down the fractional part to less than 50ms. For example, we have a minimum of 200ms for the word "saN", as summed duration time is 220ms (=80+60+80),

the multiplied duration time is 242ms (=220×1.1) and the cut-down duration time is 200ms. On the other hand, maximum duration time (Dmax) is obtained in the same way, except for multiplying by the factor 0.9 and counting the fractional part to more than 50ms.

After these constructing rules are applied, sometimes some entries in the word dictionary are modified on the basis of experimental evidences.

Table III-3 Constructing rules of duration time

phoneme	duration		notes
	Dmin	Dmax	
a,i,u,e,o, N	60ms	150ms	
	0	70	may be devocalized
	80	200	final position of a word
	150	300	elongated
y,w,r	30	60	
m,n,$\tilde{g}$,z,c	40	100	
b,d,g	40	80	
s	80	160	
h	50	120	
p,t,k	30	80	
.	20	60	closure of plosive (0 to 40 for /c/)
-	150	250	choked sound

III-4 Recognition Method of Spoken Words

III-4-1 Outline of Spoken Words Recognition Method

An output of the Phoneme Recognizer is a sequence of segments, each consisting of the first and second candidates of phonemes, the degree of confidence of the first candidate, and the segment duration. The first three constituents of the j-th segment in a sequence will

be denoted by J, l and p($0 \leq p \leq 1.0$), respectively. An element by which a word in the word dictionary is described is either a main-phoneme or a sub-phoneme plus a weighting factor. The constituents of the i-th element of a lexical entry will be denoted by I, k and c, respectively. In order to match a portion of a segment string against a word, we must first define the similarity between a segment and an element in the entry. This similarity is defined by the following operation:

$$S(I,k,c;J,l,p) = \max\begin{Bmatrix} S(I,J) \\ c \times S(k,J) \\ p \times (I,J) + (1-p) \times S(I,l) \\ p \times c \times S(k,J) + (1-p) \times c \times S(k,l) \end{Bmatrix} . \qquad \text{(III-3)}$$

where max is an operator which selects the largest value among the values in the parentheses, and $S(*,*)$ is the similarity between phonemes given in Table III-1. I simply denote $S(I,k,c;J,l,p)$ as $S_0(i,j)$. By using this measure, we can evaluate the matching score between an element string of a word and an input string (by which I mean some portion of a segment string).

If an element in a word associates with a segment of an input string, the similarity is directly calculated by Eq. (III-3). If it associates with plural segments, the similarity for these associations is the arithmetic mean of these associated measures. The similarity for a given word is obtained from the arithmetic mean of the similarities for all elements. Thus, we evaluate the similarity of a word for all possible associations between an input string and an element string; we then say that the similarity with the highest score is the likelihood for that word. An input string is recognized as the word which gained the highest likelihood. This calculation is carried out efficiently by the use of dynamic programming. Fig. III-2 illustrates the matching procedure. An association between two strings corresponds to a path (or route) on the lattice plane. The horizontal axis denotes an input string of recognized segments, and the vertical axis denotes a string of elements of a lexical entry.

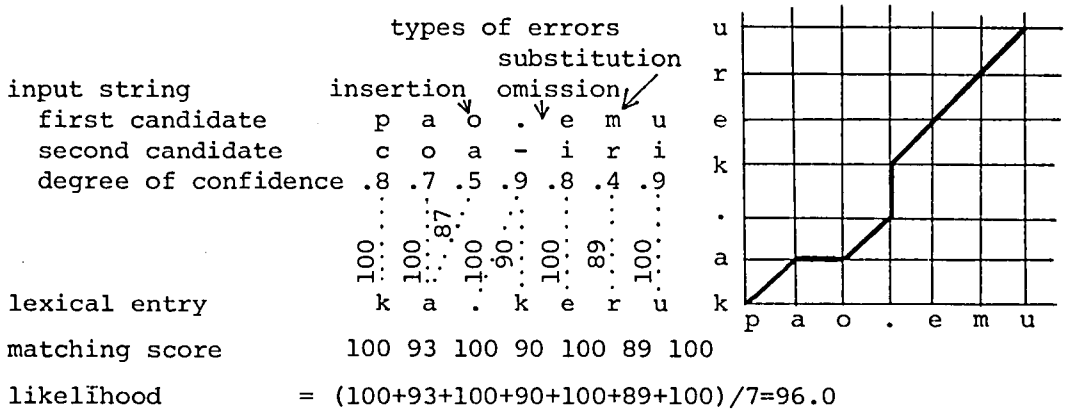


Fig. III-2 Graphic representation of word matching and matching score

III-4-2 Restrictions on Word Matching

Abe[1] recently proposed 'Penalty Automaton' for recognition of time-continuous or space-continuous patterns. This method confronts the difficult problem of constructing an automaton. However, if his method becomes more flexible, it will be similar to the method proposed here. We consider that variations of patterns can be processed to a certain extent by using a similarity matrix. However, when a subtle variation happens to be a significant feature for a given pattern, our method mentioned above does not take such a case into consideration. Therefore, we make the following restrictions with respect to the matching between an input string and an element string. These could be regarded as reasonable restrictions, judging from the performance of the Phoneme Recognizer.

(1) Except for elements marked with the symbol (-), a vowel and the syllabic nasal in the word dictionary can be associated with three or less segments in a recognized phoneme string.

(2) A constant can be associated with two or less segments.

(3) Three or more successive elements cannot be associated with one segment except when there is an element marked with the symbol (+).

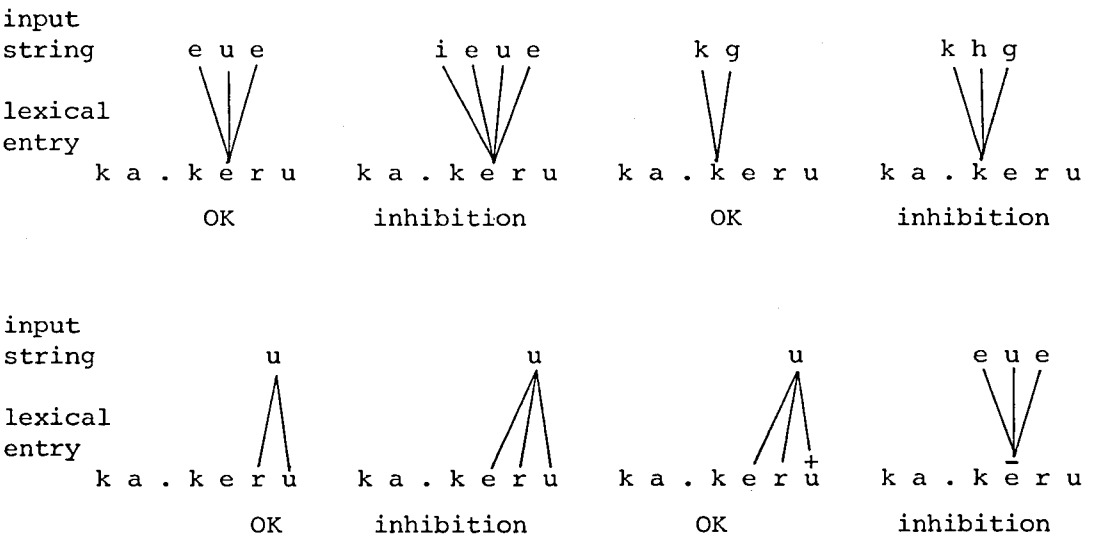


Fig. III-3 Examples of restrictions on the matching.

Examples of these restrictions are illustrated in Fig. III-3.

(4) When the total duration time of three successive segments is beyond 250ms, a vowel element (except for an elongated vowel) does not have to be associated with the three segments.

(5) When the duration time of one segment is not beyond 100ms, an elongated vowel element cannot be associated with only this segment.

(6) If a word matching is performed outside the range of the duration time specified by the lexical entry, the matching score is decreased by the weighting function. This is shown in Fig. III-4, where Dmin and Dmax denote the minimum and maximum duration time, respectively.

(7) Although a word has different time structures every time it is

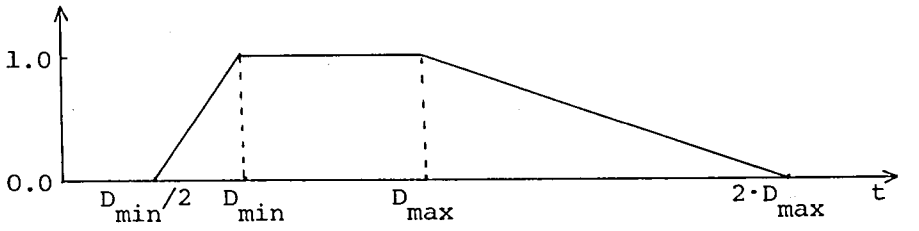


Fig. III-4 Weighting function on duration time

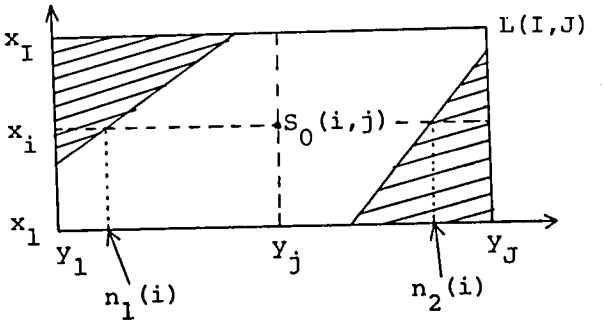


Fig. III-5 Matched window on the lattice plane.

pronounced, it has almost the same segment (or phoneme) structure. That is, the rules mentioned above are micro-scopic rules for local matching, but for global matching, macro-scopic rules are necessary. Therefore, the matching domain on the lattice plane can be restricted as shown in Fig. III-5. In the figure, $n_1(i) \leq j \leq n_2(i)$ for all i, where $n_1(i)$ is set to $[i/2]$ and $n_2(i)$ is set to $\max(i+5, 2\times i-5)$.

(8) Of course, the possible routes on the lattice plane are limited: the route is monotonous, namely, only the paths from left to right along the horizontal, from bottom to top along the vertical, from left-bottom to right-top.

III-4-3 Normalization of Coarticulation

Speech sounds which are produced through the vocal apparatuses are influenced by contexts, as the apparatuses move continuously. This phenomenon is called coarticulation, and it makes automatic phoneme recognition very difficult. In this section, we propose a new technique which normalizes coarticulation in the word recognition stage. When we use the matching algorithm as described in section III-4-1, the similarity between an element x_i in a lexical entry and three successive segments in an input string $\{y_{j-1}, y_j, y_{j+1}\}$ is calculated by the following equation:

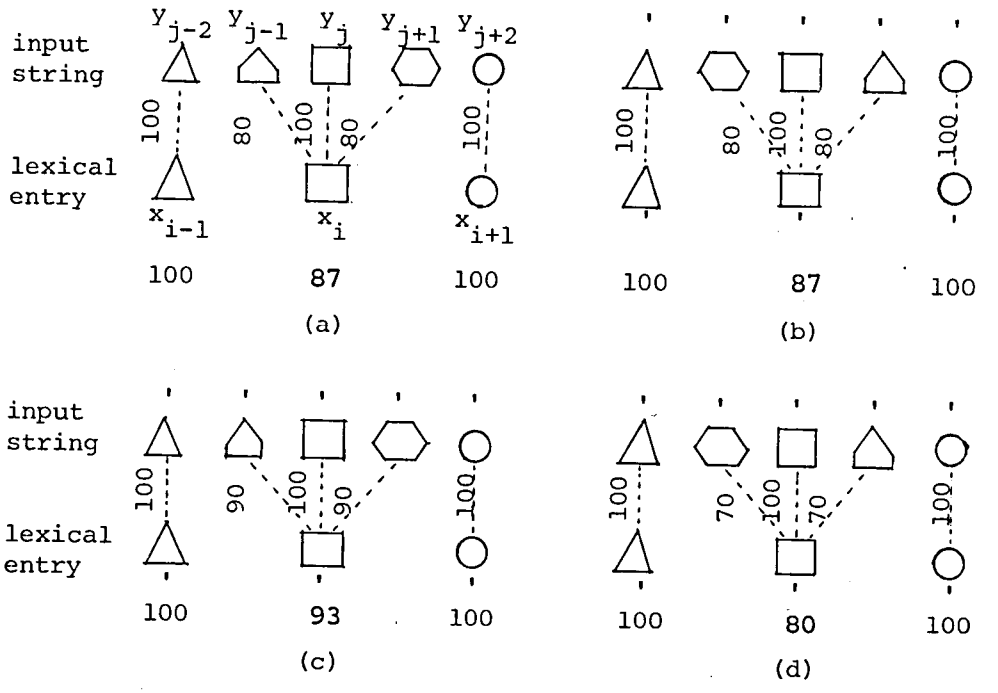


Fig. III-6 Graphic model of normalization of coarticulation.

(a) and (b) : before normalization
(c) and (d) : after normalization

where, we assume that $S(\square,\triangle)=S(\square,\bigcirc)=S(\triangle,\triangle)=S(\bigcirc,\bigcirc)=80$, $S(\triangle,\square)=S(\bigcirc,\square)=60$, $S(\triangle,\bigcirc)=S(\square,\bigcirc)=40$, $k=0.5$.

$$S(x_i;y_{j-1},y_j,y_{j+1}) = \{S_0(i,j-1) + S_0(i,j) + S_0(i,j+1)\}/3. \quad \text{(III-4)}$$

Now let us consider matched combinations such as illustrated in Fig. III-6. For the cases (a) and (b) in the figure, the equation (III-4) gives the same similarity. But we realize that obviously the association of (a) is more natural than that of (b), and so this consideration leads us to improve the matching algorithm. In Fig. III-6, if we regard the association between x_i and $\{y_{j-1},y_j,y_{j+1}\}$ as proper, we can assume that y_{j-1} or y_{j+1} is a transient segment between x_{i-1} and x_i, or x_i and x_{i+1}, respectively. In this case, the following inequalities are expected to be satisfied, because the transient segment would represent an intermediate phonemic category between two successive phonemes.

$$S_0(i-1,j-1) > S_0(i-1,\dot{i})$$
$$S_0(i+1,j+1) > S_0(i+1,\dot{i}) \quad , \qquad \text{(III-5)}$$

where both a and b in $S_0(a,\dot{b})$ denote the a-th and b-th elements in the lexical entry. On the other hand, if that association is regarded as improper, the inequalities will not be satisfied in all the cases. We consider that normalization of coarticulation can be performed by modification of the similarity. This modification is defined as follows:

$$\tilde{S}_0(i,j-1) = S_0(i,j-1) + K\{S_0(i-1,j-1) - S_0(i-1,\dot{i})\}$$
$$\tilde{S}_0(i,j+1) = S_0(i,j+1) + K\{S_0(i+1,j+1) - S_0(i+1,\dot{i})\}, \qquad \text{(III-6)}$$

where K is a constant value. Finally, the association between x_i and $\{y_{j-1},y_j,y_{j+1}\}$ is evaluated by the following equation.

$$S(x_i;y_{j-1},y_j,y_{j+1}) = \{\tilde{S}_0(i,j-1) + S_0(i,j) + \tilde{S}_0(i,j+1)\}/3 \qquad \text{(III-7)}$$

Figs. III-6(c) and (d) illustrate applications of this algorithm for the case of K=1/2. Moreover, when the element x_i in a lexical entry associates with two successive segments $\{y_{j-1},y_j\}$in an input string, this association is evaluated by the following equation:

$$S(x_i;y_{j-1},y_j) = \{S_0(i,j-1) + K[S_0(i-1,j-1) - S_0(i-1,\dot{i})]$$
$$+ S_0(i,j) + K[S_0(i+1,j) - S_0(i+1,\dot{i})]\}/2 \qquad \text{(III-8)}$$

III-4-4 Matching Algorithm by Dynamic Programming

The matching algorithm mentioned above is efficiently performed by using dynamic programming. Now, we consider how to calculate the likelihood for a given word. Let $L(i,j)$ be the highest cumulative similarity (or score), when considering the i-th element of the lexical entry for this word and the j-th segment of a recognized phoneme string. In other words, $L(i,j)$ is determined by evaluating

all possible paths from the point (1,1) to the point (i,j) on the lattice plane. When the i-th element is a vowel or the syllabic nasal, $L(i,j)$ is calculated successively by the following operation:

$$L(i,j) = \max\{L_1(i,j), L_2(i,j), L_3(i,j), L_4(i,j), L_5(i,j), L_6(i,j)\}, \quad \text{(III-9)}$$

where

$$\begin{aligned}
L_1(i,j) &= L^*(i-1,j) + S_0(i,j) \\
L_2(i,j) &= L(i-1,j-1) + S_0(i,j) \\
L_3(i,j) &= L^*(i-1,j-1) + \{\tilde{S}_0(i,j-1) + \tilde{S}_0(i,j)\}/2 \\
L_4(i,j) &= L(i-1,j-2) + \{\tilde{S}_0(i,j-1) + \tilde{S}_0(i,j)\}/2 \\
L_5(i,j) &= L^*(i-1,j-2) + \{\tilde{S}_0(i,j-2) + S_0(i,j-1) + \tilde{S}_0(i,j)\}/3 \\
L_6(i,j) &= L(i-1,j-3) + \{\tilde{S}_0(i,j-2) + S_0(i,j-1) + \tilde{S}_0(i,j)\}/3
\end{aligned} \quad \text{(III-10)}$$

If the i-th element is marked with the special symbol (+), $L^*(i,j) = L(i,j)$; otherwise $L^*(i,j) = \max\{L_2(i,j), L_3(i,j), L_4(i,j), L_5(i,j), L_6(i,j)\}$. This selection of $L^*(i,j)$ corresponds to the restriction (3) in matching described in section III-4-2. The boundary (or initial) conditions are the following:

$$\begin{aligned}
L(1,j) &= 0 \qquad : j \geq 4 \\
L(1,1) &= S_0(1,1) \\
L(1,2) &= \{\tilde{S}_0(1,1) + \tilde{S}_0(1,2)\}/2 \\
L(1,3) &= \{\tilde{S}_0(1,1) + S_0(1,2) + \tilde{S}_0(1,3)\}/3
\end{aligned} \quad \text{(III-11)}$$

When the i-th element is a consonant, silence or an element with the special symbol (-), $L(i,j)$ is calculated by the following equation:

$$L(i,j) = \max\{L_1(i,j), L_2(i,j), L_3(i,j), L_4(i,j)\} \quad \text{(III-12)}$$

$$L(1,3) = 0$$

L_1, L_2, ... and L_6 correspond to all possible routes on the plane, that is, r_1, r_2, ... and r_6 in Fig. III-7, respectively.

We introduced a reject function to reduce the matching time. The system rejects an entry if its cumulative likelihood does not exceed pre-set thresholds at some steps during the match operation.

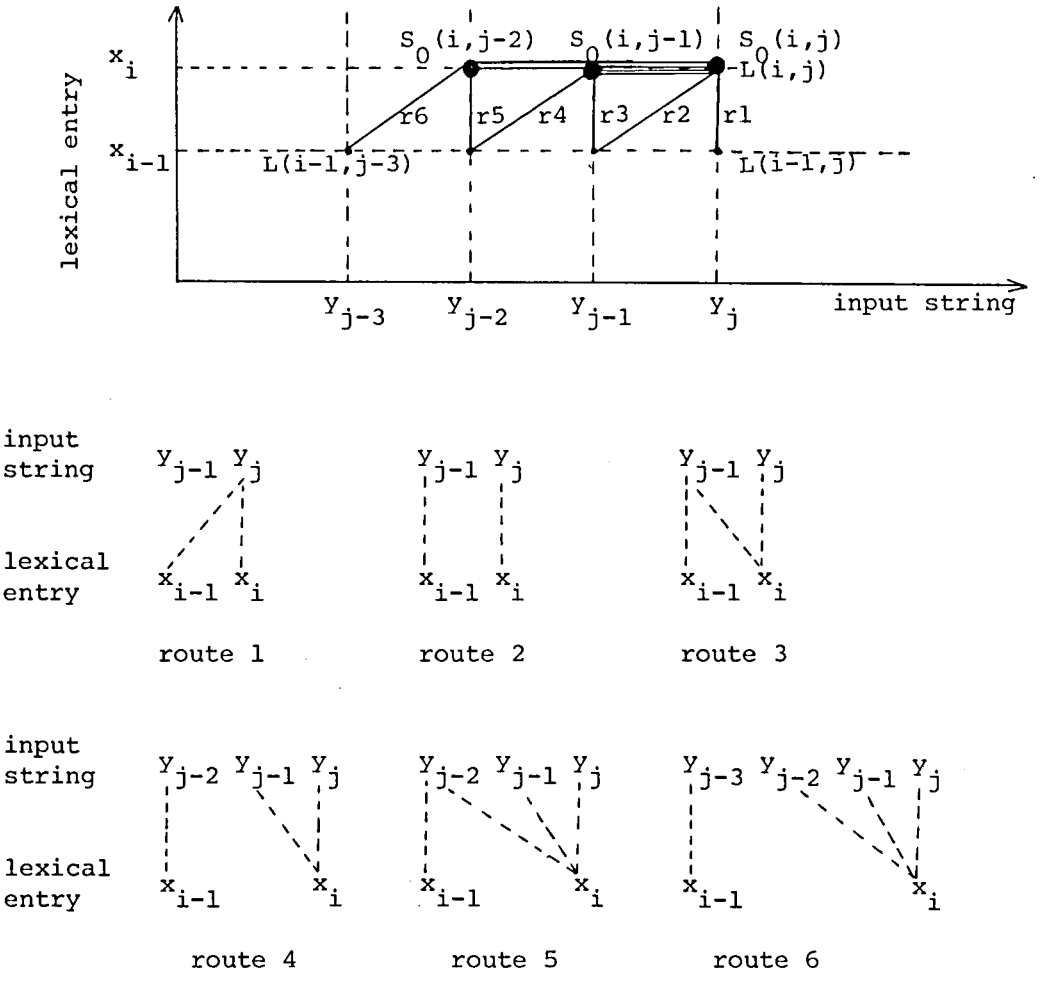


Fig. III-7 Kinds of possible matching routes on lattice plane.
route 1~6 for vowels and the syllabic nasal.
route 1~4 for other phonemes.

If the numbers of elements in a lexical entry and input string are i_0 and j_0, respectively, the likelihood of this word is calculated for isolated words recognition as $L(i_0,j_0)/i_0$. For word identification (or recognition) in continuous speech, the likelihood is calculated as $\max_j L(i_0,j)/i_0$. To avoid any identification error, the best j_1 and 2nd best j_2 are sought over all j associated with i_0. Thus, the Word Identifier generates two identified locations in a recognized string for a given word. In the next section, this procedure will be described in detail.

III-5 Augmented Matching Methods for Continuous Speech

III-5-1 Sequential Matching Method

In section III-4, I described the method of recognition for isolated words. The method described in this section is for word recognition in continuous speech.

Recognition of continuously uttered words is much more difficult than that of words uttered in isolation, since word boundaries become ambiguous in the former. This type of ambiguity makes it very difficult to determine where to start the matching between a phoneme string of a lexical entry and a recognized phoneme string and where to terminate. This difficulty can be explained in Fig. III-8. The last phoneme /N/ of the preceding word "saN"[=3, three] can be associated with only /n/ or with /n/ and /r/ of the input string. However, the first of the two associations will be selected by our matching method,because in fact S(N,n) is larger than S(N;n,r). It is, however, clear that the last phoneme /N/ has been mis-segmented and misrecognized as /n/ and /r/. In addition, the first phoneme /w/ of the next word to be identified as "waru"[divide] should not be associated with the inserted phoneme /r/. Furthermore, the initial segment to which the first element of the lexical entry corresponds is ambiguous in the recognized phoneme string. To overcome these difficulties, we combine the last element (main-phoneme) of the preceding word with the next word to be identified. We attempt to

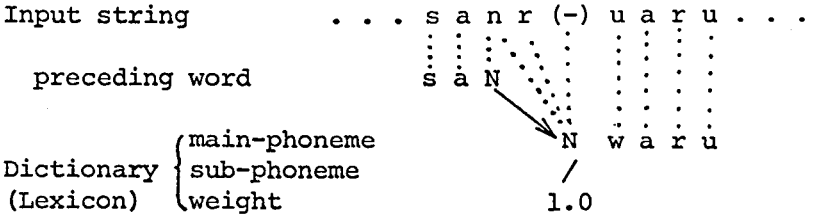


Fig. III-8 Sequential matching method

( ) means an optional phoneme.

identify this new phoneme string "Nwaru", from the segment previously associated with the last element of the preceding word in the recognized phoneme string. Besides we add a sub-phoneme /// for the last element of the preceding word, since there may exist a pause (or word boundary) between two successive words.

The authors will call this method a 'sequential matching method' (one end-point free method), the formal version of which is shown in Fig. III-9. The last phoneme x'_I, of the preceding word is assumed to have been associated with the segment y_r. The concatenation x'_I, x_1 x_2 ... x_I of the element x'_I, and the next word x_1 x_2 ... x_I is thus associated with the sub-string starting at the segment y_r. This association is then terminated at the segment y_{t_1}, such that t_1 satisfies the following inequality:

$$L(I,t_1) \geqq L(I,t),$$

where $r+[(I+1)/2] \leqq t \leqq \min(r+I+5, r+2I-4)$, and $L(I,t_1)/(I+1)$ is the likelihood for this word. To avoid any identification error, a second best match $\{y_r, y_{r+1}, \ldots y_{t_2}\}$ is sought in addition to the best one such

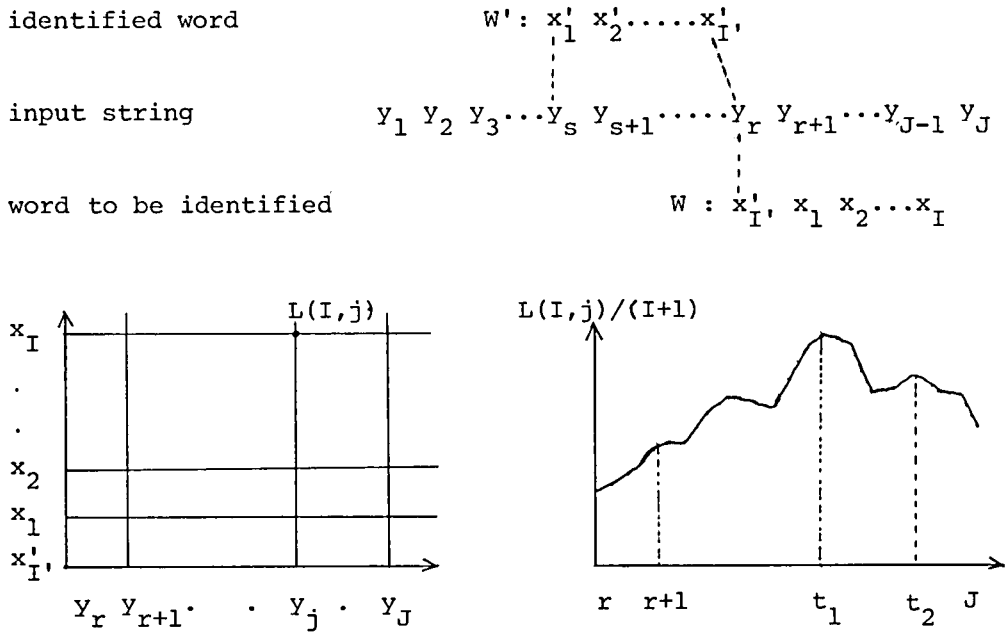


Fig. III-9 Sequential matching method (one end-point free method).

that $L(I,t_2) \geq L(I,t)$, where $r+[(I+1/2] \leq t \leq t_1-2$ or $t_1+2 \leq t \leq \min(r+I+5, r+2I-4)$. Thus, the next sequential matching should start from both y_{t_1} and y_{t_2}.

In order to avoid context effect across adjacent two words, we introduce the rewriting rule $XY \rightarrow xy$ type, where X is the last phoneme in the preceding word and Y is the first phoneme of the word to be identified. This rule means that the sub-phonemes of X and Y are replaced by x and y, respectively(word juncture rule).

III-5-2 Direct Matching Method

At the language processing stage of a speech understanding system, it is often desired to find all appearances of a particular word in continuous speech. This problem is called 'Word Spotting' (Baker[5]). We have not seen a systematic procedure for 'Word Spotting' except for the method proposed recently by Christiansen and Rushforth [12]. Now we propose a new algorithm for this aim.

Let $x_1\ x_2\ \ldots\ x_I$ be a phoneme string of the given word. We want to find at all the places in a recognized string where this word exists with likelihood over a threshold. In comparison with the sequential matching method, this problem is difficult, because the starting point in the matching process is uncertain. To overcome the difficulty, we introduce the pseudo phoneme $/x_{psd}/$. It is added before the word $x_1\ x_2\ \ldots\ x_I$. This new string $\{x_{psd}\ x_1\ x_2\ \ldots\ x_I\}$ is identified throughout the range of an input string, so that we can now obtain the likelihood of the word at any arbitrary portion of the string. In this case, the similarities between $/x_{psd}/$ and other phonemes are always zero, and $/x_{psd}/$ can be associated with one or more segments of the input string. In other words, $S(x_{psd};y_1,y_2,\ldots y_j)$ is zero for all j such that $1 \leq j \leq j$, that is, $L(psd,j)=L(0,j)$ is zero. This pseudo phoneme has the substantial effect to locate the starting point of dynamic programming at arbitrary points. The authors call this method

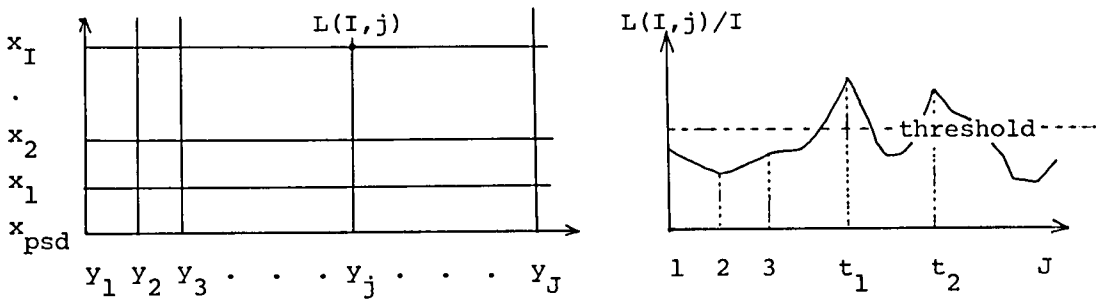


Fig. III-10 Direct matching method (both end-points free method).

```
input                  kesaN ki cyuono zikitepu so  ci saNbaNkarakesaN kigazo e d e  tayoNoro doscyo
segment number         0.....1.....2.....3.....4.....5.....6.....7.....0.....1.....
first candidate        /pucau.pi.cuuoro-puciges.csou.ci.caobaopao-pucaN.pidazouu.bgcu.cabogoroogusce/
second candidate       -cisegpcuphiNuuu.hihuducppcuopsbpsoNrorcor/cisNuphuŋoduce-ubuNphonNuubeudocug-
reliability (x100)     61  6  11612642    13 24916 29229 3 73 2  11 1313 31 14 342 1 9471 22113 243 6
                       06620000662426020806068098060928980484000020822664860620004446984466460444444O
```

	main-pho.	/	k	e	i	s	a	N	:	k	i		s	o	.	c	i
	sub-pho.	k		e				k		k						c	c
lexicon										1						1	1
	weight	9		9				9		0						0	0
	(X100)	5		5				5		0						0	0

	score	94	94	94	91	89	90	91	95	95	94	89					
keisanki	head	0	0	0	0	0	0	52	52	52	52	52					
	tail	6	7	10	11	12	13	60	61	62	63	64					
	score	90	91	93	89	90	92	89	89	97	98	94	90	91	92	93	91
sochi	head	1	3	3	3	7	21	21	23	32	32	32	36	53	55	55	55
	tail	3	7	10	11	13	24	25	31	36	37	40	41	55	61	62	63

Fig. III-11 Detection of "keisanki[=computer]" and "sōchi [=device]" by direct matching method. (An elongated vowel is represented as a vowel)

a 'direct matching method' (both end-points free method). It is mainly used for detecting key words in utterances.

The formal description is illustrated in Fig. III-10 and an example of matching by this method is shown in Fig. III-11. The utterance is "Keisanki cyūō-no zikidisuku sōchi san-ban kara keisanki gazō-e dēta yon-o rōdoseyo". (Load the 4th datum from the 3rd magnetic disk device of the central computer to the image-processing computer.) The detection of three key words, i.e., "dengen"[=power source], "keisanki"[=computer] and "sōchi"[=device] was tried. The key word "dengen" was not detected in the output form (Fig. III-11), on which was printed only the words in this utterance having a score (or

likelihood) above 89. Overlapping locations are allowed to keep only one location, where the matching score is highest. The key word "keisanki" was detected at two locations [0,6] and [52,61], illustrated by Gothic letters in the figure. (The actual locations are [0,7] and [53,62].)

III-5-3 Inverse Matching Method

At the parsing stage of an SUS, we would sometimes like to try not only left-to-right parsing, but also right-to-left. If right-to-left parsing starts at some point on a recognized phoneme string, the Word Identifier must identify the predicted words backward from that point. In short, we need inverse matching between the two strings. Such is easily performed by reversing the order of both strings in applying the sequential method.

An example of reversing a lexical entry is illustrated in Fig. III-12. Here we must distinguish the reversal of an element with the special symbol (+). We will use this method for identification (or verification) of predicates (or verbs) at the last portion of an input string.

input string $y_1\ y_2\ y_3 \cdots y_{J-1}\ y_J \rightarrow y_J\ y_{J-1} \cdots y_3\ y_2\ y_1$

lexical entry	main phoneme	h	a	.	c	$\overset{+}{\underset{-}{i}}$			$\overset{-}{i}$	c	$\overset{+}{.}$	a	h
	sub phoneme				c	c	→		c	c			
	weight				1.0	1.0			1.0	1.0			

Fig. III-12 An example of reversal of strings for inverse matching method.

III-6 Evaluation of Matching Algorithms

The performance of this sequential matching method was examined under various conditions by using a set of arithmetic expressions consisting of 80 sentences (560 words), uttered continuously by five male adults. The (sequential) matching method includes various kinds of subprocedures and data bases. Seven experiments were made by excluding one or more subprocedures or certain parameters (data base). They are listed below:

(a) Only one end boundary of a word is located.
(b) The degree of confidence of the first candidate is not used. Such is simply performed by setting that degree at zero.
(c) The second candidate phoneme is not used. This amounts to the case where the degree of confidence is always 1.0.
(d) The duration time of each segment is not employed, that is, the matching restrictions (4), (5) and (6) in section III-4-2 are not imposed.
(e) The last element of preceding word is not added to the next word, but the first candidate of the recognized segment associated with the last element is added.(e.g., in Fig. III-8, 'nwaru' instead of 'Nwaru'.)
(f) A new similarity matrix is used which is derived from distinctive features of Japanese phonemes, as shown in Table III-4 (Fujimura[20]). The similarity between the two phonemes i and j is defined by

$$S(i,j) = 100 - 5 \times \sum_{k=1}^{10} |i_k - j_k|, \qquad \text{(III-13)}$$

where k denotes the k-th distinctive feature, and corresponding to i_k=+, -, or a blank, i_k is transformed into 1, -1, or 0, respectively.
(g) The sequential matching method itself, in which two end boundaries are located, is tested.

The results are shown in Table III-5. Syntactic information for the arithmetic expression was used for all experiments. The parameters for the tree pruning in parsing were $\alpha=\beta=\gamma=5$ (see section V-3). The results suggest that our proposed matching method is very successful, particularly in regard to phonemic representation with

Table III-4 Distinctive features of Japanese phonemes

distinctive feature	a	i	u	e	o	N	y	w	m	n	g̃	b	d	g	r	z	s	c	h	p	t	k
vocalic/nonvocalic	+	+	+	+	+	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-
consonantal/non-consonantal	-	-	-	-	-	+	-	-	+	+	+	+	+	+	+	+	+	+	-	+	+	+
diffuse/compact	-	+	+	-	-							+	+	-						+	+	-
acute/grave	-	+	-	+	-		+	-	-	+		-	+							-	+	
flat/plain	-				-																	
tense/lax												-	-	-		-	+	+	+	+	+	+
voiced/unvoiced	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	-	-	-	-	-
strident/mellow																+	+	+				
nasal/oral						+			+	+	+											
continuant/interrupted	+	+	+	+	+	+	+	+	+	+	+	-	-	-	+		+	-	+	-	-	-

Table III-5 Evaluation of word matching methods on speech recognition of arithmetic expressions.
'o' denotes the use of information of left column.

method	a	b	c	d	e	f	g
first candidate of phonemes	o	o	o	o	o	o	o
second candidate	o	o	×	o	o	o	o
degree of confidence of first candidate	o	×	×	o	o	o	o
duration time of the segment	o	o	o	×	o	o	o
last element of a preceding word (lexical entry)	o	o	o	o	×	o	o
similarity matrix derived from Bhattacharyya distance	o	o	o	o	o	×*	o
number of identified location	1	1	1	1	1	1	2
recognition rate of word	93.5%	89.4%	91.3%	91.8%	91.2%	84.4%	93.8%
recognition rate of sentence	60.0	40.0	53.0	58.0	54.0	45.0	64.0

* was derived from distinctive features. ($\alpha=\beta=\gamma=5$, see Chapter V)

the degree of confidence and similarity matrix derived from Bhattacharyya distance. We should be aware that the performance of strategies (b) and (c) depends on the performance of the Phoneme Recognizer. However, for poor performance our method behaves like strategy (b), and for better performance, like strategy (c). Thus, our method is actually independent of the Phoneme Recognizer, thanks to the introduction of the degree of confidence.

Next, we examined the performance of the matching procedure in normalization of coarticulation. For speech material, a set of three successive vowels was used in this experiment, as these kinds of phoneme strings can be remarkably influenced by coarticulation. Ten meaningless words were selected at random, and uttered five times by each of 10 male adults. The results are shown in Table III-6. The recognition rate without normalization was 93.6, while normalization improved the rate to 95.6%.

Table III-6 Confusion matrix of speech recognition of three successive vowels.

(a) without normalization of coarticulation

in \ out	uie	uoi	oia	iei	oue	aeo	ueo	uai	ioi	eia
uie	50									
uoi		48			1			1		
oia	1		46		3					
iei				49					1	
oue					50					
aeo						49				
ueo							50			
uai		10			1	1		38		
ioi		2		11					37	
eia										49

(b) with normalization of coarticulation

in \ out	uie	uoi	oia	iei	oue	aeo	ueo	uai	ioi	eia
uie	50									
uoi		48			1			1		
oia			48		2					
iei				49					1	
oue					50					
aeo						49				
ueo							50			
uai		3			1			46		
ioi		2		11					37	
eia										49

III-7 Experimental Results of Spoken Digits Recognition

As described in the section on system specifications (section I-3-1), word identification is one of the most critical parts to develop a successful SUS. Since it is easier to recognize spoken words in isolation than that in continuous speech, it is desired that recognition be performed with high accuracy. Therefore, the Word

Table III-7 Confusion matrix of speech recognition of ten digits

(a) by common reference patterns for 10 male speakers, used for the system design.

in \ out	1	2	3	4	5	6	7	8	9	0
1	50									
2		49		1						
3			49						1	
4		3		42	4		1			
5				4	45				1	
6						43				
7		1					49			
8								49	1	
9									50	
0		1								49

(b) by personal reference patterns for each of 10 male speakers.

in \ out	1	2	3	4	5	6	7	8	9	0
1	50									
2		49			1					
3			49						1	
4				43	6		1			
5				2	48					
6						43				
7							49			1
8								49	1	
9									50	
0										50

(c) by common reference patterns for 10 new male speakers.

in \ out	1	2	3	4	5	6	7	8	9	0
1	50									
2		50								
3			50							
4				45	5					
5		1		3	46					
6						50				
7				1			49			
8	2							48		
9									50	
0										50

Identifier was tested for ten isolatedly uttered digits. The lexicon is the same ones as shown in Table III-2. The experiments were composed of the following three types:

Experiment 1. The common reference patterns (similarity matrix, reference spectra) for the ten male speakers, whose different speech materials were used in the system design, were employed.

Experiment 2. The personally tuned reference patterns were used for each of the same ten speakers.

Experiment 3. The digits spoken by 10 new speakers were recognized by using the common reference patterns used in *Experiment 1.*

For each experiment, each of the ten speakers uttered every digit five times. However, few data from the digits ("roku") were eliminated because the power of /ku/ was very small and the system could not detect it. The experimental results are shown in Table III-7. The recognition rates of *Experiments 1, 2 and 3* were 96.3%, 97.8% and 97.6%, respectively. From these results we can guess that hopefully the performance of our Word Identifier is sufficient as a subsystem of an SUS.

86 項欠

CHAPTER IV

WORD PREDICTOR

This and next chapters will describe the higher levels of the LITHAN system - the Word Predictor and Parsing Director. The description of the Responder, which is now under development, is omitted here. In this chapter, I will first explain the grammar for defining a task language which consists of simple Japanese sentences, and then I will describe the procedure for production of plausible words in unknown parts of input speech.

IV-1 Introduction

Context-free grammars provide adequately simple, powerful tools for natural language processing by computer, however, they are formally incapable of dealing with certain kinds of coordinate constructions and discontinuous constituents. On the other hand, context-sensitive grammars have more sufficient formal power for such constructions, but they provide no useful structural descriptions. As our task treats simple sentences, a context-free grammar can be used for the descriptions. Moreover, we incorporate additional information such as pragmatics, phonological rules, and also procedures for restricting applications of some rules on special contexts. Such a grammar is called an augmented context-free grammar. This additional information and procedure will be used by the Word Restrictor (section IV-4).

Kinds of knowledge the Word Predictor uses are illustrated in Fig. IV-1. Japanese grammar is constructed such that the predicate

part is situated at the last portion of a simple sentence, and the role of this part is very important for syntactic structures and semantics. Therefore, LITHAN places prime importance on the predicate parts of sentences. Nagao, Tsujii and Tanaka [54] have also proposed such a parsing method for typed Japanese sentences.

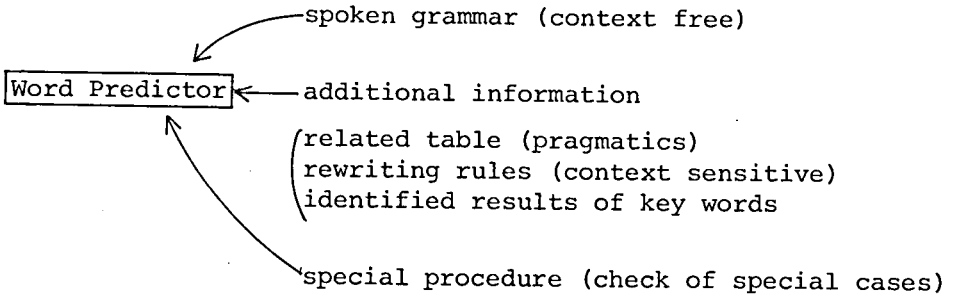


Fig. IV-1 Kinds of knowledge for Word Predictor.

IV-2 Description of Spoken Grammar

In the LITHAN system, the syntax of a given task is defined by a context-free grammar. There is a close relation between predicate and sentence structure or between predicate and the meaning of a sentence. For this reason, we provide syntactic rules for each predicate. In so doing, we gain the advantage of being able to construct a spoken grammar more easily. A partial sentence which is obtained by eliminating only the predicate from a sentence is called a 'sentence structure'. Table IV-1 shows examples of the relationship between predicate and sentence structure. Hereafter, we call this 'PS-table'. Detailed descriptions of 'key word' contained in Table IV-1 will be given in Chapter V.

Fig. IV-2 illustrates an example of spoken grammar, where an uppercase and lowercase letter denote a nonterminal and terminal symbol, respectively, and ε denotes an empty symbol. The words in the

Table IV-1 Relations among predicate, sentence structure and number of key word (PS-table). ε denotes an empty symbol.

predicate	sentence structure	special rewriting rule	key word		
			a		.....
			min	max	
P1	S1		0	2	
P2	S2	R1→a	1	2	
P3	S2	R1→ε	0	1	

vocabulary of a given task are classified into several classes syntactically and semantically. One word can belong to a few classes, each of which is called a 'word class' for convenience. In the rules, such a class is represented by a nonterminal symbol. In Fig. IV-2, 'F' is part of the word classes. In the prediction process, a word class is regarded as a terminal symbol. A word class is an approach of semantically subcategorizing the syntactic categories of the grammar.

We have special symbols in the nonterminals, denoted by 'R_i', and rewritten in a context-sensitive mannar. In Fig. IV-2, 'R1' is such a symbol. It is replaced by the symbol 'a' if the sentence structure is associated with the predicate P2, and replaced by the empty symbol 'ε' if the sentence structure is associated with the predicate P3. Thus, S2 generates a partial sentence 'cfab...' for the predicate P2, and 'cfb...' for the predicate P3. The relationship between a special symbol and a predicate is also registered in the PS-table. It is possible to describe the same thing by a context-free grammar, but using such special symbols can both reduce the number of sentence structures and simplify grammatical structures.

For the purpose of efficient prediction and parsing of a sentence, a spoken grammar should not be ambiguous. Furthermore, we require that this condition exists in any left-most substring of a sentence. In other words, when a left-most substring is shared by two sentences,

(1) S1→A B C d

(2) S1→E f g

(3) S2→c f R1 b A

(4) A→a

(5) A→ε

(6) B→b

(7) B→e

(8) B→F

(9) C→E

(10) C→a

(11) E→F h

(12) F→i

(13) F→j

Fig. IV-2 An example of context free grammar.

S1→a b C d

S1→a B' C E

S1→E f g

B'→e

B'→F

(a)

S1→a B" d

S1→E f g

B"→B C

B"→b C

(b)

Fig. IV-3 Examples of undesirable context free grammar.

(a) inefficient grammar
(b) ambiguous grammar

it must be derived uniquely (unambiguously). The authors call this property of a grammar 'efficient'. Thus a grammar is unambiguous if it is efficient. Examples of undesirable context-free grammars are shown in Fig. IV-3. Example (a) shows an inefficient grammar, that is, the system twice predicts the initial word 'a' of the sentence structure (S1), and so the system must always be conscious of whether it has predicted the same words or not.

From the view point of efficient representations of a grammar, recursion rules may be suitable for computational processing of natural language analysis. However, these rules are not adequate for an SUS, because they may introduce a lot of worthless parsing efforts as opposed to typed input. Therefore we have avoided recursive representations.

The LITHAN system has the following restrictions with respect to grammar:

1. a context-free grammar without recursive rules.
2. an 'efficient' grammar (implying unambiguity).

IV-3 Procedure for Word Prediction

I will explain the process of word prediction by using the grammar shown in Fig. IV-2. By word prediction, I mean the syntactic analysis of a partial sentence as well as prediction of words which can follow that partial sentence.

First I define two terms - 'head symbol' and 'body': a head symbol is that on the left side of a production or rule, and a body is the string on the right side of a production. In Fig. IV-2, for the production (1), a head symbol is S1 and a body is ABCD. A pair of two integers (p,n) is used to refer to the n-th symbol in the p-th production, where n=0 for a head symbol and n=1 for the first symbol of a body. This pair will be called 'coordinates'.

At the initial stage of word prediction, the Parsing Director, using the key word process described in Chapter V, activates the Word Predictor by giving one of the sentence structures, which are hypothesized. The Word Predictor searches for all the productions with the sentence structure as a head symbol. The further process is then the same as at the advanced stage of word prediction. At this advanced stage, the Word Predictor is given a partial sentence together with related syntactic information. This information includes what combination of productions can generate the partial sentence.

In the grammar shown in Fig. IV-2, for example, the partial sentence 'ih' is derived as follows: S1 $\xrightarrow{(2)}$ Efg $\xrightarrow{(11)}$ Fhfg $\xrightarrow{(12)}$ ihfg. This results from applications of the productions (2), (11) and (12). Now, such information could be represented by a sequence of coordinates: (2,1)(11,1)(12,1)(11,2). However, for word prediction it is not necessary to have a process in which a nonterminal is fully converted into a string of terminals. Therefore this sequence is simplified to (2,1)(11,2). Generally speaking, a sequence of coordinates only has to include information about productions which have not come to an end. In a production, whenever a nonterminal symbol (except the special symbol 'R_i'; i=1,2) comes out in a body, a branch appears in a sequence of

coordinates. Such a branch indicates an application of another production with that nonterminal as a head symbol. This branch disappears when the second production comes to an end.

A flow chart for the Word Predictor is illustrated in Fig. IV-4.

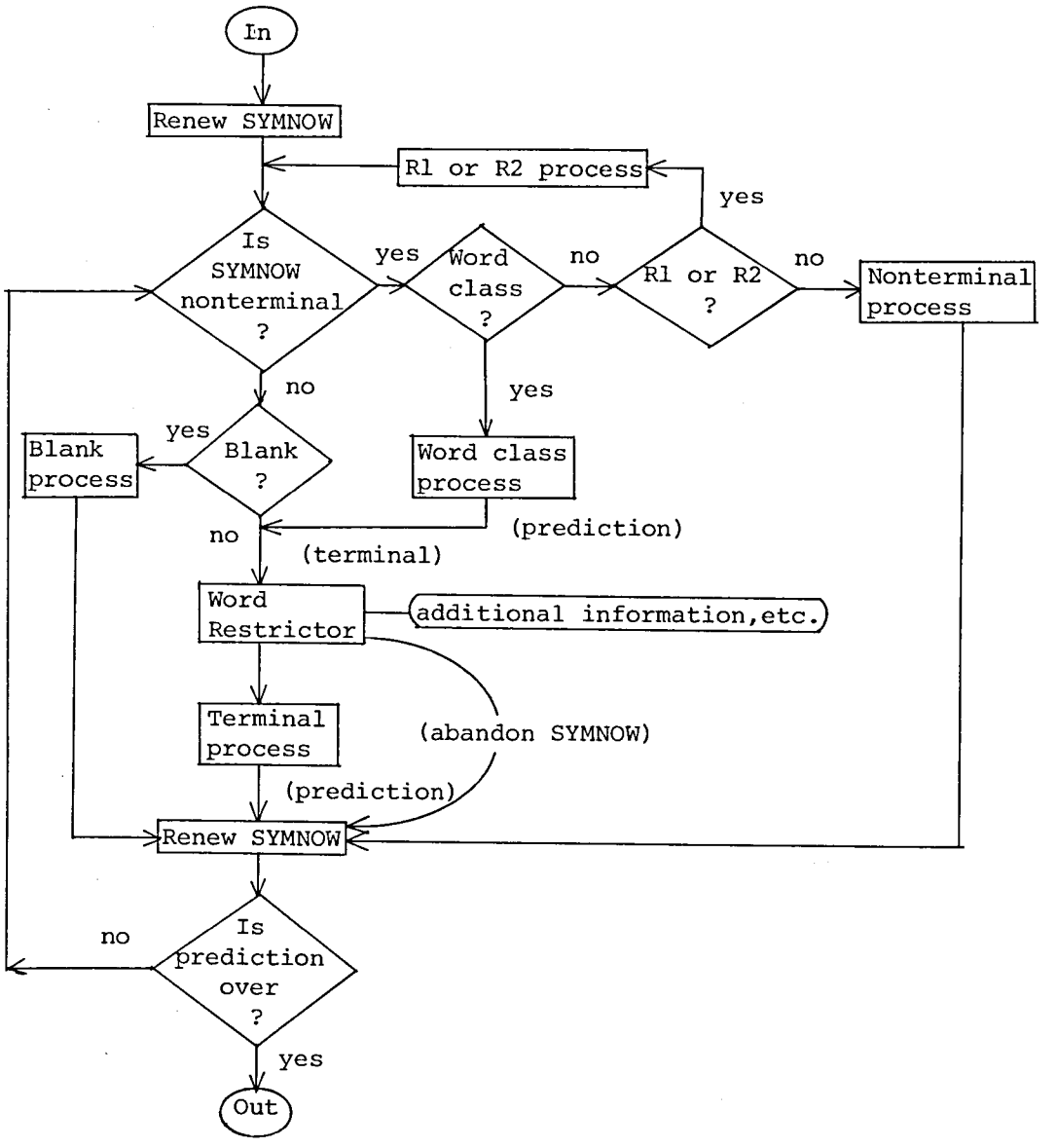


Fig. IV-4 Flow chart in Word Predictor.

In this chart, a partial sentence and its associated sequence of coordinates are given to the Word Predictor. The variable 'SYMNOW' is used to denote a symbol to be subjected to processing. The Prediction Tree Table (PT-table), whose column consists of the number of itself, coordinates and its 'PARENT' number, is used on a stack of sequences of productions. The PARENT number is used to return to the position before branching. The variables 'PRONOW' and 'NONOW' denote the column number of the current process and the total number of columns, respectively. Finally, a word prediction is performed by the following sequence of the steps:

(1) A given sequence composed of n_0 coordinates is transferred to the PT-table. The PARENT value of the first coordinates is set at zero. Each successive PARENT value is increased by 1. PRONOW$\leftarrow n_0-1$ and NONOW$\leftarrow n_0$. Go to (2).

(2) PRONOW$\leftarrow$PRONOW+1. If PRONOW>NONOW, then exit from the Word Predictor. Otherwise, go to (3).

(3) Add 1 to the second term of the PRONOW coordinates and assign to SYMNOW the symbol indicated by the resulting coordinates. SYMNOW is among the certain symbols classified as special, word class, nonterminal which is neither of the two former, terminal or blank. (We say that SYMNOW is blank when the production under consideration comes to an end.) There is a sub-process for each of those symbols. Go to (4).

(4) Go to an appropriate sub-process, depending upon SYMNOW.

(5) *The Special Symbol Process* replaces SYMNOW by a nonterminal or terminal symbol based on the corresponding relation described in the PS-table. Now go to (4).

(6) *The Word Class Process* predicts words belonging to that class. Now go to (8).

(7) *The Nonterminal Process* searches for all the productions with that nonterminal as a head symbol and puts their coordinates corresponding to the heads and PRONOW into the coordinates and PARENT's of open columns in the PT-table. If the number of

productions searched is m, NONOW←NONOW+m. Now go to (2).

(8) *The Terminal Process* sees whether the predicted words (indicated by SYMNOW or predicted by (6)) satisfy necessary contextual and other conditions. This process is called 'Word Restrictor', and will be described in the next section. If a predicted word is allowed, a new sequence of coordinates corresponding to a new partial sentence is generated. Now go to (2).

(9) *The Blank Process* removes from the given sequence the coordinates corresponding to a production currently processed. In other words, this column is replaced by that corresponding to its PARENT. Now go to (2).

The sequence of coordinates for a new partial sentence is generated by backtracking along the PT-table until the PARENT value comes to zero. When the Word Predictor quits these processes, there are four possible cases.

(1) An input partial sentence has turned out to be a complete sentence, and also some words are predicted.

(2) An input has turned out to be a complete sentence, and no word is predicted.

(3) Some words are predicted.

(4) No words are predicted.

Example of word prediction Let me explain the prediction of the right-hand side of the partial sentence 'aj' produced by the grammar of Fig. IV-2. In this case, the sequence of memorized coordinates is (1,2)(8,1)(13,1). This sequence is transferred to the PT-table. First, to the second term of the final coordinates is added 1, and the symbol corresponding to the coordinates is a blank. Thus, in this column, the blank process copies the content of its PARENT' column. Then the predictor adds 1 to the second term of the new coordinates. This corresponding symbol is also a blank. Therefore, the blank process again generates a new coordinates in this column. Since the corresponding symbol is the nonterminal 'C', the predictor searches for all productions with 'C' as a head. In this case, the productions (9) and (10) are selected

and the coordinates of their heads are tranferred to the next two columns....,and so on. Table IV-2 shows the whole predictive process for the example. In this example, three words 'a', 'i' and 'j' are predicted. Their new sequences of coordinates are (1,3)(10,1), (1,3)(9,1)(12,1), and (1,3)(9,1)(13,1), respectively.

Table IV-2 An example of word prediction in Fig. IV-2.
Prediction tree table (PT-table)

prediction process

No.	coordinates	PARENT	order of generation	No. of process
1	(1,2)	0	1	1
2	(8,1)	1	2	1
3	(13,1)	2	3	1
	(13,2)	2	4	3
	(8,1)	1	5	9
	(8,2)	1	6	3
	(1,2)	0	7	9
	(1,3)	0	8	3
4	(9,0)	3	9	7
	(9,1)	3	11	3
5	(10,0)	3	10	7
	(10,1)	3	14	3
6	(12,0)	4	12	7
	(12,1)	4	15	3
7	(13,0)	4	13	7
	(13,1)	4	16	3

Final situation

No.	coordinates	PARENT
1	(1,2)	0
2	(8,1)	1
3	(1,3)	0
4	(9,1)	3
5	(10,1)	3
6	(12,1)	4
7	(13,1)	4

IV-4 Word Restrictor and Additional Information

The LITHAN system uses an augmented context-free formalism to describe the syntax of a task. These augments are concerned with the

manipulation of stacks (or registers), and they also make it easy to construct a complicated spoken grammar. They are given in a production as additional information accompanied by symbols.

(a) Additional information

The internal representation of a symbol used in a grammar is composed of three characters: two for the name of a symbol and one for additional information. (A character consists of 6 bits.) The process is such that a register is associated with each bit (see Fig. IV-4'). When the system meets a word with a 'bit on', this word is stored in the corresponding register. Fig. IV-4' illustrates the relationship between the internal representation and registers. The subscript of a nonterminal symbol is tranferred to the successor(ex. $S \rightarrow AB_1cdE_1 \rightarrow Ae_1cdE_1$ in Fig. IV-5).

1. subscript 1 The LITHAN system has an 'R-table' which shows the semantic or pragmatic relation between two words. Table IV-3 is an example of this R-table, indicating that the symbol 'e' is closely related to the word 'c', 'g' or 'i' in sentences. For example, assume that the Word Predictor predicts a terminal symbol with the subscript 1. If the content of the register 1 associated with the subscript 1 is not empty, it predicts only the words which are correlated by the content in the R-table. On the other hand, if the content of the register is empty, this predicted symbol is stored in the register, and the Word Predictor can predict all words

$S \rightarrow A\ B_1\ c\ d\ E_1$

$A \rightarrow a$

$A \rightarrow b$

$B \rightarrow e$

$B \rightarrow f$

$E \rightarrow g$

$E \rightarrow h$

Fig. IV-5 An example of grammar with subscript 1.

Table IV-3 An example of relative table between words (R-table).

e	c, g, i
f	d, h

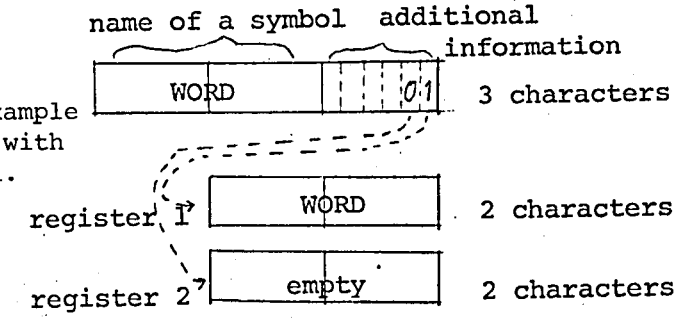


Fig. IV-4' Relatioship between the internal representation and registers.

syntactically permitted.

In the grammar shown in Fig. IV-5, let us consider the prediction of words which can follow the partial sentence 'aecd'. Then by the rule $B_1 \rightarrow e_1$, the register has 'e' as its content. Although the Word Predictor predicts two terminal symbols, 'g' and 'h' in this situation, the latter is abandoned because it is not consistent with 'e'. In case that a partial sentence is 'afcd', then 'g' will be abandoned. Fig. IV-6 illustrates the flow of this process.

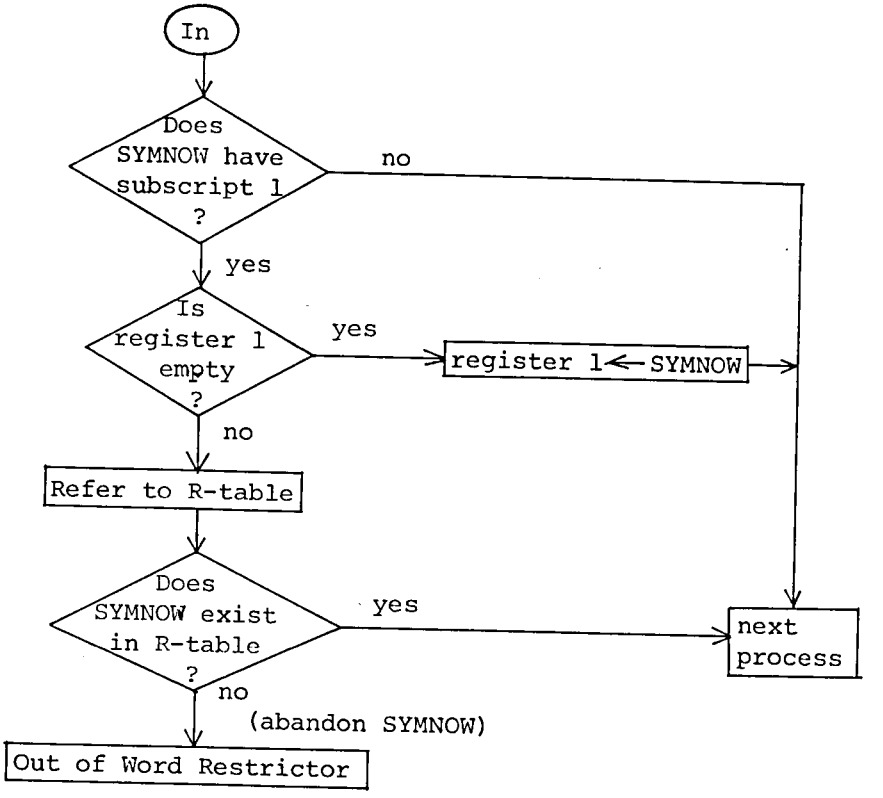


Fig. IV-6 Process related with register 1 (subscript 1).

2. *subscript 2* The subscript indicates that two same nonterminals must be rewritten to the identical terminal when they are accompanied by the subscript 2 in a sentence.

In Fig. IV-7, when an input to the Word Predictor is 'becd', the

content of the register 2 is 'e'. Although the predictor predicts here two terminal symbols, 'e' and 'f', only 'e' will be unconditionally accepted. On the other hand, if the input is 'bfcd', only 'f' will be accepted. Fig. VI-8 illustrates the flow of this process.

$S \rightarrow A\ B_2\ c\ d\ B_2$

$A \rightarrow a$

$A \rightarrow b$

$B \rightarrow e$

$B \rightarrow f$

Fig. IV-7 An example of grammar with subscript 2.

3. subscript 3 When a word affixed to the subscript 3 is predicted, that word is to be followed by some key word. Since a key word has been identified by the key word detection process (see

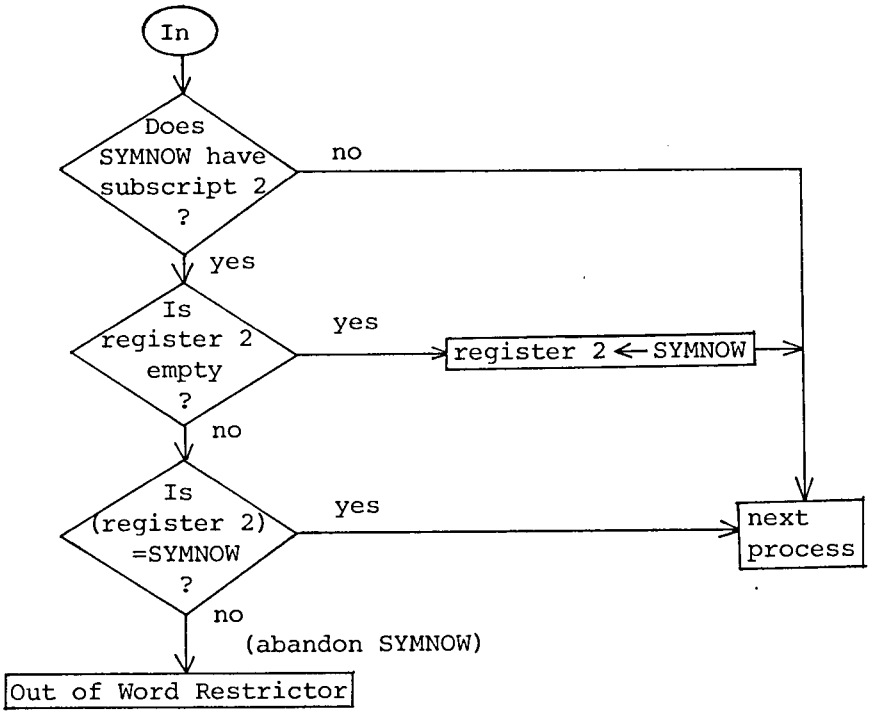


Fig. IV-8 Process related with register 2 (subscript 2).

$S \rightarrow A_1\ B_3\ c_5\ d_4\ E_1$		$S \rightarrow a_1\ B_3\ c_5\ d_4\ E_1$
$A \rightarrow a$		$S \rightarrow b_1\ B'_3\ c_5\ d_4\ E_1$
$A \rightarrow b$		$B \rightarrow e$
$B \rightarrow e_6$	$\equiv$	$B \rightarrow f$
$B \rightarrow f$		$B' \rightarrow f$
$E \rightarrow g$		$B' \rightarrow i$
$E \rightarrow h$		$E \rightarrow g$
$b\ e \rightarrow b\ i$		$E \rightarrow h$
(a)		(b)

Fig. IV-9 Examples of augmented context free grammar.
(a) with context sensitive rewriting rule.
(b) without context sensitive rewriting rule.

Chapter V), the Word Identifier is forced to match that predicted word against the part of the recognized phoneme string. If either the key word has not been identified, or that part in the phoneme string is too long, the predicted word is rejected. In Fig. IV-9, when the Word Predictor predicts a terminal symbol '$e_{3,6}$' or 'f_3' from 'B_3', key word 'c' must be found.

4. subscript 4 In Japanese sentences uttered at a normal speed, pauses do not exist between noun and postposition, or between a number and its unit. The subscript 4 indicates that a pause does not exist in front of a so-designated word. Therefore, the sub-phoneme /// at the top of the lexical entry is not taken into consideration in the sequential matching method (see Fig. III-8).

5. subscript 5 In general, for a word, the larger the number of phonemes, the better the reliability of the likelihood score. The subscript indicates that a concatenation of a word with this subscript and the immediately following one should be subjected to the identification. Therefore, if the Word Predictor predicts a word with the subscript 5, it must further predict the following words to be connected. Because the computation time for word identification expresses a linear function of the length of the entry and the

number of predicted words (namely, words to be identified), the subscript 5 should only be affixed to words that are obviously going to be followed by one word. Because of this procedure, the system does not have to detect a boundary between two words.

6. subscript 6 If the subscript 6 is affixed to the word B, the system searches for the rewriting rules of AB→AC type, and applies them to the word B. If C is an empty symbol, B is rejected, and if A is an empty symbol, B is replaced by C.

In Fig. IV-9, when the Word Predictor predicts a terminal symbol 'e_6' from B for a partial sentence 'b', then 'e' is replaced by 'i'. On the other hand, if a partial sentence is 'a', then 'e' is not replaced. That is, this grammar with such a facility is equivalent to the grammar shown in Fig. IV-9(b).

In general, other additional information which is complementary to the subscript 2 may be necessary. Let this information denote by the subscript 7. Assume that the Word Predictor predicts a terminal symbol with the subscript 7. If this symbol is equal to the content of the register associated with this subscript, this symbol is rejected. This subscript is useful for the phrase of 'a to/matawa b' [a and/or b], where 'a' and 'b' are belonging to the same syntactic category. However, this phrase does not appear in our tasks.

Such rewriting rules provide a lot of conveniences in constructing spoken grammars. Other types of context-free grammar have been augmented by conditions and actions associated with the grammar rules in a similar way(Woods[99], Klovstad and Mondshein[40]), but such grammars, I think, lose the considerations on spoken grammar (namely, subscript 4 and 5).

(b) Procedures for pragmatics

The Word Restrictor has procedures for treating the pragmatics of a task. As an example of such a procedure, let me explain the task 'Computer Network'. The expression "A kara B made"(from A to B) usually refers to the relation of time order between A and B. In the task, this expression appears in sentences for reserving computer

resources, for example, "gozen 8:30 kara gogo 3:30 made"(from 8:30 AM to 3:30 PM). The Word Restrictor predicts words utilizing the fact that the ending time (the latter time) comes after the starting time (the former time).

In the task 'Calendar', if a day and month are given, the day of the week is decided by a special program (see section VI-3).

102項欠

CHAPTER V

PARSING DIRECTOR

V-1 Introduction

There are big differences between the parsing schemes for text and speech input. For text input, traditional parsing schemes always assume that input sentences are syntactically correct. While it is true that such schemes are designed to discover syntactic errors which the parsers might make, their facilities are not adequate to accommodate the kinds of inherent errors of speech input. Thus, when the input is assumed to be noisy or ambiguous, it is necessary to parse all the partial sentences which are likely to appear as the input. Parsing speech is more difficult than parsing text. Particularly, when the input is continuous speech, its parsing becomes still more difficult, because the word boundaries are usually obscured. As the plausible partial sentences for the noisy input might be enormous, one of the aimes of the parser is to cut down search space for the input. For this reason, a parser must have a certain criterion for evaluating each partial sentence.

The structure of the Parsing Director is shown in Fig. V-1. It does not only parse input sentences, but also performs the evaluation of partial sentences and tree search as mentioned above. This operates in a cyclic mannar: A list structure called 'Path List' consists of partial sentences. Thus the Path Selector selects current 'better' paths to be parsed from this Path List. The 'better' path is one which has a higher score (or likelihood) based on the scoring mechanism described in the following section. For each of these selected paths, words which can appear in unknown parts of the path are predicted by the Word Predictor and stored in the Word List. The verification of

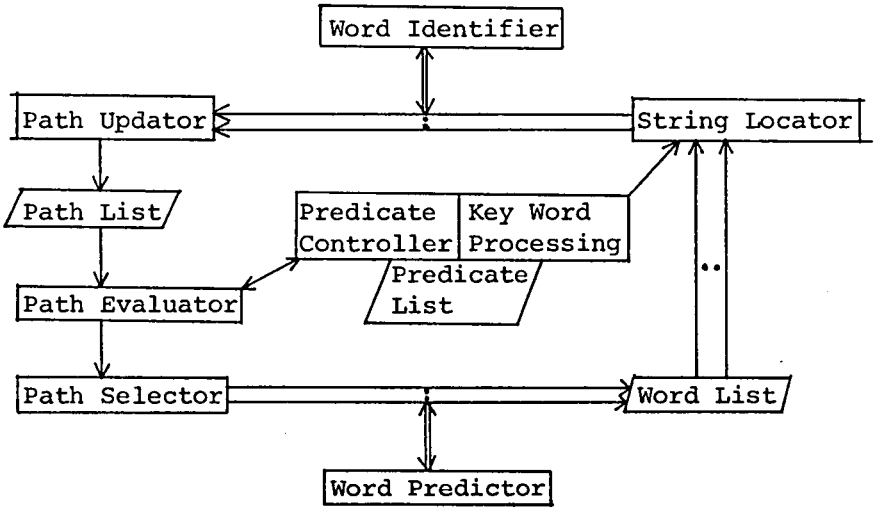


Fig. V-1 Structure of Parsing Director

these predicted words is tried at the location which the String Locator indicates on the basis of *a posteriori* knowledge including the last position in a recognized phoneme string corresponding to the final word of a partial sentence, and so on. Next the Path Updator generates new paths by connecting a selected path with each of the words identified with a certain likelihood. These selected paths are processed in parallel. Sometimes there are two or more partial sentences which have the same syntactic structure, namely, the same sequences of coordinates in the grammar; they further have the same last word and end at the same position in a recognized string. In such a case, only the partial sentence with the highest score is kept, and all others are removed, as these partial sentences will later be processed through the same procedure.

Parsing through such a cycle increases the number of the paths — partial sentences. Therefore the Path Evaluator estimates the likelihood for each of new paths to be expanded and accepts only the limited number with sufficient score. The Predicate Controller always checks whether the sentence structure corresponding to the next predicate should be newly parsed in comparison with the partial sentences of the Path List. The Key Word Processor selects plausible predicates for a given input. It also indicates a possible location

in a recognized phoneme string for a word with the subscript 3 (refer to section IV-3).

V-2 Searching Strategy of Word Sequences

The aim of a speech recognition system is to determine, for an acoustic signal A and a task language L, a sequence of words W such that this sequence is consistent with the syntax, semantics and pragmatics of the task, and further that this sequence maximizes the *a posteriori* probability $P(W/A,L)$. It is possible to introduce an *a priori* probability for the appearance of either each sentence in a task domain or of each word in a word class (Paxton[68]). However, such introduction is beyond the goal of our research, as it only leads to increase the total recognition rate as an expected value. We assume an equal probability for all the sentences. Now $P(W/A,L) = P(W/L) \times P(A/W,L)/P(A/L)$. The term $P(A/L)$ in the denominator is independent of W, and therefore does not have to be known. Also $P(W/L)$ does not have to be known because as mentioned above we assumed that $P(W/L)$ was always constant for all correct sequences of words. Semantic knowledge, however, would not flexibly be introduced to this method. Based this philosophy, Jelinec et al.[35] and Baker [3] have been constructing automatic speech recognition systems. In Japan, Nakatsu et al.[56] and Sakoe[82] have done the same for continuously spoken words. However these methods consume much computing time, because the approach is equivalent to searching all possible paths or words. (In recently, we heard, Lowerre in the Harpy system has reduced the computation requirement by about a factor of 5 over Dragon without any loss of accuracy(Reddy[72] or see Ph.D. thesis of Lowerre, Apr. 1976).)

Seeing this difficulty, we coped with the problem by dividing the whole process into two stages: the acoustic signal-to-phoneme sequence and the phoneme sequence-to-word sequence. This division corresponds to approximating $P(W/A,L)$ by

$$P(W/A,L) = \sum_{PH} P(PH/A,L) \times P(W/PH,L) \propto P(W/PH_0,L) \quad , \qquad (V\text{-}1)$$

where PH denotes a sequence of phonemes and PH_0 a recognized sequence of phonemes. Furthermore, the maximum of $P(W/PH_0,L)$ is approximately obtained from the sequential matching and tree search methods.

If an utterance is composed of s words, and t alternative words are identified for each of these, the number of sequences to be considered is s^t. Therefore the problem of continuous speech recognition can be seen as that of searching through a space of s^t word sequences to find the most acceptable sequence (that is, the sequence with the highest score), that does not contradict all the given knowledge; syntax, semantics, pragmatics, etc.

The Parsing Director performs left-to-right parsing, activating the Word Predictor and Word Identifier when necessary. In this process, the Parsing Director calculates the score of a given word sequence by averaging the likelihoods of all the words contained in it. Fig. V-2 illustrates an example of a parsing tree. In this figure, the alphabetical symbol and numerical value on a branch denote an

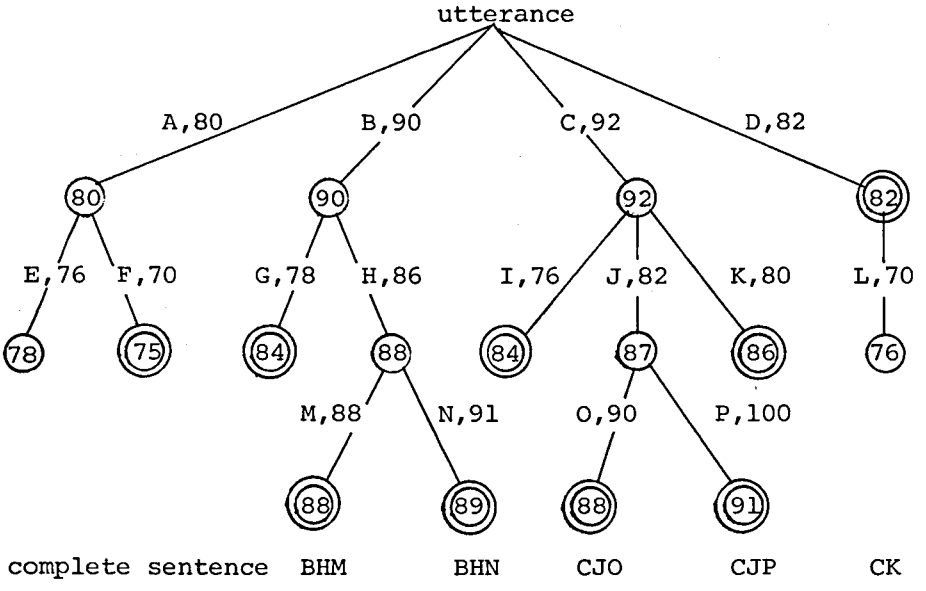


Fig. V-2 An example of tree searching.

identified word and its likelihood, respectively. An encircled numeral on a node indicates the average likelihood for a word sequence corresponding to the path from the root node to that node. A double encircled numeral on a terminal node indicates the score of a complete sentence. The LITHAN system rejects words having a score below a predetermined value, say, 80. It rejects sentences below another value, say, 85. Thus, in Fig. V-2 there are five complete sentences: BHM, BHN, CJO, CJP, and CK. If the best-first tree search is applied, the sentence BHN will be selected after the node selection sequence C→CJ→B→BH→BHN. On the other hand, if we use the depth-first search, BHM will be selected (A→AE→AF→B→BG→BH→BHM). We would, however, like to select the sentence CJP. Now, the best-first search may guarantee the sentence with highest score, if the score of a word sequence has a true cost function for searching (namely the breadth-first search). However, our searching scheme is not such a case. Although many search techniques have been developed in artificial intelligence research(Nilsson[66]), we will propose a new search technique and pruning method suitable for speech understanding.

Pruning method Let α be a number of nodes expanded at one time (parsed in parallel). If the number of new nodes which have been expanded by a selected node exceeds a pre-set threshold(β), only the best β nodes are kept and others are pruned. If the total number of newly generated word sequences exceeds a pre-set threshold(γ), only the best γ word sequences are kept, and others are removed. These three consecutive processes of node selection, expansion, and word sequence generation compose 'one round'. If no word sequence has a better likelihood than that of the next best predicate to be parsed, the sentence structure corresponding to this predicate is newly expanded at the next round. The Parsing Director repeats this process until m reliable sentences are generated.

For example, in Fig. V-2, let α, β, γ, and m be 2,3,4 and 2, respectively. Here the following word sequences will be kept at each round. (The two underlined sequences are to be expanded at the next

round.)

Round 1: B, C, D

D is a complete sentence(rejected) and a partial sentence.

Round 2: BH, CK, CJ, D

G and I are rejected. CK is a complete sentence.

Round 3: CK, BHM, BHN, CJO, CJP

These are all the complete sentences.

In the end, CJP is selected as the recognized (understood) sentence.

Larger values of α, β and γ would bring a better recognition result, but such would also increase the tree search time. Therefore, we have to decide the optimum thresholds of α, β and γ. Preliminary experiments carried out on 'Arithmetic Expression' (see Chapter VI) showed that the sentence recognition rate was almost saturated in the case where α=5, β=5 and γ=15. In the case where α=2, β=5 and γ=10, the rate did not decrease so remarkably. So we used the latter threshold values for the experiments in the task 'Computer Network', although I imagine that the optimum threshold probably depends on the content of a task.

V-3 Parsing Procedure

V-3-1 Use of Key Words

In the previous section, I described an efficient method for pruning partial sentences, where the number increases enormously as the pruning advances. In order to increase the efficiency of parsing, we will introduce the concept of key words.

Key words are selected on the following criteria:

1) They can be exactly detected in a recognized phoneme string.
2) They contribute greatly to syntactic or semantic analysis.
3) They are frequently used in a given task.

Key words are detected by the direct matching method, as described in section III-5-2. Assume that the key word W_k has been identified at several portions in an input utterance. Then identified images (locations) of this key word are divided into two groups by the two thresholds θ_1^k and $\theta_2^k (\theta_1^k \geq \theta_2^k)$, where (a) the likelihood is greater than θ_1^k and (b) the likelihood is greater than θ_2^k but less than θ_1^k. The first group is regarded as existing exactly in the identified location, but the second is not so. Let n_1 be the number of identified images and n_2 the number of images in the first group. As briefly described in section IV-2, there are relationships between the predicate 'P' and the key words. For each of these key words, Table IV-1 gives the minimum and maximum numbers of usages in the sentence structure associated with this predicate. Let these numbers be $n_{k,min}(P)$ and $n_{k,max}(P)$, respectively. In an utterance, if the number of the key word w_k included is n_k, the following inequality holds:

$$n_2 \leq n_k \leq n_1 \tag{V-2}$$

If the predicate of the utterance is P, then

$$n_{k,min}(P) \leq n_k \leq n_{k,max}(P) \tag{V-3}$$

From Eqs. (V-2) and (V-3),

$$n_2 \leq n_{k,max}(P), \qquad n_{k,min}(P) \leq n_1 \tag{V-4}$$

The predicate will be rejected if it is not consistent with Eq. (V-4) for any key word. Furthermore, as explained in section IV-4, key words are used for word restriction.

V-3-2 Parsing Flow

The flowchart for parsing is shown in Fig. V-3. First, the Parsing Director tries to detect the key words in a recognized phoneme string. For each key word, we can obtain the number of its images, their locations and their likelihoods. Such findings make it possible to select some predicates that meet the conditions mentioned

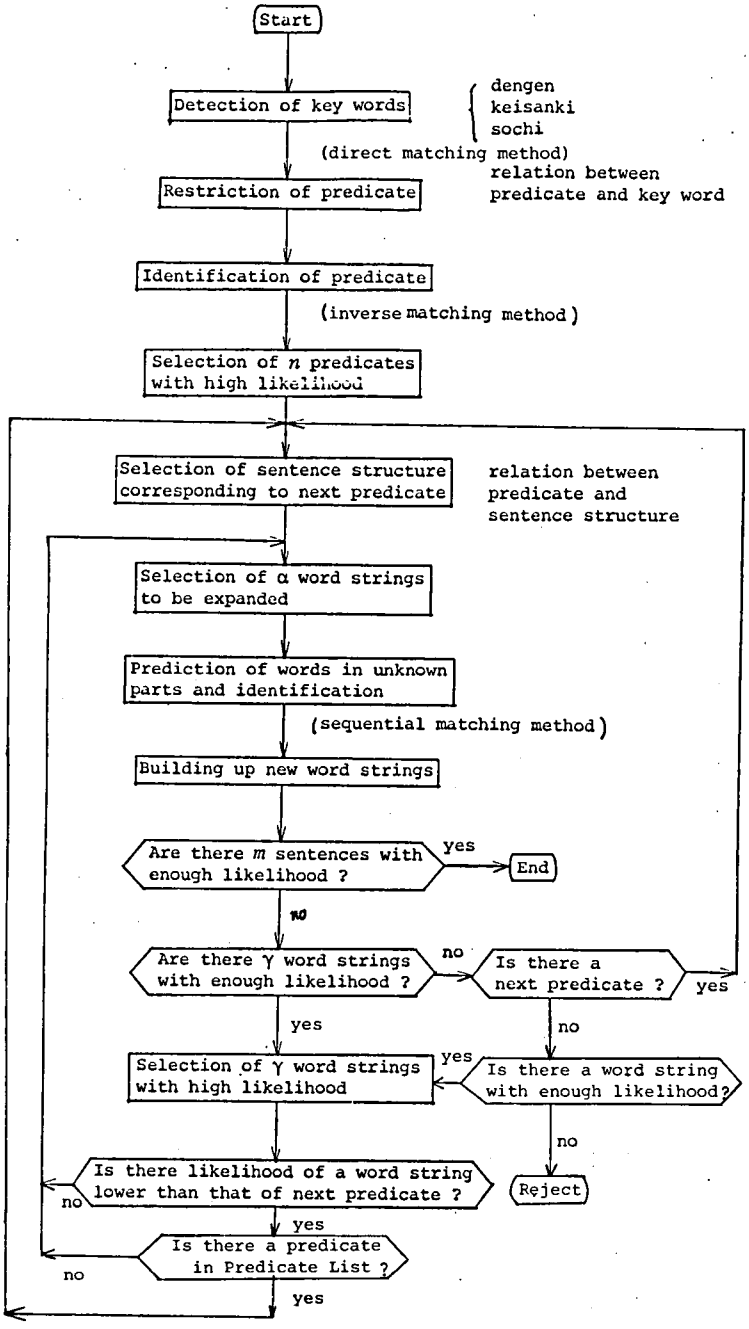


Fig. V-3 Flowchart of parsing

in section V-3-1.

The predicate plays an important role in regard to syntax. Also, it appears at the end in a simple Japanese sentence. Therefore, by using the inverse matching method, each of the selected predicates is matched at the end portion of a recognized phoneme string. Next, n predicates with high likelihood are stored in the Predicate List. From this list, the Parsing Director selects first l predicates with the highest likelihood, and then the associated sentence structures by the PS table (see Table IV-1). Next, the Parsing Director performs left-to-right parsing, activating the Word Predictor and the Word Identifier when necessary. The director begins the process of tree search. During this process, the Predicate Controller always watches whether the sentence corresponding to the next best predicate in the Predicate List is more hopeful and should be parsed. If the likelihood is higher than that of any partial sentence in the Path List, this sentence structure is transformed into the Path List to be parsed newly. Thus, it might happen that some sentence structures get parsed in parallel.

It is considered that the longer the word sequence, the higher the reliability of its score. Therefore, whether a sentence is reliable or not is decided on the basis of the threshold TH dependent on the length l (number of words in the sentence).

$$TH = TH1 - \frac{TH2}{l+1} \quad , \tag{V-5}$$

where TH1 and TH2 pre-set thresholds.

Sometimes a reliable sentence is obtained while some portions remain unmatched in a recognized phoneme string. In such a case the likelihood is decreased in accordance with the number of unmatched segments.

Although the optimum parsing procedure may depend on the performance of the lower levels (Acoustic Processor, Phoneme Recognizer) the system cannot know any more than its expected performance. Nevertheless, we have proposed a powerful tree searching technique. In this technique, our tree search method behaves as the best-first

method for good performance at low levels. For poor performance at low levels, the breadth-first-like method takes over. Therefore, we hope and believe that the Parsing Director will behave as if it knows the performance of phoneme recognition.

CHAPTER VI

SPEECH RECOGNITION OF 'ARITHMETIC EXPRESSION' AND 'CALENDAR'

The speech understanding system LITHAN, which has been presented in the preceding chapters, was throughly tested with three tasks: 'Arithmetic Expression', 'Calendar', and 'Computer Network'. These tasks were carefully chosen; syntactic, semantic, and pragmatic information shows different degree of contribution to understanding of sentences in these tasks.

This and next chapters present the results of the experiment and discussion.

VI-1 Introduction

The higher linguistic information in SUS's refers to syntactic, semantic and pragmatic. Here we have chosen three tasks which can make good use of such information. The task 'Arithmetic Expression' was chosen as the first task in our early research from following reasons.

1. The vocabulary is small and the grammar is well-defined.
2. The task still has the main difficult problems associated with continuous speech recognition by machine.
3. This recognition result is considered to give fundamental data towards a new task selection (i.e., vocabulary size, syntactic complexity, etc.).

Thus, we have developed a speech understanding system to recognize utterances for 'Arithmetic Expression'.

The task 'Calendar' was selected in order to evaluate the LITHAN system. This task makes strong use of syntactic and pragmatic

information.

The sentences dealt with in both tasks are not Japanese sentences. There are no predicates. In the parsing procedure, however, we can regard them as special cases of simple sentences. We consider that each sentence is associated with a dummy predicate whose lexical entry is composed of no element. The Parsing Director bypasses the processing of the predicate.

VI-2 Speech Recognition of 'Arithmetic Expression'

VI-2-1 Task Delineation

The task 'Arithmetic Expression' calculates works as a simple voice-input desk calculator. It calculates an arithmetic expression containing at most two operators. In this task, the vocabulary size is 24, where each word has an average of 2.1 syllables. The vocabulary and word dictionary are shown in Table VI-1. As indicated, the number of lexical entries is not 24 but 28, but this is because pronunciations of some words vary in special contexts. These variations are trickily processed by the use of the subscript 6, that is, the context-sensitive rules.

The spoken grammar is specified in Backus-Normal Form (BNF). The set of BNF productions for 'Arithmetic Expression' is given in Fig. VI-1. An expression can contain numbers whose absolute values are less than ten thousand, whose accuracy is to one decimal place, and which can either have sign parts or not. The expression can end with or without the word "dewa"[=, equal]. It can also contain one pair of parentheses. The following sentence is a typical example: '1.2×(345-6789.0)='.

This task is well-defined on syntax. However, ten digits usually belong to the same syntactic class, and therefore this task is

Table VI-1 Vocabulary and word dictionary of 'Arithmetic Expression'

word		sym-	lexical entry	duration	
Japanese	English	bol		Dmax	Dmin
rei	zero	0	r/p(0.85) e i/e(0.95)	400	100
ichi	one	1	i ./c(1.0) c i/c(1.0)	350	100
ni	two	2	n i	300	100
san	three	3	s a N	550	200
yon	four	4	y/g(0.95) o N	450	150
go	five	5	g o	300	100
roku	six	6	r o ./k(0.95) k u/*(1.0)	450	100
nana	seven	7	n/g(0.85) a/N(0.85) n/a(0.85) a/N(0.85)	550	200
hachi	eight	8	h a/N(0.85) ./c(1.0) c i/c(1.0)	500	150
kyu	nine	9	//c(0.95) k/c(0.95) y/u(0.95) u	500	200
zyu	ten	z	z/d(0.9) y/h(0.95) u	400	200
hyaku	hundred	F	h y/a(0.95) a ./k(0.95) k u/*(1.0)	500	200
sen	thousand	S	s e N	400	150
purasu	plus	P	/p(0.95) p u/p(0.95) r a s u/s(1.0)	500	300
mainasu	minus	M	m a/N(0.85) i n a/N(0.85) s u/s(1.0)	600	300
ten	point	.	//t(0.95) t e N	400	100
tasu	add	+	//t(0.95) t a s u/s(1.0)	500	200
hiku	subtract	-	h i/h(1.0) ./k(0.95) k u/*(1.0)	500	200
kakeru	multiply	*	//k(0.95) a ./k(0.95) k e r u	600	300
waru	divide	/	w/b(0.9) a r u	500	200
migi	right	R	m i g/i(0.85) i	400	200
hidari	left	L	h i/h(0.95) d a r i	500	300
kakko	par.	K	//k(0.95) k a - k o	650	300
dewa	equal	=	d/t(0.9) e w/e(0.9) a/o(0.9)	400	150
byaku	hundred	BF	b y/a(0.95) a ./k(0.95) k u/*(1.0)	500	200
yaku	hundred	Pf	y/a(0.95) a ./k(0.95) k u/*(1.0)	400	150
zen	thousand	ZS	z e N	400	150
en*	thousand	QS	e N	350	100

* corresponds to "sen".

difficult relative to the vocabulary size. The number of total utterances allowed by this grammar is $1.7249768712\times10^{18}$. If the numeral value is restricted below 1000, the number of utterances reduces to 1.724984712×10^{15}.

The pronunciation of numbers varies euphonically as follows:

san hyaku [three hundred] → sanbyaku

roku hyaku [six hundred] → roppyaku

<UT>::=<EX>|<EX>=

<EX>::=<NM><EF>|<EM><OP1><NM>

<EF>::=<OP1><EY>|<OP2><NM><EG>

<EG>::=<OP><NM>|ε

<EY>::=<EM>|<NM><EG>

<EM>::=L_5 K <NM><OP2><NM> R_5 K

<NM>::=<IN><IT>

<IT>::=. <DQ>|ε

<IN>::=<SN><D5>|<D5>

<D5>::=<DG><DS>|S_6 <D4>|<DF>|0

<DS>::=S_6 <D4>|<D3>

<D4>::=<DG><D3>|<DF>|ε

<DF>::=F_6 <D2>|Z <D1>|1

<D3>::=F_6 <D2>|Z <D1>|ε

<D2>::=<DG><DZ>|Z <D1>|1|ε

<DZ>::=Z <DC>|ε

<D1>::=<DC>|ε

<DQ>::=<DG>|0|1

<DC>::=<DG>|1

<DG>::=2|3|4|5|6|7|8|9

<SN>::=P|M

<OP>::=+|-|*|/

<OP1>::=*|/

<OP2>::=+|-

context sensitive rules

3 F →3 BF

6 F →6 PF

8 F →8 PF

3 S →3 ZS

8 S →8 Qs

Fig. VI-1 Spoken grammar of 'Arithmetic Expression'. ε denotes an empty symbol.

hachi hyaku [eight hundred] → happyaku

san sen [three thousand] → sanzen

hachi sen [eight thousand] → hassen

For convenience, for such euphonic changes we change only the lexical entry of the latter word. For example, when the input is 'roku hyaku', we consider this euphonic change as follows (on lexical

level):

ro.ku + hya.ku → ro.kuhya.ku → ro.kya.ku → ro-pyaku

Thus, 'hya.ku' preceded by 'ro.ku' is changed into 'ya.ku'. This procedure is easily performed with the aid of the subscript 6.

VI-2-2 Experimental Results

In the task 'Arithmetic Expression', the LITHAN system was tested on a total of 80 utterances containing 560 words, spoken by five male adults at a normal speed. Some examples of input sentences are shown below.

1. zyū ichi tasu ni kakeru san: 11+2×3
2. hyaku yon hiku go waru roku: 104-5/6
3. mainasu nana kakeru hidari kakko hachi tasu kyū migi kakko: -7×(8+9)
4. rei kakeru sen ten rei dewa: 0×1000.0=

The phoneme confusion matrix is shown in Table II-15. This task was also used for evaluating the performance of the Word Identifier (see section III-6). A key word was not used in this experiment.

An example of parsing is illustrated in Fig. VI-2. The detailed procedure is given in Appendix C. The parameters for the tree search are α=2, β=5, γ=10 and m=2. The input utterance is 'zyū ichi tasu ni kakeru san' [zl+2×3 or 11+2×3], spoken by the speaker SK. The thick line indicates a correct word string. The first number under a word indicates the likelihood of that word and the second number indicates the likelihood of the word string up to that word; the number above the branch gives the order of expansion. The symbol '#' denotes the end of a complete sentence. (The detailed result of the Phoneme Recognizer is presented in Fig. II-8.)

At the first stage of the parsing process, none of the words in the sentence have been recognized. The processing begins left to

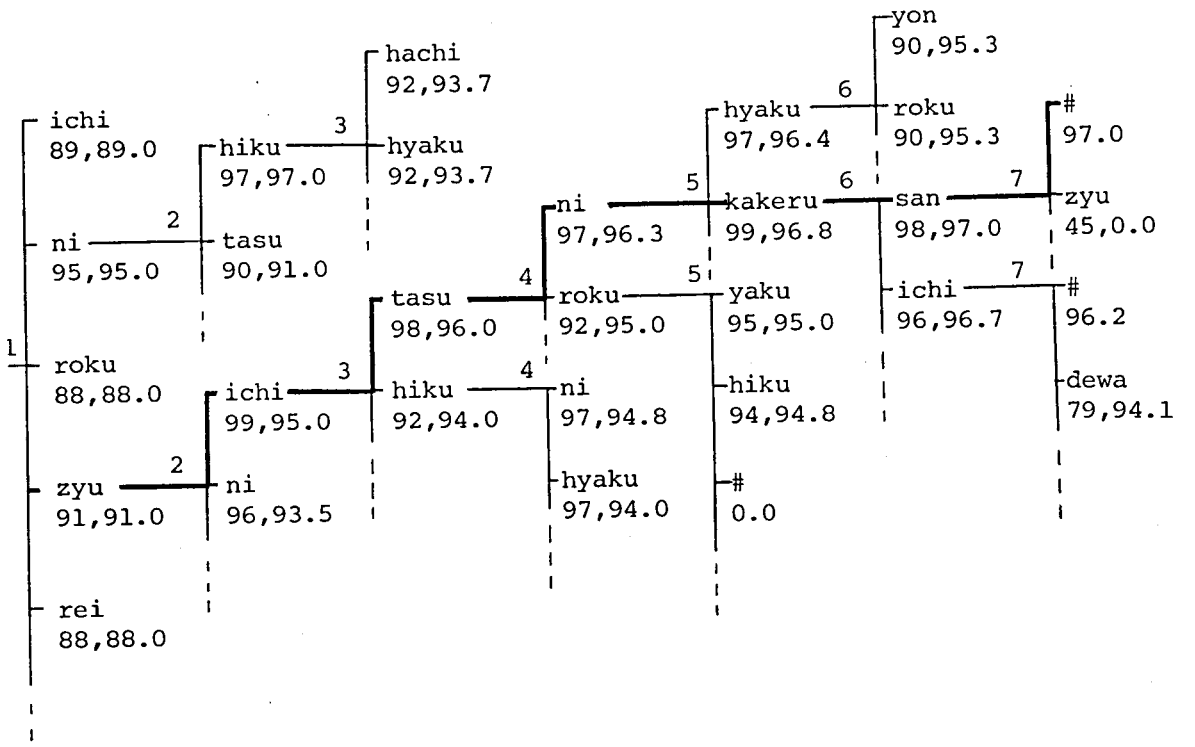


Fig. VI-2 An example of parsing, "zyū ichi tasu ni kakeru san [=1l+2×3]".

right. Then, sixteen possible words are predicted by the Word Predictor and verified by the Word Identifier. Out of these, the Parsing Director selects the best five (because β=5), i.e., "ichi"[1, one], "ni"[2, two], "roku"[6,six], "rei"[0,zero], and "zyū"[z,ten]. Further out of these five, the best two (because α=2), i.e., "2" and "z" are selected and expanded. Out of the remaining three words plus ten newly generated partial sentences (because $\alpha\times\beta$=10), only ten sentences (because γ=10) are selected and kept. This procedure continues until two reliable complete sentences are generated. In the figure, the sentences with the symbol '#' are complete sentences. Although 'zl+6'

in the six round is both a complete and partial sentence, the complete sentence is rejected because there is still much of the input utterance to be identified; then the score is reset to zero. The parsing halts when 'z1+2×3' and 'z1+2×1' were generated.

The relationship between recognition rates and parameters of the tree search is shown in Table VI-2. In this table, the search time column indicates for each case the ratio of the number of verified words to the case where α=1, β=1 and γ=1. In average, the number of predicted words at one time is about 10. From these results, we can conclude that the breadth-first search is superior to the best-first search, and that the case where α=2, β=5 and γ=10 approximates well in the breadth-first search. The value of β is always pre-set at 5, because if a word really uttered belongs to a predicted word class, probably it will be identified as one of the best five words. The method corresponding to the final result uses the subscript 5 (i.e., L_5 and R_5), and identifies each word at two locations.

Table VI-2 Recognition results of 'Arithmetic Expression'.

α	β	γ	search time	recognition rate		notes
				word	sentence	
1	1	1	1.0	80.1%	31.0%	no back tracking
2	2	2	2.0	88.4	47.0	
3	3	3	3.0	92.1	56.0	
4	4	4	4.0	93.1	58.0	
5	5	5	5.0	93.5	60.0	standard (refer to Table III-5)
6	5	6	6.0	93.8	61.0	
1	5	10	1.9	90.3	52.0	(nearly) best-first search
2	5	10	2.6	91.9	58.0	
3	5	12	3.3	93.2	60.0	
5	5	15	5.1	93.6	62.0	(rough) breadth-first search
5	5	15	12.0	77.7	20.0	no syntactic information
5	5	15	5.1	93.9	71.0	final result*

* used the subscript 5 and identified at two locations.

The LITHAN system correctly recognized 71% of the sentences and about 94% of the words. This system also recognized 18% of the sentences which differed from the actual sentence by one word. The detailed results of this experiment are shown in Table VI-3. These results approximately satisfy the following equation:

$$q = p^n , \qquad \text{(VI-1)}$$

where q and p are the recognition rates of sentences and words, respectively, and n is the average number of words contained in a sentence. This means that the number of predicted words at one time is almost constant.

Table VI-3 Detailed results of speech recognition of 'Arithmetic Expression'.

speaker	SK	OT	MG	NK	YS	total
sentence	94%	59%	66%	69%	66%	71%
word	99.1	92.4	94.2	93.1	91.5	94.0
vowel,/N/	88.1	82.1	92.7	85.6	80.3	85.9
semi-vowel	28.6	28.6	14.3	42.9	35.7	30.0
voiced consonant	33.7	31.7	39.6	37.5	28.8	34.3
unvoiced consonant	90.2	84.1	72.0	74.6	83.5	81.0

VI-3 Speech Recognition of 'Calendar'

VI-3-1 Task Delineation

The 'Calendar' deals with a date. Such a task can be quite valuable. As an example, for hotel or seat reservations, being able to use computer for date inputs is going to become more and more necessary. In the task 'Calendar', input sentences are composed of a month, day, and day of the week, such as 'ichi gatsu tsuitachi suiyōbi'(Wednesday, January 1). The third term (day of the week) is

redundant, but it can be used effectively for recovery of misrecognition. The vocabulary contains 30 words, and a word has an average of 2.5 syllables. The vocabulary and word dictionary are shown in Table VI-4. In this word dictionary, there are many elongated vowels, represented by a special notation in computer memory, and there are also many monosyllabic words such as "ni", "shi", "go", "ku" and "zyū". Also, there are many pairs of confusable words, such as "hutsuka" and "itsuka", "hutsuka" and "hatsuka", "mikka" and "muika", "yokka" and "yōka", "nichiyōbi" and "getsuyōbi", "suiyōbi" and "kinyōbi", and so on.

Table VI-4 Vocabulary and word dictionary of 'Calendar'

word		sym-	lexical entry	duration	
Japanese	English	bol		Dmax	Dmin
tsuitachi	first	-1	//c(1.0) c ū/c(1.0) i ./t(0.95) t a ./c(1.0) c ī/(1.0)	800	300
hutsuka	second	-2	h ū/h(1.0) ./c(1.0) c ū/c(1.0) ./k(0.95) k a	700	300
mikka	third	-3	m i/m(0.95) - k a	700	300
yokka	fourth	-4	y/g(0.9) o - k a	700	300
itsuka	fifth	-5	i ./c(1.0) c ū/c(1.0) ./k(0.95) k a	650	250
muika	sixth	-6	m u i/u(0.9) ./k(0.95) k a	650	250
nanuka	seventh	-7	n/g(0.85) a/N(0.85) n u ./k(0.95) k a	750	350
yoka	eighth	-8	y/g(0.9) o ./k(0.95) k a	700	350
kokonoka	ninth	-9	//k(0.95) k o ./k(0.95) k o n ō ./k(0.95) k a	1050	450
toka	tenth	TO	//t(0.95) t o ./k(0.95) k a	700	300
hatsuka	twentieth	HA	h a/N(0.85) ./c(1.0) c ū/c(1.0) ./k(0.95) k a	750	300
ichi	one	01	i ./c(1.0) c ī/c(1.0)	350	100
ni	two	02	n i	300	100
san	three	03	s a N	550	200
yon	four	04	y/ḡ(0.95) o N	450	150
go	five	05	g o	300	100
roku	six	06	r o ./k(0.95) k ū/*(1.0)	450	100
nana	seven	07	n/g(0.85) a/N(0.85) n/a(0.85) ā/N(0.85)	550	200
hachi	eight	08	h a/N(0.85) ./c(1.0) c ī/c(1.0)	500	150
ku	nine	09	//k(0.95) k ū/k(1.0)	300	100
zyu	ten	10	z/d(0.9) y/h(0.95) u	400	200
gatsu	month	GT	g a/N(0.85) ./c(1.0) c ū/c(1.0)	500	200
nichi	day	NC	n/g(0.9) i/N(0.9) ./c(1.0) c ī/c(1.0)	500	200
nichiyobi	Sunday	NI	n/g(0.9) i/N(0.9) ./c(1.0) c ī/c(1.0) y/o(0.95) o b i	1100	400
getsuyobi	Monday	GE	g e ./c(1.0) c ū/c(1.0) y/o(0.95) o b i	1100	400
kayobi	Tuesday	KA	//k(0.95) k a y/o(0.95) o b i	900	350
suiyobi	Wednesday	SU	s ū/s(0.95) i y/o(0.95) o b i	1100	400
mokuyobi	Thursday	MO	m o ./k(0.95) k ū/*(1.0) y/o(0.95) o b i	1100	400
kinyobi	Friday	KI	//k(0.95) k ī/k(0.95) N y/o(0.95) o b i	1100	400
doyobi	Saturday	DO	d o y/o(0.95) ō b i	900	350

We usually pause between a day and a day of the week. Thus, the word boundary symbol '/' is added to the first element in the lexical entry of the latter word.

The set of BNF productions for 'Calendar' is given in Fig. VI-3. We cannot briefly explain how to pronounce the days in Japanese, the spoken grammar is not given systematically.

In this task, we can take advantage of the following pragmatics:
(a) The number of months is limited to 12, and upper limit for days depends on the month; for example, January has 31 days.
(b) There is a one-to-one correspondence between a date and a day of the week; for example, the first of January is Wednesday (in 1975).

This relation is dealt with by a special program in the Word Restrictor. Nevertheless, this task has turned out to be unexpectedly difficult, because it includes so many monosyllabic and confusable words.

<UT>::=<BG>$_5$ GT$_4$ <D1><WK>|<SM>$_5$ GT$_4$ <D2><WK>|02$_5$ GT$_4$ <D3><WK>|10 <H5>
<H5>::=01$_5$ GT$_4$ <D2><WK>|02$_5$ GT$_4$ <D1><WK>|GT$_4$ <D1><WK>
<D1>::=<D4>|03$_5$ 10$_4$ <H1>
<H1>::=01$_5$ NC$_4$|NC$_4$
<D4>::=<IN>|<H2>|02$_5$ 10$_4$ <H3>
<H2>::=10 <H3>
<H3>::=<DG>$_{4,5}$ NC$_4$ |-4$_4$
<D2>::=<D4>|03$_5$ 10$_4$ NC$_4$
<D3>::=<IN>|<H2>|02$_5$ 10$_4$ <H4>
<H4>::=<HC>$_{4,5}$ NC$_4$|-4$_4$
<BG>::=01|03|05|07|08
<SM>::=04|06|09
<IN>::=-1|-2|-3|-4|-5|-6|-7|-8|-9|TO|HA
<DG>::=01|02|03|05|06|07|08|09
<HC>::=01|02|03|05|06|07|08

Fig. VI-3 Spoken grammar of 'Calendar'

VI-3-3 Experimental Results

In the task 'Calendar', 120 utterances were recognized. These contained 647 words, and were spoken by twelve male adults at a normal speed. Some input sentences for 'Calendar' (in 1975) are shown below:

1. ni gatsu zyū yokka kinyōbi. (Friday, February 14)
2. go gatsu itsuka getsuyōbi. (Monday, May 5)
3. zyū ichi gatsu ni zyū san nichi nichiyōbi. (Sunday, November 23)
4. zyū ni gatsu san zyū ichi nichi suiyōbi. (Wednesday, December 31)

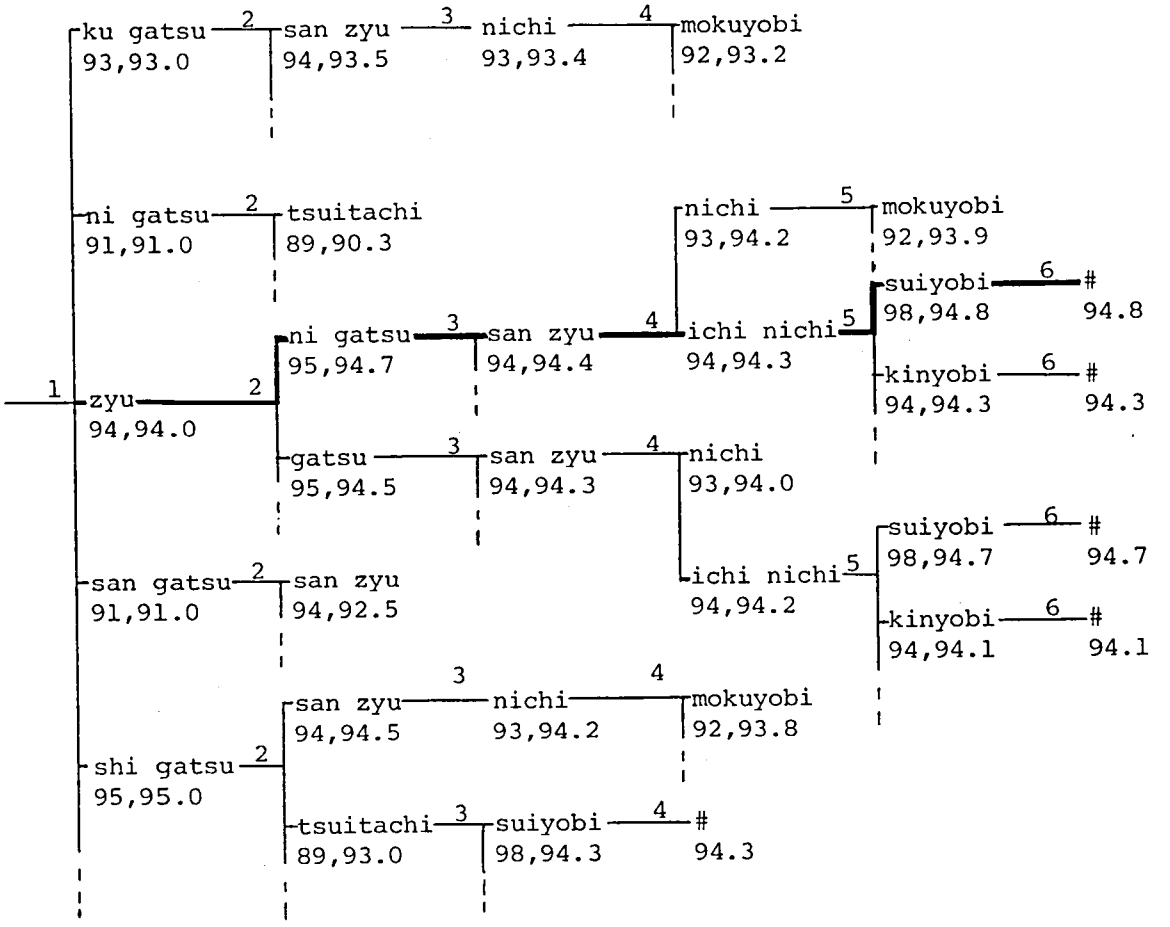


Fig. VI-4 An example of parsing "zyu ni gatsu san zyu ichi nichi suiyobi [Wednesday, December 31]".

No key word was used in these experiments. Fig. VI-4 illustrates an example of this parsing. The detailed procedure is given in Appendix C. Here, the parameters for the tree search are α=5, β=5, γ=5 and m=2. The input utterance is 'zyū ni gatsu san zyū ichi nichi suiyōbi'(Wednesday, December 31) spoken by the speaker SM. Although by the fourth round a complete sentence (shi gatsu tsuitachi suiyōbi; Wednesday, April 1) was generated, the parsing continued until the generation of another complete sentence. In this example, the pragmatic information between a day and a day of the week was not used

Table VI-5 shows the experimental results for 'Calendar'. These experiments were conducted for the three cases: the system with acoustic knowledge only, with acoustics and syntax, and with acoustics, syntax, and pragmatics. The column of the table corresponds to one case. Note that use of pragmatics could not improve the recognition rate of words in comparison with that of sentences. This results because if a sentence is misrecognized, two or more words in that sentence are usually misrecognized.

Table VI-5 Recognition results of 'Calendar'

(a) detailed results

speaker		NK	UK	SM	OK	YN	AR	HY	FZ	TN	MG	AD	NG	total
correct		8	8	7	8	8	7	7	4	8	5	5	8	83
error	1 word													
	2 words	1	1	2	2		3		1		2	1		13
	3 or more			1		1		3	1	2	1	1	1	11
rejected		1	1			1			4		2	3	1	13

(b) by using different linguistic information

used knowledge		acoustics	acoustics, syntax	acoustics, syntax, pragmatics
recognition rate	word	54%	83%	90%
	sentence	8%	40%	70%

CHAPTER VII

SPEECH RECOGNITION OF 'COMPUTER NETWORK'

VII-1 Introduction

In spite of its scientific interest and practical importance, there has been no study done on automatic continuous speech recognition (ACSR) of Japanese. This fact shows that it is very difficult to develop an ACSR system. For example, consider the current art of ACSR for continuously-uttered speech with a 100 word-vocabulary for unspecific speakers. Under such a condition, the word recognition accuracy beyond 80% cannot be obtained. Even if it could be obtained, it can be expected that the recognition accuracy of a sentence would decrease exponentially - to about 10% for a sentence of 10 words. However, the use of higher linguistic information have been expected to be able to lessen this difficulty of an ACSR system. This chapter demonstrates the first attempt to recognize continuously-uttered simple Japanese sentences; these are all based on LITHAN as described in the previous chapters. These experiments have showed that the contribution of higher linguistic information toward the recognition is very remarkable.

The implemented task is 'Computer Network', which is concerned with operational commands and questions about the status of a computer network. The system was tested on a set of 200 utterances spoken by ten male adults. The detailed word dictionary and spoken grammar are given in Appendix B, and some detailed recognition results in Appendix C.

Fig. VII-1 illustrates a model of the computer network which is assumed in the task. This model is based on KUIPNET (Sakai et al.[81],

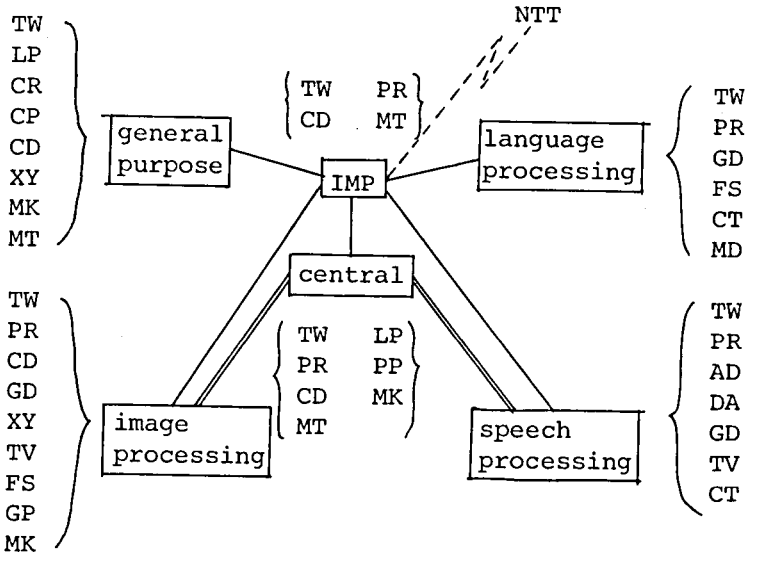


Fig. VII-1 A model of computer network for the task.

A box denotes a computer, double lines denote computer complex. TW:Tele Type Writer, LP:Line Printer, PR:Paper Tape Reader, PP:Paper Tape Puncher, CR:Card Reader, CP:Card Puncher, CD:Character Display, AD:A/D Converter, DA:D/A Converter, GD:Cathode Ray Tube, TV:Color Display, FS:Flying Spot Scanner, GP:Graf Pen, MK:Magnetic Disk, MD:Magnetic Drum, MT:Magnetic Tape, CT:Cassette Magnetic Tape, IMP:Interface Message Processor, NTT:Channel of Nippon Telegraph and Telephone Public Corporation.

see also Fig. II-1). Fig. VII-1 also shows I/O devices connected with computers. It is supposed that each computer has ten units of each I/O device, if it has that kind at all. There is relations of connectivity among computers and between a computer and its devices. In this task domain, such relations will be used as pragmatic information.

VII-2 Vocabulary

The vocabulary of 'Computer Network' contains 101 words: 60 nouns,

Table VII-1 The vocabulary of 'Computer Network'
An elongated vowel is represented as a vowel.

a part of speech		word	sym-bol	word	sym-bol	word	sym-bol
NOUN	computer	kokan	KK	cyuo	KC	hanyo	KH
		onsei	KO	gazo	KZ	gengo	KG
	user	ko	LK	otsu	LO	hei	LH
	device	taipuraita	TW	kosokuinzi	LP	zikitepu	MT
		kasettozikitepu	CT	zikidoramu	MD	zikidisuku	MK
		kamitepuyomitori	PR	kadoyomitori	CR	kamitepusenko	PP
		kadosenko	CP	mozihyozi	CD	gazohyozi	GD
		shikisaihyozi	TV	nizigenhyozi	XY	gazonyuryoku	FS
		gazonyuryoku	GP	onseinyuryoku	AD	onseisyutsuryoku	DA
	number	rei	00	ichi	01	ni	02
		san	03	yon	04	go	05
		roku	06	nana	07	hachi	08
		kyu	09	zyu	10		
	others	gaisen	GS	syukioku	SK	shiyosha	SY
		wariatezyotai	YR	shiyozikan	SZ	dengen	DG
		yoyaku	YY	zyobu	ZB	genzaizikoku	GZ
		deta	DT	puroguramu	PG	seigyobun	SB
		syuryozikoku	SR	yusenzyuni	YZ	zi	ZI
		han	HN	ban	BN	gozen	AM
		gogo	PM	zyotai	ZT	keisanki	KS
		sochi	SC				
interrogative		darega	W1	ikutsu	W2	dorega	W3
		doreo	W4	itsu	W5		
adjactive		aoi	IB	shiroi	IW	kuroi	IK
		kiiroi	IY	akai	IR		
postposition		made	JM	ga	JG	no	JN
		o	JQ	to	JT	wa	JW
		de	JD	ni	JI	e	JH
		kara	JK				
PREDICATE	interrogative	aiteiruka	VA	tsukatteiruka	VB	hashitteiruka	VC
		tsunagatteiruka	VD	haitteiruka	VL		
	imperative	tsunage	VE	yomikome	VF	rodoseyo	VG
		warikome	VH	hazimeyo	VI	tomeyo	VJ
		makimodose	VK	ireyo	VM	kire	VN
		shimese	VO	kakikome	VS	uchidase	VQ
		hazuse	VR				
	affirmative	torikesu	VT	suru	VU		

5 interrogatives, 5 adjectives, 10 postpositions and 21 predicates. Word classification (or part of speech) differs slightly from that of normal Japanese grammar. For example, a predicate is composed of only a verb for an affirmative or imperative sentence, and of a verb, plus an auxiliary verb, and a postposition for an interrogative sentence. Words and their abbreviations are shown in Table VII-1. The word dictionary with equivalent English words attached are given in Appendix B. The nouns include names of computers, users, and I/O devices, digits "rei"[zero] through "zyū"[ten], and others.
Two of the digits have euphonic variants; "yo"(Z4) for "yon"(04:four) and "ku"(Z9) for "kyu"(09:nine) - these variants appear in the phrase for expressing time.

VII-3 Spoken Grammar

VII-3-1 Predicate, Sentence Structure and Key Word

The relationships among a predicate, sentence structure, special rules, and the number of key words are shown in Table VII-2. Three key words are chosen for this task from view points as mentioned in section V-3-1. The key word column shows, for instance, that the key word "dengen"[power source] is only contained in a sentence structure (P7), and that a sentence structure (P5) contains no key word. In the column listing a rewriting rule, D3, D8, T5, T6, T9, TA, and WJ represent word classes(see p.89).

VII-3-2 Pragmatics

Table VII-3 shows which computer has which I/O devices. These relations, introduced from the model of the computer network, are used as pragmatic information. As described in section IV-3, it is handled

Table VII-2 Relation among predicate, sentence structure and key word

predicate	sentence structure	special rule	Key Word					
			dengen		keisanki		sochi	
			mini	max	mini	max	mini	max
aiteiruka	P1	R1→itsu R2→dorega	0	0	1	1	0	1
tsukatteiruka	P1	R1→darega R2→doreo	0	0	0	1	0	1
hashitteiruka	P2		0	0	1	1	0	0
tsunagatteiruka	P3	R1→wa	0	0	1	2	0	1
tsunage	P3	R1→o	0	0	1	2	0	1
yomikome	P4	$R1 \rightarrow D3^{1,3}$ R2→ε	0	0	1	2	1	1
rodoseyo	P4	$R1 \rightarrow D8^{1,3}$ R2→T6	0	0	1	2	1	1
warikome	P5	R1→WJ	0	0	0	0	0	0
hazimeyo	P5	R1→o	0	0	0	0	0	0
tomeyo	P5	R1→o	0	0	0	0	0	0
makimodose	P6		0	0	1	1	1	1
haitteiruka	P7	R1→wa	1	1	0	1	0	1
ireyo	P7	R1→o	1	1	0	1	0	1
kire	P7	R1→o	1	1	0	1	0	1
shimese	P8	R1→T9	0	0	1	2	1	2
kakikome	P8	R1→T5	0	0	1	2	1	2
uchidase	P8	R1→TA	0	0	1	2	1	2
hazuse	PA		0	0	1	1	1	1
kakeyo	P9		0	0	1	1	1	1
torikesu	PB		0	0	1	1	0	1
suru	PC		0	0	0	1	0	1

Table VII-3 Connective relations between a computer and I/O devices or other computers (refer to Fig. VII-1).
An elongated vowel is represented as a vowel.

computer	I/O device	computer
kokan	taipuraita,kamitepuyomitori,mozihyozi, zikitepu	cyuo,hanyo,onsei, gazo,gengo,gaisen
cyuo	taipuraita,kosokuinzi,kamitepuyomitori, kamitepusenko,mozihyozi,zikidisuku,zikitepu	kokan,onsei,gazo
hanyo	taipuraita,kosokuinzi,kadoyomitori, kadosenko,mozihyozi,nizigenhyozi, zikidoramu,zikitepu	kokan
onsei	taipuraita,kamitepuyomitori,onseinyuryoku, onseisyutsuryoku,gazohyozi,shikisainyozi, kasettozikitepu	kokan,cyuo
gazo	taipuraita,kamitepuyomitori,mozihyozi, gazohyozi,nizigenhyozi,shikisaihyozi, gazonyuryoku,zahyonyuryoku,zikidisuku	kokan,cyuo
gengo	taipuraita,kamitepuyomitori,gazohyozi, gazonyuryoku,kasettozikitepu,zikidoramu	kokan

by the subscript 1. This task exploits other pragmatic information that is related to phrases to reserve computer resources. An example was already explained in section IV-3. Time is represented by half-hours. The number of resources such as magnetic tape devices, magnetic tapes, programs, and so forth which one computer has is supposed to be up to 10. The number of users is three: "Kō"[meaning the first or the A], "Otsu"[B], and "Hei"[C].

VII-3-3 Restriction of Syntax

In this sub-section, I will explain the restriction of syntax or allowable sentences in the task.

1. An input sentence is a simple Japanese sentence. In other words, it contains only one predicate.
2. A name label follows a labeled word. For example, in the phrase 'keisanki cyūo' (computer central), 'shiyōsha Kō' (user A) and

'zikitēpu ichi ban' (magnetic tape first), "cyūō", "Kō" and 'ichi ban' are labeles.

3. Although it is a remarkable feature of Japanese sentences that word order is very free, the order of input sentences is fixed except for the following cases:

(a) the position of interrogatives "itsu"[when], "ikutsu"[how many] and "darega"[who],

(b) the phrase order between an indirect object and direct object,

(c) the position of the phrase '...kara...made' (from ... to ...).

Ten examples of allowable sentences are shown below. These sentences were actually used for the experiments. The equivalent English sentences are given in parentheses.

1. Keisanki cyūō-no zikidisuku sōchi san-ban kara keisanki gazō-e dēta yon-o rōdoseyo. (Load the 4th datum from the 3rd magnetic disk device of the central computer to the image-processing computer.)

2. Zyobu hachi-o tomeyo. (Halt the 8th job.)

3. Keisanki kōkan-no zikitēpu sōchi nana-ban-o makimodose. (Rewind the 7th magnetic tape device of the IMP computer.)

4. Keisanki kōkan-to gaisen-o tsunage. (Connect the IMP computer with the channel of N.T.T.)

5. Keisanki hanyō-no kōsokuinzi sōchi-ni genzai-zikoku-o uchidase. (Print the present time on the line printer of the general purpose computer.)

6. Shiyōsha Kō-ga gozen-zyū-ni-zi kara gogo ichi-zi-han made keisanki gazō-no nizigen-hyōzi sōchi-no yoyaku-o suru. (User A reserves the X-Y plotter of the image-processing computer from twelve o'clock in the morning to half past one in the afternoon.)

7. Keisanki gengo-no zikidoramu sōchi-wa dorega aiteiruka? (Which device is free among the magnetic drums of the language-processing computer?)

8. Keisanki cyūō-de zyobu-wa ikutsu hashitteiruka? (How many jobs

are being executed in the general purpose computer?)
9. Onsei syutsuryoku sochi-no dengen-wa haitteiruka? (Has the power source of the D/A converter been switched on?)
10. Keisanki onsei-no kasetto zikitepu sochi rei-ban-e kuroi kasetto zikitepu go-ban-o kakeyo. (Set the 5th black cassette magnetic tape on the 0th cassette magnetic tape device of the speech-processing computer.)

The total number of utterances allowed by this grammar would be several millions.

VII-3-4 An Example of the Spoken Grammar

Fig. VII-2 shows some examples of BNF productions of the spoken grammar. All the BNF productions are given in Appendix B.

For example, when the word class $<D8>_1$ is predicted after recognizing a partial sentence 'Keisanki cyuo no', the Word Predictor predicts only "zikitepu" and "zikidisuku" by consulting the row "cyuo" of Table VII-3 (process of the subscript 1). And also, when "zyobu$_5$" is predicted in a partial sentence 'Keisanki cyuo de', it is connected with the following word "wa", and then the string "zyobuwa" is verified in a recognized phoneme string (process of the subscript 5). As the proper production of time phrase is difficult, it is produced by a special procedure.

VII-4 An Example of Parsing Procedure

Fig. VII-3 illustrates an example of parsing procedure. The detailed procedure is given in Appendix C. The parsing parameters are α=2, β=5, γ=10, m=2, n=5 and ℓ=1. The utterance, spoken by the speaker UK at a normal speed, is 'Keisanki gengo-no zikidoramu sochi-wa dorega aiteiruka?', which means 'Which device is free among the

<P1>::=<Q9><Q1>|<D1>$_3$ sochi <Q7>

<P2>::=<Q2> de$_4$ zyobu$_5$ wa$_4$ <WW>

<P9>::=<Q2> no$_4$ <D8>$_{1,2,3}$ sochi <WS>$_5$ ban$_4$ <WJ>$_4$ <Q6><WS>$_5$ ban$_4$ o$_4$

<Q1>::=<Q7>|no$_4$ <T8>

<Q2>::=keisanki <WK>$_{1,5}$

<Q6>::=<WI><D2>$_2$|<D6>$_2$

<Q7>::=wa$_4$ <U1>

<Q9>::=keisanki <WK>$_1$

<U1>::=<R1>|ε

<U7>::=ikutsu|<R2>|ε

<T6>::=<WS>$_5$ ban$_4$

<T7>::=<D1>|<D3>|<D4>|<D5>|<D7>

<T8>::=<T7>$_{1,3}$ sochi <Q7>|<D8>$_{1,3}$ sochi <TB>

<TB>::=<T6> wa$_4$|wa$_4$ <U7>

<WW>::=ikutsu|dorega

<WS>::=ichi|ni|san|yon|go|roku|nana|hachi|kyu|rei

<WJ>::=ni|e

<WI>::=aoi|shiroi|kiiroi|akai

<D1>::=zahyonyuryoku|onseinyuryoku|onseisyutsuryoku

<D2>::=zikitepu|kasettozikitepu

<D3>::=kamitepuyomitori|kadoyomitori

<D4>::=taipuraita|kosokuinzi

<D5>::=mozihyozi|gazohyozi|shikisaihyozi

<D6>::=zikidoramu|zikidisuku

<D7>::=kamitepusenko|kadosenko|nizigenhyozi|gazonyuryoku

<D8>::=zikitepu|kasettozikitepu|zikidoramu|zikidisuku

Fig. VII-2 Examples of BNF productions of the spoken grammar.
An elongated vowel is represented as a vowel.
ε denotes an empty symbol.

magnetic drum devices of the language-processing computer?'. To explain the process, I will use Fig. VII-3. In this figure, a thick line indicates a correct word string. The first number indicates the likelihood of that word, and the second number the likelihood of the word string up to that word; the number above the branch gives the order of expansion. The symbol '#' denotes the end of a complete sentence.

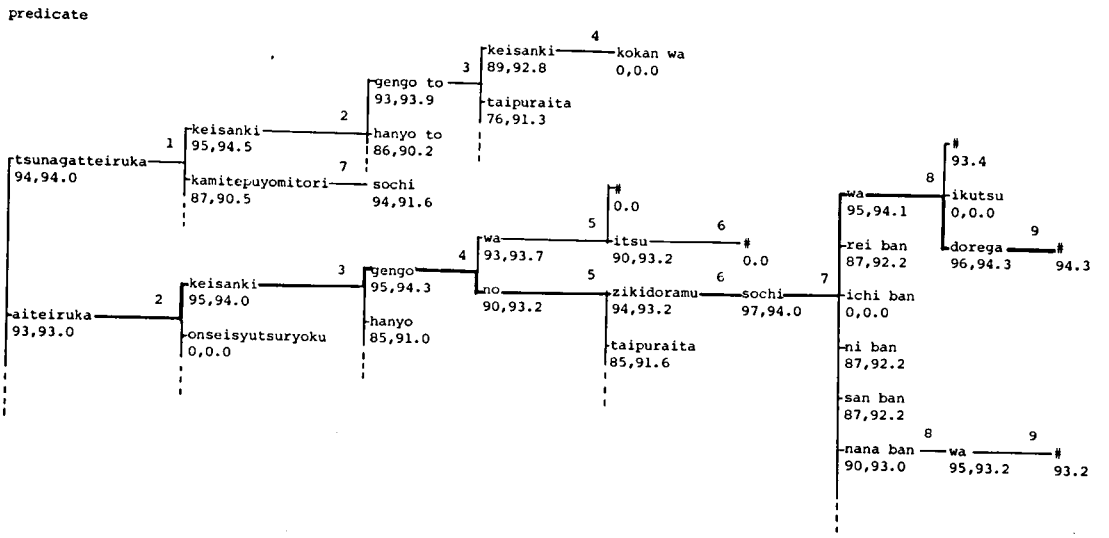


Fig. VII-3 An example of parsing procedure, 'Keisanki gengo-no zikidoramu sōchi-wa dorega aiteiruka?' (Which device is free among the magnetic drum devices of the language-processing computer?) **An elongated vowel is represented as a vowel.**

First, the Parsing Director tried to detect three key words, "dengen"[power source], "keisanki"[computer] and "sōchi"[device], in the recognized phoneme string. As a result of detection, "keisanki" and "sōchi" were respectively detected at two locations, but "dengen" was not. Therefore, the system rejected the predicates which required the key word "dengen" in the sentence structure, such as "haitteiruka" [VL: Is...switched on?], "ireyo"[VM: Switch on...] and "kire"[VN: Switch off...]. Next, the Word Identifier attempted to identify the remaining predicates at the end of the phoneme string. As the likelihood of "tsunagatteiruka"[Is...connecting?] was the highest, the sentence structure corresponding to this predicate was first parsed. In the second round, the best partial sentence 'Keisanki... tsunagatteiruka?' and the second best predicate "aiteiruka"[Is...free?] were parsed newly since the likelihood of this predicate was larger than that of the second best partial sentence 'Kamitepuyomitori... tsunagatteiruka?'. Notice that each digit was connected with the following word "ban"[number], and then they were verified at the same time in the seventh round (process of the subscript 5).

In the midst of parsing, 'Keisanki gengo-wa aiteiruka?'(Is the language-processing computer free?), 'Keisanki gengo-wa itsu aiteiruka?' (When is the language-processing computer free?'),etc. were generated. The Parsing Director, however, rejected these complete sentences, because there still remained much of an unidentified spoken input. Parsing ended when two complete sentences with reliable likelihood were generated: 'Keisanki gengo-no zikidoramu sōchi-wa aiteiruka?' (Is the magnetic drum device of the language-processing computer free?) and 'Keisanki gengo-no zikidoramu sōchi-wa dorega aiteiruka?' (Which device is free among the magnetic drum devices of the language-processing computer?). Finally, the latter complete sentence had higher likelihood, and was determined as the input sentence.

Fig. VII-4 diagrams the parsing tree of this example. P1 and R2 are related with the predicate (aiteiruka). It is observed that the description of this grammar is different from that of a transformation grammar of Japanese because the system requires an 'efficient' spoken grammar (see section IV-2) and the grammar includes semantic knowledge.

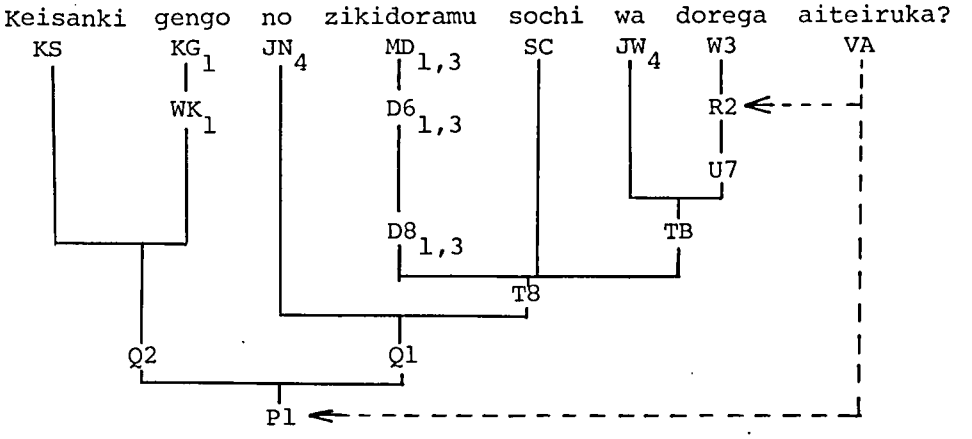


Fig. VII-4 Parsing tree of the input sentence.
An elongated vowel is represented as a vowel.

VII-5 Experimental Results

The current implementation of the LITHAN system was tested on a total of 200 sentences containing 1983 words, spoken at a normal speed by the same 10 male adults used for the system design. A sentence consisted of 4 to 22 words, with an average of 10 words. The ten sentences described in section VII-3-3 were uttered commonly for all speakers. There were fifty other sentences, each uttered by two speakers.

Recognition results are shown in Table VII-4. The parameters on parsing are α=2, β=5, γ=10, m=2, n=5, and l=1. The LITHAN system correctly recognized 131 out of 200 utterances or exactly 65.5%. It also recognized 34 utterances which differed from the actual utterance by one word. Out of these misrecognized 34 words, 16 were digits and 3 were the confusion between "ni" and "e" (postpositions). Also six utterances were rejected. Because none of the correct predicates in these could be identified within the best five predicates. When pragmatic information (the subscript 1) was not used, 60% of the utterances were recognized correctly.

The correct rate of sentence recognition was greatly influenced

Table VII-4 Recognition results of 200 utterances in 'Computer Network'

speaker		NG	FZ	TN	UK	OK	YS	NK	MG	MT	AR	total
correct (not unique)		16	7 (1)	9	14	15 (1)	14 (2)	8 (1)	18	18 (1)	12	131 (6)
error	1 word	3	3	7	6	3	4	3	1	0	5	34
	2 words	0	2	2	0	1	0	3	1	1	1	11
	3 or more words	1	5	1	0	2	2	5	0	1	1	18
rejected		0	3	1	0	0	0	1	0	0	1	6
predicate	best	14	7	14	12	14	12	11	15	15	11	125
	2nd best	3	5	4	6	5	4	5	4	5	7	48
	within best five	19	14	18	20	20	20	20	20	20	19	190
	after parsing	19	14	18	20	19	18	15	20	20	18	183

by both speakers and the speed of utterance. For example, 18 out of 20 utterances were recognized correctly for the speaker MG (utterance speed of 9 phonemes/sec.), but only 7 out of 20 for another speaker FZ (13 phonemes/sec.).

On the other hand, about 93% of the words in the output sentences were recognized correctly. For predicates, the identification results are given in Table VII-4. Such identification is very difficult, because the pronunciation at the end of a sentence is prone to be fast and obscure. Nevertheless, the recognition rate reached about 92% after parsing. In other words, the LITHAN system was able to correctly extract the sentence structures of 183 utterances and 6 other utterances which differed from each other only a predicate.

The current recognition rate for digits was about 80% - much worse than that of spoken digits in isolation (see section III-7). This result suggests that digit recognition in continuous speech is very difficult.

The number of predicted words at one time was 20 for special cases. (of course 21 words for predicates). However, such number was usually reduced to less than 7 as various linguistic information was used.

Except two, all of the 352 key words in test utterances were correctly detected. In these utterances, the number of detected places was almost as twice as those where key words actually appeared.

138 項欠

CHAPTER VIII

CONCLUSION

VIII-1 Summary of the LITHAN System

The primary focus of the present work has been on developing the first speech understanding system for Japanese sentences. The secondary focus has been on improving the systems developed by other researchers and the third on investigating the contribution of linguistic information to automatic speech recognition.

The LITHAN speech understanding system described in this thesis uses many types of *a priori* knowledge: the word dictionary, spoken grammar with additional information, and semantic and pragmatic information. This is the first system that recognizes continuously spoken utterances in simple Japanese sentences.

The decomposition of LITHAN is shown in Table VIII-1. This system consists of the Acoustic Processor, Phoneme Recognizer, Word Identifier, Word Predictor, Parsing Director, and Responder (under development). It is also decomposed of the acoustic, parametric, phonemic, lexical and sentence levels. Principal results obtained and techniques developed in this research are summarized as follows:

Acoustic, Phonemic Level

(1) In order to normalize speaker differences, we proposed new learning methods for spectra with or without teacher, and we verified the effectiveness.

(2) By using learned vowel spectra, we proposed a new method which estimates the spectra of voiced consonants and the optimum similarity matrix for a new speaker.

(3) The output of the Phoneme Recognizer is a sequence of segments, each consisting of four-tuples: the first and second candidates of phonemes,

Table VIII-1 Decomposition of the LITHAN system

level	acoustic	parametric	phonemic	lexical	sentence
input	speech	feature parameter	parameter, segment	phoneme string	word string
output	feature parameter	segment	phoneme	word	sentence
procedure	spectral analysis	empirical rule	Bayes' rule	optimum matching	tree search
processing order	left-right	left-right	left-right	left-right right-left	left-right right-left
strategy	filter bank	heuristic	feed-back, learning	dynamic programming ,bottom-up	pruning, top-down
knowledge		phonological rule	statistic of phoneme, rewriting rule	phoneme similarity, word dictionary	syntax, semantics, pragmatics

the reliability of the first candidate, and the duration time of the segment. Such descriptions of an input utterance were very powerful.

Lexical Level

(4) We proposed a new method for recognizing spoken words and we extended this method to continuous speech recognition. A technique for removing the influence of coarticulation is also included. This method is expected to be applicable pattern recognition of any time-dependent string.

(5) We adopted an automatic description for the word dictionary which useful for reducing matching time.

Sentence Level

(6) The predicate was a most active part in the parsing procedure. Thanks to this fact, we could easily construct spoken grammar and parse an input utterance effectively.

(7) By detecting key words before parsing, we could parse sentences effectively and reliably.

(8) We tried to describe our spoken Japanese grammar by an augmented context-free grammar.
(9) We proposed a new tree searching method suitable for continuous speech recognition, combining the advantage of the best-first search with that of the breadth-first search.

System Organization

(10) We based knowledge flow on the data base (or blackboard) model, and processing flow on the hierarchical model. Thus, we could easily evaluate the contribution of each component to speech recognition.
(11) The LITHAN system can be applied to a new task simply by preparing a new grammar and word dictionary. Such can be done with a minimal effort.

VIII-2 Summary of Experimental Results

The necessity of the use of syntax, semantics and pragmatics has been pointed out to overcome the poor performance of phoneme recognition in continuous speech recognition systems. Many researchers have verified the effectiveness of linguistic information by computer simulation (Takeya and Kawaguchi[92], Niimi and Asami[63], Fukunaga et al.[23]). Their results are not always exact since it is very difficult to simulate phoneme strings generated by a phoneme recognizer. In this thesis, I investigated the contribution of each knowledge to speech recognition by using various tasks: 'Digit', 'Arithmetic Expression', 'Calendar' and 'Computer Network'. These tasks are all different in the degree of consulting with linguistic information.

Similarity matrix and reference patterns were calculated from many speech materials spoken by ten male adults. But reference patterns for 'Arithmetic Expression' were obtained from the same material as the test sentences. Four out of 12 speakers in the 'Calendar' experiment were not subjects for the system design. Therefore

attention must be payed in comparing the rate of phoneme recognition. The rank of success, from highest to lowest, was 'Arithmetic Expression', 'Digit'(in isolation), 'Computer Network' and 'Calendar'. The performance of a system does not depend on only the rate of phoneme recognition but also the content of task language(complexity of task, confusability among words, and so on). The complexity of task language is approximately expressed by the branching factor, which is the average number of words that can follow any other word in the syntax. The factors of 'Digit', 'Arithmetic Expression', 'Calendar', and 'Computer Network' are 10, 10, 8, and 5, respectively.

Experimental results are summarized in Table VIII-2. The 'acoustic' row gives the recognition results of the LITHAN system with an acoustic-phonemic source of knowledge, that is, the syntactic and pragmatic modules can be deactivated so as to permit it to run like a

Table VIII-2 Performance results of the LITHAN system

task		Digit	Arithmetic Expression	Calendar	Computer Network
vocabulary size		10	24	30	101
number of speakers		20	5	12	10
number of test sentences			80	120	200
words in test sentences		1500	560	647	1983
acoustics	sentence		20%	8%	——
	word	97%	78%	54%	——
acoustics, syntax	sentence		71%	40%	58%
	word		94%	83%	91%
acoustics, syntax,pragmatics	sentence			70%	64%
	word			90%	93%
branching factor		10.0	10.0	8.0	5.0
vowel, syllabic nasal		76%	86%	70%	74%
semi-vowel		12%	30%	15%	10%
voiced consonant		10%	34%	22%	28%
unvoiced consonant		65%	81%	64%	44%

connected speech recognition system. However such an experiment was not done for the task 'Computer Network' because I guessed that no utterances could recognize without linguistic information. From these results, we can conclude the following:

1. The apparent contribution of linguistic information to the recognition rate is observed.
2. The degree of this contribution depends on the content of a task.
3. It has been quantitatively shown that it is more difficult to automatically recognize continuous speech than uttered speech in isolation.
4. If a task is well-organized, man will be able to communicate with a machine by speaking continuously in natural language.

VIII-3 Areas for Further Work

I have listed above the characteristics of the LITHAN system and the experimental results obtained in our study. I hope that this work will give a hopeful future to speech understanding. However, at the same time I admit that there are many improvements which can be made even within the framework of the current system. Such improvements are:

Acoustic, Phonemic Level

1. Introduction of prosodic features (stress, duration and pitch pattern). This would serve in the easy detection of word or phrase boundaries, and also increase the reliability of word matching.
2. Development of a more powerful method of phoneme recognition, independent of speaker differences or contexts.

Lexical Level

3. Establishment of phonological rules. This would allow for a variety of the rate of speech or pronunciation.
4. Consideration of the weight on each phoneme in the word matching

evaluation. Vowels can be recognized with higher reliability than consonants and have very important roles on the constitution of words. Therefore, we should make great account of vowels.

Sentence Level

5. Introduction of a model of man's information processing, such as left-to-right, right-to-left, bottom-up and top-down parsing, or a model consisting of short and long term memories. Such would result in speech understanding in the presence of speech-like noise or non-linguistic speech such as breathing, mumble, cough, and hesitation.
6. Development of a parsing method for complex Japanese sentences and automatic descriptions for additional information of spoken grammar by using syntactic and semantic knowledge.

System Organization

7. A close interaction between lower and higher levels. Such interaction would produce an effective ability to recover errors and remove contextual effects.
8. Capability of the definition of a new word or knowledge by man-machine communication. Such capability makes a more flexible system.

Unfortunately, the Responder (under development) of LITHAN was not described in this thesis. If we want to communicate with a machine through speech, the system must understand natural language. This problem is one of main themes on artificial intelligence, and has been studied by many researchers. Although the parsing method of speech input differs from that of text input, I think, both the researchers must cooperate with each other for developing a successful speech-language understanding system. I hope in conclusion that our research will be helpful to developing such a system.

REFERENCES

[1] Abe.K: "Synthesis of an Automaton which Recognizes Patterns Described as Strings of Symbols", *Technical Report of the Professional Group on Pattern Recognition of IECEJ*, PRL 74-5, May 1974. (in Japanese)

[2] Alter.R: "Utilization of Contextual Constraints on Automatic Speech Recognition", *IEEE Trans.* Vol. AU-16, p.6 (1968).

[3] Baker.J.K: "The DRAGON System--an Overview", *IEEE Trans.*Vol. Assp-23, p.24 (1975).

[4] Baker.J.K: "Stochastic Modeling as a Means of Automatic Speech Recognition," *Ph.D.Thesis*,Carnegie-Mellon U. (1975).

[5] Baker.J.K.: "Stochastic Modeling for Automatic Speech Understanding", Speech Recognition, p.521, D.R.Reddy,ed. Academic Press, (1975).

[6] Barnett.J.A: "A Vocal Data Management System", *IEEE Trans.* Vol. AU-21, p.186 (1973).

[7] Barnett.J.A: "A Phonological Rule Compiler", *Proc. IEEE Symposium on Speech Recognition*, Pittsburgh, p.188, (1974).

[8] Bates.M: "The Use of Syntax in a Speech Understanding System", *IEEE Trans.* Vol. ASSP-23, p.112 (1975).

[9] Becker.R.W and I.Poza: "Acoustic Phonetic Research in Speech Understanding", *IEEE Trans.* Vol. ASSP-23, p.416 (1975).

[10] Bhattacharyya.A: "On a Measure of Divergence between Two Statistical Populations Defined by their Probability Distributions, " *Bull. Calcatt math. Soc.*, Vol. 35, p.99, (1943).

[11] Chiba.S. and H. Sakoe: "Application of Time-Normalized Similarity to Speaker Identification", *Record of Joint Convention of ASJ* 2-2-18, Oct. 1974. (in Japanese)

[12] Christiansen.R.W. and C.K.Rushforth: "Word Spotting in Continuous Speech Using Linear Predictive Coding", *IEEE International Conference on ASSP*, p.557, Apr. 1976.

[13] Cohen.P.S. and R.L.Mercer: "The Phonological Component of an Automatic Speech-Recognition System," *Proc, IEEE Symposium on Speech Recognition*, Pittsburgh, p.177, (1974).

[14] Denes.P: "The Design and Operation of the Mechanical Speech Recognition at University College London", *Journal Brit. I.R.E.* Vol. 19, No. 4, p.219 (1959).

[15] Erman.L.D.: "An Environment and System for Machine Understanding of Connected Speech", *Ph.D. Thesis*, Stanford Univ, (1974).

[16] Fant.G: "Automatic Recognition and Speech Research", *STL-QPSR* 1 (1970).

[17] Fant.G: "Key-Note Address", Speech Recognition P.1x, D.R. Reddy, ed. Academic Press, (1975).

[18] Flanagan.J.L: "Speech Analysis, Synthesis and Perception", Academic Press, (1965).

[19] Furui.S: "Learning and Normalization of Talker Differences in the Recognition of Spoken Words", *Technical Report of the Professional Group on Speech of ASJ*, S75-25, Nov. 1975. (in Japanese)

[20] Fujimura.O.: "Speech of Japanese", *Commemoration of 20th Anniversary*, NHK, Radio Tel. Calture Res. Inst., P.363, Nov. 1975. (in Japanese)

[21] Fujisaki.H and T.Kawashima: "The Roles of Pitch and Higher Formants in the Perception of Vowels", *IEEE Trans.* Vol. AU-16, p.73. (1968).

[22] Fujisaki.H, M. Yoshida and Y.Sato: "Formulation of the Coarticulatory Process in the Formant Frequency Domain and its Application to Automatic Recognition of Connected Vowels", *SCS*-74, (1974).

[23] Fukunaga.K,T.Morikawa and T.Kasai, "Recognition of Words with Semantic Information", *Jour. of IECEJ*. Vol. 59-A, No. 1, p.40, (1976). (in Japanese)

[24] Gerstman.L.J: "Classification of Self-Normalized Vowels", *IEEE Trans.* Vol. AU-16, p.78, (1968).

[25] Gold.B and C.M.Rader: "The Channel Vocoder", *IEEE Trans.*Vol. AU-15, p.148, (1967).

[26] Haton.J.P. and Pierrel.J.M.: "Organization and Operation of a Connected Speech Understanding at Lexical, Syntactic and Semantic Levels", *IEEE International Conference on ASSP*, p.430, Apr. 1976.

[27] Hattori.S.: "Methodology of Linguistics", Iwanami Co. 1960. (in Japanese)

[28] Hirakawa.H. and T.Matsuura: "Experiments of Speech Recognition Using Time Inclination", *Record of Joint Convention of IECEJ*, Mar. 1976. (in Japanese)

[29] Hyde.S.R: "Automatic Speech Recognition. Literature Survey and Discussion". *Res.Dep.* Rep. 45, G.P.O., U.D.C. 534.78 (1968).

[30] Ichikawa.A,Y.Nakano and K.Nakata: "Evaluation of Various Parameter Sets in Spoken Digits Recognition", *IEEE Trans.* Vol. AU-21, P.239, (1973).

[31] Ichino.M. and K.Hiramatsu: "Feature Effectiveness Criteria of Statistical Pattern Classifier", *Jour. of IECEJ*,Vol. 53-C, p.748, (1970). (in Japanese)

[32] Itahashi.S,S.Makino and K.Kido: "Discrete-Word Recognition Utilizing a Word Dictionary and Phonological Rules", *IEEE Trans.* Vol. AU-21, p.239, (1973).

[33] Itahashi.S,H.Suzuki and K.Kido: "Discrimination of Some Consonants in Words with the Aid of Dictionary", *Jour. of IECEJ*, Vol. 54-C, p.10, (1971). (in Japanese)

[34] Itakura.F.: "Minimum Prediction Residual Principle Applied to Speech Recognition", *IEEE Trans.* Vol. ASSP-23, No. 1, p.67 (1975).

[35] Jelinek.F,L.R.Bahl and R.L.Mercer: "Design of a Linguistic Statistical Decoder for the Recognition of Continuous Speech", *Proc. IEEE Symposium on Speech Recognition*, at CMU, p.255, (1974).

[36] Kanamori.Y.: "On the Characteristics of Individual Vowel and the Statistical Characteristics of Formant Pattern in Connected Speech", *Jour. of IECEJ*, Vol. 58-D, p.23, (1975). (in Japanese)

[37] Kasuya.H, H.Suzuki and K.Kido: "On Properties of Formant Frequencies of Vowels in Meaningless Words Composed of Three Mores", *Technical Report of the Professional Group on Electric Acoustics of IECEJ*, EA 68-13, Oct. 1968. (in Japanese)

[38] Klatt.D.H. and K.N.Stevens: "On the Automatic Recognition of Continuous Speech: Implications from a Spectrogram-Reading Experiment", *IEEE Trans.* Vol. AU-21, p.210, (1973).

[39] Klein.W,R.Plomp and L.C.W.Pols: "Vowel Spectra, Vowel Spaces, and Vowel Identification", *JASA*, Vol. 48, p.999, (1970).

[40] Klovstad.J.N. and K.F.Mondshein: "The CASPERS Linguistic Analysis System", *IEEE Trans.* Vol. ASSP-23, p.118, (1975).

[41] Kohda.M,S.Hashimoto and S.Saito: "Spoken Digit Mechanical Recognition System", *Jour. of IECEJ*, Vol. 55-D, p.186, (1972). (in Japanese)

[42] Kohda.M, R.Nakatsu and K.Shikano: "Speech Recognition in the Question-Answering System Operated by Conversational Speech", *IEEE International Conference on ASSP*, Apr. 1976.

[43] Kohda.M. and S.Saito: "Mechanical Recognition of Spoken digits by Incomplete Learning Samples", *Record of Joint Convention of ASJ 3-2-15*, May 1972. (in Japanese)

[44] Ladefoged.p. and D.E.Broadbent: "Information Conveyed by Vowels", *JASA*, Vol. 29, p.98, (1957).

[45] Lea.W.A.: "Prosodic Aids to Speech Recognition V: A Summary of Results to Date", *Univac Report* No. PX 11087, (1974).

[46] Lea.W.A,M.F.Medress and T.E.Skinner: "A Prosodically-Guided Speech Understanding Strategy", *IEEE Trans.* Vol. ASSP-23, p.30, (1975).

[47] Lesser.V.R,R.D.Fennel,L.D.Erman and D.R.Reddy: "Organization of the HEARSAY II Speech Understanding", *IEEE Trans.* Vol. ASSP-23, p.11, (1975).

[48] Lieberman.A.M,F.S.Cooper,D.F.Shankweiler and M.Studdert-Kennedy: "Perceptions of the Speech Code", *Psychological Review,* Vol. 74, p.431, (1967).

[49] Makino.S,H.Suzuki and K.Kido: "Recognition of Spoken Word Utilizing Word Dictionary", *Record of Joint Convention of ASJ* 2-1-19, May 1971. (in Japanese)

[50] Massaro.D.W.: "Perceptual Units in Speech Perception", *Jour. of Experimental Psychology,* Vol. 102, p.199, (1974).

[51] Miller.P.L.: "A Locally Organized Parser for Spoken Input", *Ph.D.Thesis,* M.I.T. (1973).

[52] Mori.R.D,P.Laface and E.Piccolo: "Automatic Detection and **Description of Syllabic Features in Continuous Speech",** *IEEE Trans.* **Vol.ASSP-24, p.365, (1976).**

[53] Mori.R.D,S.Rivoira and A.Serra: "A Speech Understanding System with Learning Capability", *Proc. Fourth Int. Joint Conference on Artificial Intelligence,* Tbilisi(USSR), p.468, Sept. 1975.

[54] **Nagao.M,J.Tsujii** and K.Tanaka: "Analysis of Japanese Sentences by Using Semantic and Contextual Information-Semantic Analysis", *Jour. of IPSJ* Vol. 17, p.10, (1976). (in Japanese)

[55] Nagy.F.: "Classification Algorithms in Pattern Recognition", *IEEE Trans.* Vol.AU-16, p.203, (1968).

[56] Nakatsu.R and M.Kohda: "Recognition of Spoken Words on VCV Syllable Unit", *Record of Joint Convention of ASJ* 2-1-12, May 1973. (in Japanese)

[57] Nakatsu.R and M.Kohda: "Computer Recognition of Spoken Connected Words on VCV Syllable", *Record of Joint Convention of ASJ* 2-1-12, Oct. 1974. (in Japanese)

[58] Nash-Webber.B.: "Semantic Support for a Speech Understanding Systems", *IEEE Trans.*Vol .ASSP-23, p.124, (1975).

[59] Neely.R.B.: "On the Use of Syntax and Semantics in a Speech Understanding System", *Ph.D. Thesis,* Standford Univ., (1973).

[60] Neely.R.B. and G.M.white: "On the Use of Syntax in a Low Cost Real Time Speech Recognition System", *IFIP.*74', p.748, (1974).

[61] Newell.A.: "A Tutorial on Speech Understanding Systems", Speech Recognition, p.3, ed. D.R.Reddy, Academic Press (1975).

[62] Newell.A, et al.: "Speech Understanding Systems:--Final Report of a Study Group", Computer Science Department, Carnegie-Mellon University (May 1971), Reprinted by North-Holland (1973).

[63] Niimi.Y and T.Asami: "The Use and Effects of Linguistic

Knowledges in a Speech Recognition System", *Jour. of IECEJ.* Vol. 58-D, p.741, (1975). (in Japanese)

[64] Niimi.Y,Y.Kobayashi,T.Asami and Y.Miki: "The Speech Recognition System of 'SPOKEN-BASIC'", *Proc.2nd USA-JAPAN Computer Conference,* Tokyo, p.375, Aug. 1975.

[65] Nilsson.N.J.: "Learning machines", McGraw-Hill Co. New York, (1965).

[66] Nilsson.N.J.: "Problem-Solving Methods in Artificial Intelligence", New York, McGraw-Hill (1971).

[67] Otten.K.W. "Approaches to the Machine Recognition of Conversational Speech", Advances in Computers, Vol.11, P.127, M.Yovits, ed. Academic Press (1971).

[68] Paxton.W.H.: "A Best-First Parser", *IEEE Trans.* Vol.ASSP-23, p.426 (1975).

[69] Pierce.J.R.: "Whither Speech Recognition?", *JASA,* Vol.46, p.1049, (1969).

[70] Plomp.R.: "The Ear as a Frequency Analyzer", *JASA,*Vol.36, p.1628, (1964).

[71] Pols.L.C.W,L.J.Th.Kamp and R.Plomp: "Perceptual and Physical Space of Vowel Sounds", *JASA,* Vol.46,p.458, (1969).

[72] Reddy.D.R.: "Speech Recognition by Machine; A Review", *Proceedings of the IEEE,* Vol.64, No.4, p.501 (1976).

[73] Reddy.D.R. and L.D.Erman: "Tutorial on System Organization for Speech Understanding", Speech Recognition, p.457, ed. D.R.Reddy, Academic Press (1975).

[74] Reddy.D.R,L.D.Erman and R.B.Neely: "A Model and a System for Machine Recognition of Speech", *IEEE Trans.* Vol.AU-21, p.229, (1973).

[75] Ritea.H.B.: "A Voice-Controlled Data Management System", *Proc. IEEE Symposium on Speech Recognition,*at CMU, p.28, (1974).

[76] Sakai.T. and S.Doshita: "The Automatic Speech Recognition System for Conversational Sound", *IEEE Trans.* Vol.EC-12, p.835, (1963).

[77] Sakai.T. and S.Nakagawa: "A Classification Method of Spoken Words in Continuous Speech for Many Speakers", *Jour. of IPSJ* Vol.17, p.650 (1976). (in Japanese)

[78] Sakai.T. and S.Nakagawa: "A Speech Understanding System of Simple Japanese Sentences in a Task Domain", *Jour. of IECEJ* **Vol.60-E, p.13, (1977).**

[79] Sakai.T,S.Nakagawa and K.Maegawa: "Speech Data Compiling System by KUIPNET Resource Sharing", *Joint Convention of IPSJ,* Dec. 1974. (in Japanese)

[80] Sakai.T and K.Ohtani: "Speech Research System by Computer Network", *Technical Report of the Professional Group on Speech of ASJ* S73-32, Jan. 1974. (in Japanese)

[81] Sakai.T,K.Tabata,H.Ohnishi and S.Kitazawa: "Inhouse Computer Network and HOST Computer", *Jour. of IPSJ*. Vol.15, p.948, (1974). (in Japanese)

[82] Sakoe.H.: "Recognition of Continuously Spoken Words Based on Two-Level DP-matching", *Record of Joint Convention of ASJ*, 2-2-13, May 1975. (in Japanese)

[83] Sakoe.H and S.Chiba: "A Dynamic Programming Approach to Continuous Speech Recognition", *Seventh International Conference on Acoustics*, Budapest, P.65, (1971).

[84] Sato.Y. and H.Fujisaki: "A Method for Recognition of Words in a Limited Vocabulary", *Technical Report of the Professional Group on Speech of ASJ*,S75-27, Nov. 1975. (in Japanese)

[85] Sambur.M.R. and L.R.Rabiner: "A Speaker-Independent Digit-Recognition System", *Bell S.T.J*, Vol.54, p.81, (1975).

[86] Scarr.R.W.A.: "Normalization and Adaption of Speech Data for Automatic Speech Recognition", *Int.J.Man-Machine Studies*, Vol.2, p.41, (1970).

[87] Sekiguchi.Y,H.Oowa, K.Aoki and M.Shigenaga: "Speech Recognition of Programs in JIS Fortran 3000", *Technical Report of the Professional Group on Speech of ASJ*, S75-29, Nov. 1975. (in Japanese)

[88] Shimamoto.T. : "Compiling System of Speech-Data for Listening Tests and Some Considerations on Consonant Perception", *Graduation Thesis*, Dept. of Inform. Science, Kyoto University (1976). (in Japanese)

[89] Shirai.K and H.Fujisawa: "An Algorithm for Spoken Sentence Recognition and Its Application to the Speech Input-Output System", *IEEE Trans*. Vol.SMC-4, No.5, (1974).

[90] Suzuki.J,Y,Kadokawa and K.Nakata: "Formant Frequency Extraction by the Method of Moment Calculation", *JASA*, Vol.35, p.1345, (1963).

[91] Tabata.K.: "Multivariate Statistical Analysis of Speech Sounds", *Doctral Thesis*, Kyoto University, Mar. 1973.

[92] Takeya.S and E.Kawaguchi: "A Simulation of Recognition System for Connected Speech Sounds Using Linguistic Information", *Jour. of IECEJ*, Vol.56-A, No.9, p.513 (1973). (in Japanese)

[93] Takeya.S. and T.Tamachi: "On a Syntax Analysis Method for Speech Recognition Using a Dependency Grammar", *Jour. of IECEJ*, Vol.59-D, No.7,p.475, (1976). (in Japanese)

[94] Velichko.V.M. and N.G.Zagoruyko: "Automatic Recognition of 200

Words", *Int.J.Man-Machine Studies*, Vol.2, p.223, (1970).

[95] Vicens.P.: "Aspects of Speech Recognition by Computer", Report CS-127, *Ph.D. Thesis*, Stanford University (1969).

[96] Vintsiuk: "Generative Grammars and Dynamic Programming in Speech Recognition with Learning", *IEEE International Conference on ASSP*, p.446, Apr. 1976.

[97] Walker.D.E.: "The SRI Speech Understanding System", *IEEE Trans.* Vol.ASSP-23, p.397, (1975).

[98] White.G.M. and R.B.Neely: "Speech Recognition Experiments with Liner Prediction, Bandpass Filtering, and Dynamic Programming", *IEEE Trans.* Vol.ASSP-24, p.183, (1976).

[99] Woods.W.A.: "Syntax, Semantics and Speech", Speech Recognition, p.345, ed. D.R.Reddy, Academic Press (1975).

[100] Woods.W.A.: "Motivation and Overview of BBN SPEECHLIS, An Experimental Prototype for Speech Understanding Research", *IEEE Trans.* Vol.ASSP-23, p.2, (1975).

IECEJ: The Institute of Electronics and Communication Engineers of Japan

IPSJ: Information Processing Society of Japan

ASJ: The Acoustical Society of Japan

PUBLICATIONS AND TECHNICAL REPORTS BY THE AUTHOR

Publication

1. T.Sakai and S.Nakagawa: "Continuous Speech Understanding System -LITHAN-", *STUDIA PHONOLOGICA* IX, p.45, 1975 (in English)

2. T.Sakai and S.Nakagawa: "A Classification Method of Spoken Words in Continuous Speech for Many Speakers", *Jour. of IPSJ* Vol.17, p.650, 1976 (in Japanese)

3. T.Sakai and S.Nakagawa: "A Speech Understanding System of Simple Japanese Sentences in a Task Domain", *Jour. of IECEJ* Vo. 60-E, p.13, (1977).

4. T.Sakai and S.Nakagawa: "Speech Understanding System LITHAN and Some Applications", *Proceedings of the Third International Conference on Pattern Recognition*, p.621, Coronado, Nov. 1976 (in English)

5. T.Sakai and S.Nakagawa: "On-Line, Real-Time Spoken Words Recognition System with Learning Capability of the Speaker Differences", *STUDIA PHONOLOGICA* X, p.46, 1976 (in English).

Oral Presentation

1. S.Nakagawa and Y.Niimi: "A Method for Segmentation of Connected Speech by Statistical Test", *Record of Joint Convention of ASJ* 2-2-12, Oct. 1972.

2. S.Nakagawa and Y.Niimi: "A Method for Segmentation of VCV Speech Sounds", *Record of Joint Convention of ASJ* 2-1-9, May 1973.

3. T.Sakai, K.Ohtani, S.Nakagawa and H.Sakimura: "Segmentation and Recognition of Conversational Speech", *1973 Joint Convention Record of Four Institutes of Electrical Engineers of Japan*, Oct. 1973.

4. S.Nakagawa, H.Sakimura, K.Ohtani and T.Sakai: "Phoneme Recognition of Conversational Speech", *Record of Joint Convention of ASJ* 1-3-2, Oct. 1973.

5. H.Sakimura, S.Nakagawa, K.Ohtani and T.Sakai: "Phoneme-to-Word Translation in Continuous Speech Recognition", *Record of Joint Convention of ASJ* 1-3-3, Oct. 1973.

6. S.Nakagawa and T.Sakai: "Utilization of the Information in Word Strings", *1974 Joint Convention of Four Institutes of Electrical Engineers of Japan*, Oct. 1974.

7. T.Sakai, S.Nakagawa and K.Maegawa: "Speech-Data Compiling System by KUIPNET Resource Sharing", *Joint Convention of IPSJ*, Dec. 1974.

8. T.Sakai and S.Nakagawa: "Continuous Speech Recognition Utilizing Syntactic Information", *Joint Convention of IPSJ*, Dec. 1974.

9. T.Sakai and S.Nakagawa: "A Word Identification Method in Continuous Speech", *Record of Joint Convention of ASJ* 2-2-15, May 1975.

10. T.Sakai, S.Nakagawa and N.Yoshitani: "Continuous Speech Recognition Utilizing Syntactic, Semantic and Pragmatic Information", *Record of Joint Convention of ASJ* 2-2-16, May 1975.

11. T.Sakai, S.Nakagawa and T.Ukita: "Spoken Words Recognition in a Limited Vocabulary Using Mini-Computer by Preliminary Learning of Vowel Spectra", *Joint Convention of IPSJ*, Nov. 1975.

12. T.Sakai and S.Nakagawa: "Speech Understanding System -LITHAN- and Effectiveness of Task Selection", *Joint Convention of IPSJ*, Nov. 1975.

13. T.Sakai and S.Nakagawa: "Speech Understanding System LITHAN and Its Evaluation", *Technical Report of the Professional Group on Speech of ASJ*, S75-30, Nov. 1975.

14. T.Sakai and S.Nakagawa: "Speech Understanding System of Natural Language in a Task Domain -LITHAN-", *Technical Report of the Professional Group on Pattern Recognition of IECEJ*,PRL 75-54. Nov. 1975.

15. T.Sakai, S.Nakagawa and S.Hayashi: "Spoken Words Recognition in a Limited Vocabulary by Preliminary Learning of the Speaker Differences", *Technical Report of the Professional Group on Electric Acoustics of IECEJ*, EA 75-61, Jan. 1976.

16. T.Sakai and S.Nakagawa: "Real Time Recognition System of Spoken Words in a Limited Vocabulary", *Record of Joint Convention of ASJ* 4-2-13. May 1976.

17. T.Sakai and S.Nakagawa: "The Effectiveness of Linguistic Information in a Speech Recognition System - An Experimental Study by LITHAN", *Technical Report of the Professional Group on Electric Acoustics of IECEJ*, EA 76-52, Jan. 1977.

154項欠

APPENDIX A

ON-LINE, REAL-TIME SPOKEN WORDS RECOGNITION BY MINI-COMPUTER

The LITHAN system has as a subpart a spoken words recognition system for limited words (see section III-6). We have implemented this subsystem, called LISTEN (LImited Spoken Texts ENcoder), on a mini-computer, providing the great feature of real time operation. Because of this new subsystem, it became fairly easy to carry out various experiments on many speech data.

In this Appendix, I describe in detail the effect of preliminary learning of speaker differences on the recognition of spoken words. The experimental results mentioned below are obtained from abundant speech data. Thus, I believe that they are reliable, and that they give suggestions for future speech recognition research.

A-1 Outlines of the System

Fig. A-1 indicates a block diagram of LISTEN, showing the two stages for preliminary learning of speaker differences and for spoken words recognition in real time. The latter stage consists of both a phoneme classification and word recognition component. The Acoustic Processor is the same as described in section II-2-2, and the matching method is the same as in Chapter III.

First, a new user of the system is required to clearly utter the five vowels (/a/, /i/, /u/, /e/, /o/) and the syllabic nasal (/N/) for preliminary learning of his voice. Since the spectral variation of

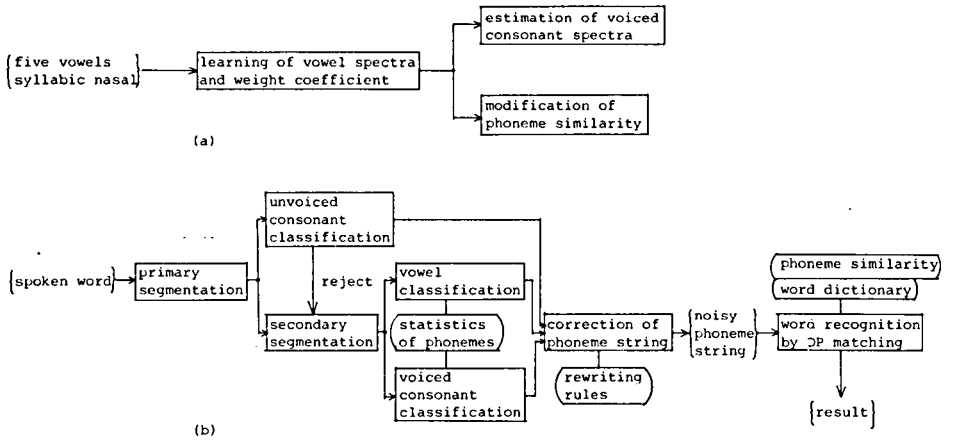


Fig. A-1 Configuration of the LISTEN spoken words recognition system.
(a) preliminary learning of speaker differences.
(b) spoken words recognition.

the syllabic nasal is very large among inter-speakers, it is learned as well as the vowels. Note that it is uttered as /uN/, because we cannot utter the syllabic nasal in isolation. The learning method is the same as described in section II-8-2. The learned spectrum of the syllabic nasal is used for estimating voiced consonant spectra.

Second, the new user can utter any digit after the learning. The primary segmentation is done in the same way as described in section II-5. In order to recognize speech in real time, the system employs a simple algorithm in the part of phoneme classification.

In processing vowels, the system computes the distance between each frame in the voiced parts and each of the six reference patterns: the five vowels and syllabic nasal (the syllabic nasal is treated like the five vowels in the following process). Cityblock distance is used as a measure. The calculation of this distance does not have to be multiplicative. Next, the two nearest neighbors are selected as candidate phonemes and put into order by applying the corresponding linear discriminant functions. The voiced consonant parts, detected by a change in spectrum or power, are recognized by Euclidean distance. These phoneme recognition procedures can all be achieved for each frame within a sampling interval of 10ms.

By rewriting rules, a recognized phoneme sequence in the voiced parts is smoothed and merged. Then, output of the phoneme classification process is a sequence of segments, each consisting of a 4-tuple: the first candidate among the phonemes, the second candidate, the degree of confidence of the first candidate, and the segment duration. Finally, the recognized phoneme sequence is passed to the stage of word recognition.

This system was implemented on mini-computer (MELCOM-70, cycle time=0.8μs, core memory=24K words). The algorithms were programmed with an assembly language. This program occupied about 4.5K words (4500 steps), and the work area about 5.5K words. The matching time required between a phoneme sequence and lexical entry in the word dictionary was about 8ms. This system can illustrate a recognition result on a graphic display within 100ms for a small vocabulary.

A-2 Experimental Results

Using this method, we conducted experiments to recognize isolated spoken digits. Test materials consisted of 2000 digits, each of ten digits was spoken five times by each of 40 male speakers. Ten of these speakers provided the training samples for the system design, although the designing materials were not digits. The ages of the speakers ranged from 21 to 26. It should be noticed that all the experimental results mentioned below are based on the 'forced decision'.

Three kinds of experiments carried out were:
Experiment a ..... All spoken digits were recognized by common reference patterns, that is, without learning speaker differences.
Experiment b ..... All were recognized by personal reference patterns obtained by preliminary learning using the previously uttered vowels and syllabic nasal in isolation.
Experiment c ..... All were recognized by personal reference patterns

Table A-1 Confusion matrices for 2000 digits of 40 male speakers.

(a) without learning: by common reference patterns for 10 male speakers.

in \ out	1	2	3	4	5	6	7	8	9	0
1	199									
2	1	197								2
3	2		175	1					22	
4				167	25				6	2
5				1	199					
6						196				
7					10	1	189			
8	1					6		192		
9		1							199	
0										200

(b) preliminary learning by isolated vowel spectrum.

in \ out	1	2	3	4	5	6	7	8	9	0
1	199									
2		195		1						4
3	2		194		1			1	2	
4				175	21		1		2	1
5					200					
6						196				
7					2		198			
8						1		198		
9		1							199	
0										200

(c) preliminary learning by some spoken digits.

in \ out	1	2	3	4	5	6	7	8	9	0
1	199									
2		199		1						
3	3		193				2		2	
4				188	10				2	
5				2	198					
6						196				
7					4		196			
8	1					5		193		
9									200	
0										200

which were obtained by preliminary learning using some digits. The spectrum of /a/ was learned by /a_1/ or /a_2/ in "na_1na_2", /i/: "ni", /e/: "rei", /o/: "go", /N/: "yon", respectively. The spectrum of /u/, however, was learned by an isolated vowel /u/.

Table A-1 sums up the experimental results. The recognition rates for *Experiments a, b* and *c* were about 95.8%, 98.0% and 98.4%, respectively. The relationship between the number of speakers and recognition rate is summarized in Table A-2. Preliminary learning had the special effect of normalizing the spectrum of a speaker whose voice strikingly contrasted with the reference voice.

Table A-2 Relation between the number of speakers and the correct rate of digit's recognition.

a: without learning.

b: preliminary learning by isolated vowel spectrum.

c: preliminary learning by some spoken digits.

	a	b	c
100%	15	19	22
98	9	11	9
96	1	5	4
94	4	2	3
92	3	1	0
90	3	1	2
less 90	5	1	0

160項欠

APPENDIX B

WORD DICTIONARY AND SPOKEN GRAMMAR OF 'COMPUTER NETWORK'

B-1 Word Dictionary

The following two pages show the word dictionary of the task 'Computer Network'. The symbol '&' denotes both the symbols '+' and '-' in section III-4-2, the symbol ':' denotes '-'. The phoneme symbol 'X' denotes /N/, and 'V' denotes /$\tilde{g}$/. The column of the word length indicates the number of phonemes in the word.

B-2 Spoken Grammar

The spoken grammar is shown in the following page of the word dictionary. The production of a word class is denoted by the special symbol ':::' and it is rewritten by the Backus-Normal Form.

Word Dictionary of 'Computer Network'.

& denotes '+' and '-'.
: denotes '-'.

```
symbol  word
  ↓      ↓
01  ichi  [one]            word length      duration (×10ms)
      &                        ↓
    I.CI ← main phoneme        04          55 ← maximum
    C C  ← sub phoneme                     10 ← minimum
    9 9  ← weight
    5 5  ← weight

02  ni  [two]
    NI                         03          30
    GA                                     10
    99
    00

03  san  [three]
    SAX                        04          55
                                           20

04  yon  [four]
    YOX                        04          45
    V                                      15
    9
    9

05  go  [five]
    GO                         02          33
                                           10

06  roku  [six]
      &
    RO.KU                      05          45
    P K *                                  13
    8 9 9
    5 5 9

07  nana  [seven]
      +
    NANA                       04          55
    GAAX                                   20
    9388
    8555

08  hachi  [eight]
        &
    HA.CI                      05          33
     XC C                                  15
     89 9
     59 5

09  kyu  [nine]
    /KYU                       04          50
    CCU                                    20
    999
    555

10  zyu  [ten]
    ZYU                        04          50
    OH                                     20
    99
    05

VA  aiteiruka  [Is free  ?]
            +
    AI.TEIRU.KA                11          110
      I     K                              50
      9     9
      5     5

VB  tsukatteiruka  [Is using  ?]
     &       +
    /CU.KA-TEIRU.KA            15          145
    C CK       4                           65
    9 99       9
    9 99       5

VC  hashitteiruka  [Is being executed  ?]
      :       +
    HASI-TEIRU.KA              13          145
      S       K                            65
      9       9
      9       5

VD  tsunagatteiruka  [is  connecting?]
     &  +      +
    /CUNAVA-TEIRU.KA           16          170
    C C XA      4                          75
    9 9 88      9
    9 9 55      5
```

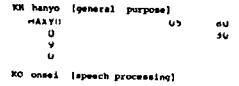
```
VE  tsunage  [Connect]
      &
    /CUNAVE                    07          75
    C C XG                                 25
    9 9 89
    9 9 55

VF  yomikome  [Read]
    YOMI.KOME                  09          95
    O   K                                  40
    9   9
    0   5

VG  rodoseyo  [Load]
      +  +
    RODOSEYO                   08          120
    0 U   E                                55
    8 8   9
    5 5   0

VH  warikome  [Interrupt]
    WARI.KOME                  09          95
    H   K                                  40
    9   9
    0   5

VI  hazimeyo  [Begin]
        +
    HAZIMEYO                   08          95
         E                                 40
         9
         0

VJ  tomeyo  [Halt]
       +
    /TOMEYO                    07          75
    T   E                                  30
    9   9
    5   0

VK  makimodose  [Rewind]
       &  +
    MA.KIMODOSE                11          110
     KC*  U  S                             45
     999  8  8
     955  5  0

VL  haitteiruka  [Is  switching on?]
          +
    HAI-TEIRU.KA               12          140
           K                               60
           9
           5

VM  ireyo  [Switch on]
     +
    IREYO                      05          65
      E                                    25
      9
      0

VN  kire  [Switch off]
     &
    /KIRE                      05          50
    KC*                                    10
    999
    555

VO  shimese  [Display]
     :
    SIMESE                     06          85
     S   S                                 30
     9   8
     0   0

VP  kakikome  [Write]
       &
    /KA.KI.KOME                11          100
    K  KC*K                                35
    9  9999
    5  9555

VQ  uchidase  [Print]
      &
    U.CIDASE                   08          90
     C C   S                               30
     9 9   8
     9 5   0

VR  hazuse  [Take off]
       +
    HAZUSE                     06          65
       S S                                 30
       8 8
       0 0

VS  kakeyo  [Set]
    /KA.KEYO                   08          65
    K  K  E                                25
    9  9  9
    5  9  0

VT  torikesu  [cancel]
              :
    /TORI.KESU                 10          100
    T    K   S                             35
    9    9   9
    5    9   9

VU  suru  [do]
      +
    SURU                       04          45
     S                                     15
     9
     5

00  rei  [zero]
     &
    REI                        03          40
    P E                                    10
    9 9
    0 0
```

```
TW  taipuraita  [tele type writer]
          &
    /TAI.PURAI.TA              13          125
    T   P PP   T                           40
    9   9 98   9
    5   5 95   5

LP  housukuinsi  [line printer]
          &
    /KOSU.KUIXZI               12          155
    K     K *                              40
    9     9 9
    5     5 9

MT  zikitepu  [magnetic tape]
      &   &
    ZI.KI.TE.PU                11          105
    C KC*I  P *                            40
    9 9999  9 9
    0 9555  5 9

CT  kasettozikitepu  [cassette magnetic tape]
             +  &    &
    /KASE-TOZI.KI.TE.PU        20          220
    K        ZC KC*I  P *                  80
    9        99 9999  9 9
    9        90 5555  5 9

MD  zikidoramu  [magnetic drum]
      &
    ZI.KIDORAMU                11          110
    C KC*                                  45
    9 999
    0 955

MK  zikidisuku  [magnetic disk]
      &   :  &
    ZI.KIDISU.KU               12          105
    C KC*   SK *                           40
    9 999   99 9
    0 955   99 9

PR  kamitepuyomitori  [paper tape reader]
                &  +
    /KAMI.TE.PUYOMI.TORI       21          215
    K    I  P *IV   T                      80
    9    9  9 999   9
    5    5  5 990   5

CR  kadoyomitori  [card reader]
            +
    /KADO-YOMI.TORI            15          190
    K     YV   T                           65
    9     99   9
    5     90   5

PP  kamitepusenko  [paper tape puncher]
               &
    /KAMI.TE.PU.SEX.KO         18          215
    K    I  P *S   K                       80
    9    9  9 99   9
    5    5  5 99   5

CP  kadosenko  [card puncher]
    /KADO-SEX.KO               12          185
    K     S   K                            65
    9     9   9
    5     9   5

CD  mozihyozi  [character display]
    MOZI-HYOZI                 10          145
        H I                                65
        9 9
        9 5

GD  gazohyozi  [cathode ray tube]
    GAZO-HYOZI                 10          155
        H I                                70
        9 9
        9 5

TV  zikisaihyozi  [color display]
     :   &+
    SI.KISAI-HYOZI             14          165
    SKC*     H O                           75
    9999     9 9
    9959     9 5

XY  nizigenhyozi  [XY-plotter]
     +
    NIZIVEX-HYOZI              13          170
     I G   H O                             75
     8 9   9 9
     5 5   9 5

FS  gazonyuryoku  [flying spot scanner]
             +  +  &
    GAZO-NYURYO.KU             14          170
        N C  O K *                         70
        9 9  9 9 9
        9 5  5 5 9

GP  zahyonyuryoku  [graf pen]
              +  +  &
    ZAHYO-NYURYO.KU            15          170
    O  O N U  O K *                        75
    9  9 9 9  9 9 9
    0  5 9 5  5 5 9

AD  onseinyuryoku  [A/D converter]
         :   +   +  &
    OXSEI-NYURYO.KU            15          185
        EN U  U K *                        80
        99 9  9 9 9
        59 5  5 5 9

DA  onseisyutsuryoku  [D/A converter]
         :  ++   &  +  &
    OXSEI-SYU.CURYO.KU         18          180
        ES SSC C O K *                     80
        99 999 9 9 9 9
        59 559 9 5 5 9
```

KK kokan [IMP].

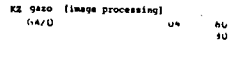
KC cyuo [central]

KH hanyo [general purpose]

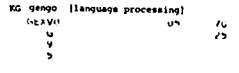


KO onsei [speech processing]

KI gazo [image processing]



KG gengo [language processing]

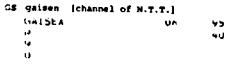


LK ko [A]

LT otsu [B]

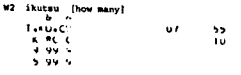
LN hei [C]

GS gaisen [channel of N.T.T.]



W1 darega [who]

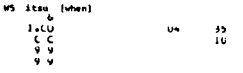
W2 ikutsu [how many]



W3 dorega [which]

W4 doreo [which]

W5 itsu [when]



JM made [to]

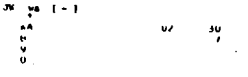
JG ga [-]

JN no [of]

JO o [-]

JT to [and]

JW wa [-]



JD de [at]

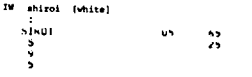
JI ni [to]

JH e [to]

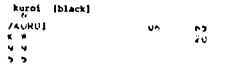
JK kara [from]

IB aoi [blue]

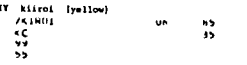
IW shiroi [white]



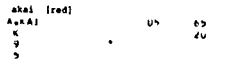
IK kuroi [black]



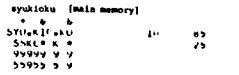
IY kiiroi [yellow]



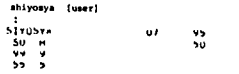
IR akai [red]



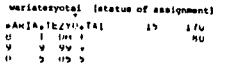
SK syukioku [main memory]



SY shiyosya [user]

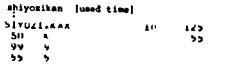


YR wariatesyotai [status of assignment]

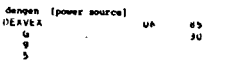


Z4 yo [four]

SZ shiyozikan [used time]



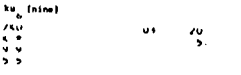
DG dengen [power source]



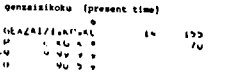
YY yoyaku [reservation]

ZB zyobu [job]

Z9 ku [nine]



GZ genzaizikoku [present time]



DT deta [data]

PG puroguramu [program]

SB seiqyobun [control statement]

SR syuryozikoku [ended time]

YE yusensyuni [priority]

ZI zi [o'clock]

HH han [half]

ZK zikoku [time]

BN ban [number]

AM gozen [morning]

PM gogo [afternoon]

ST syotai [status]

KS keisanki [computer]

SC sochi [device]

Spoken Grammar of 'Computer Network'.

additional information

symbol	subscript
0	
1	1
2	2
+	3
4	4
8	5
-	6
A	1,3
B	2,3
C	1,2,3

```
P10::=Z90Q1
P10::=01+SC0070
P20::=G20J04Z68JW4+W0
P30::=G30JT4Q30R14
P40::=G20JN4R10SC0F20JK4060WD0+S8J04
P50::=Z80WS.R14
P60::=G20JN4D2ASC0+S8BN4JG4
P70::=Q80DG8R14
P80::=G20JN4R10WJ0G40J04
P90::=G20JN4D8CSC0/S8BN4WJ0G60.S8BN4JG4
PA0::=G20JN4D8CSC0+S8BN4JK4D88J04
PB0::=G50
PC0::=G50
PC0::=Z80WS8JN4YZ8J04WS8J14
Q10::=G70
Q10::=JN4T80
Q20::=KS0WK9
Q30::=G20
Q30::=T10
Q30::=GS9
Q40::=SY0WL8JN4U90SZ8
Q40::=Z80WS8JN4T38
Q40::=G28
Q40::=G20JN4T20
Q40::=WD0WS8
Q50::=SY0WL8JG4WG0T40U40JK4WG0T40U40JM4Q80YY8JG4
Q60::=Z10D22
Q60::=D62
Q70::=JW4U10
Q80::=G20JN4U20
U90::=G80
U90::=
Q80::=D1+SC8JN4
Q90::=KS0WK1
T10::=D8ASC0U50
T10::=T7ASC0
T20::=SK8JN4YRW
T20::=U80SC8JN4YY07T8
U80::=LVA
U60::=
T30::=SR0
T30::=YZ0
T40::=WZ0Z14
T40::=1000S4Z14
T50::=D8ASC0T60
T60::=+S8BN4
T70::=D10
T70::=D30
T70::=D40
T70::=D50
T70::=D70
T80::=T7ASC0Q70
T80::=D8ASC0T60
T90::=D5ASC0
TA0::=D4ASC0
TB0::=T60JW4
TB0::=JW4U70
U10::=R10
U10::=
--0::=
U20::=T10JN4
U20::=
U30::=T60
U30::=
U40::=HN4
U40::=
U50::=010
U50::=020
U50::=
U60::=G90WJ0
U60::=
U70::=R20
U70::=R20
U70::=
**0
DV0:::TW0DA0     DV::=TW|LP|MT|CT|MD|MK|PR|CR|PP|CP|CD|GD|TV|XY|FS|GP|AD|DA
D10:::GP0DA0     D1::=GP|AD|DA
D20:::MT0CT0     D2::=MT|CT
D30:::PR0CR0     D3::=PR|CR
D40:::TW0LP0     D4::=TW|LP
D50:::CD0TV0     D5::=CD|GD|TV
D60:::MD0MK0     D6::=MD|MK
D70:::PP0CP0     D7::=PP|CP|XY|FS
D70:::XY0FS0
D80:::MT0MK0     D8::=MT|CT|MD|MK
WS0:::000000     WS::=00|01|02|03|04|05|06|07|08|09
WS0:::010090
WL0:::LK0LH0     WL::=LK|LT|LH
WK0:::KK0KG0     WK::=KK|KC|KH|KO|KZ|KG
WZ0:::000000     WZ::=00|01|02|03|Z4|05|06|07|08|Z9
WZ0:::010030
WZ0:::Z40Z40
WZ0:::050080
WZ0:::Z90Z90
WD0:::DT0SB0     WD::=DT|PG|SB
WG0:::AM0PM0     WG::=AM|PM
WI0:::IB0IR0     WI::=IB|IW|IK|IY|IR
WJ4:::JI4JH0     WJ::=JI|JH
WV0:::VA0VU0     WV::=VA|VB|VC|VD|VE|VF|VG|VH|VI|VJ|VK|VL|VM|VN|VO|VP|VQ|VR|VS|VT|VU
WW0:::W20W30     WW::=W2|W3
```

APPENDIX C

DETAILED EXAMPLES OF PARSING PROCEDURE

This appendix contains five detailed examples of parsing procedure. The input sentences and parsing parameters are given below.

No. 1: zyū ichi tasu ni kakeru san (z 1 + 2 × 3; 11+2×3).
α=2, β=5, γ=10, m=2 (refer to section VI-2-2).

No. 2: zyū ni gatsu san zyū ichi nichi suiyōbi (10 02 GT 03 10 01 NC SU; Wednesday, December 31).
α=5, β=5, γ=5, m=2 (refer to section VI-3-3).

No. 3: Keisanki cyūo-de zyobu-wa ikutsu hashitteiruka?(KS KC JD ZB W2 VC; How many jobs are being executed in the general purpose computer?).
α=2, β=5, γ=10, m=2, n=5, l=1.

No. 4: Keisanki gengo-no zikidoramu sōchi-wa dorega aiteiruka? (KS KG JN MD SC JW W3 VA; Which device is free among the magnetic drums of the language-processing computer?).
α=2, β=5, γ=10, m=2, n=5, l=1 (refer to section VII-4).

NO. 5: Keisanki cyūo-no zikidisuku sōchi san-ban kara keisanki gazō-e dēta yon-o rōdoseyo (KS KC JN MK SC 03 BN JK KS KZ JH DT 04 JQ VG; Load the 4th datum from the 3rd magnetic disk device of the central computer to the image-processing computer).
α=2, β=5, γ=12, m=2, n=5, l=2 (refer to section III-5-2).

The head character of a partial sentence denotes its status.

K = a partial sentence not to be parsed.

L = a partial sentence to be parsed.

¥ = a complete sentence and a partial sentence to be parsed.

= a complete sentence.

Q = final result.

No. 1: speaker SK

```
****
****
****
SK000101

0*******1*******2*******3*******4*******5*******6*******7*******0*******1*******2*******3*******4*******5*******6*******
.XHIUI.CI.P S.XII.PA.PERUUU.CAX/      <------------ first candidate
PUIXIUPHU-H H.UXUFHOPHIXXGIPSXR-      <------------ second candidate
612514929/5 794 8924058576550656      <------------ degree of confidence
428802929.2 8922090880802848060


                         --------          01 ROUND          --------

L 0000 *   0 0 0

WORD NO.     La  P   M   2   3   4   5   6   7   8   9   S-  0   F-  Z   1   <------------ predicted words
ECSR(000)    37  37 037 037 037 037 037 037 037 037 037 037 037 037 037 037
SCORE        85  61  82  95  75  87  84  88  86  87  87  75  88  83  91  89
HEAD         0      001 001  01 001 001 001 001 000 000 001 001 000 001 001
TAIL         14     007 004  04 004 004 004 005 005 004 005 005 004 004 005
AV. SCORE  5 0 8100820095007500870084008800860087008700750088008300910089000


                         --------          02 ROUND          --------

K 8900     1 0 89      1       <------------ partial sentenc
L 9500 8   1 0 95      2

WORD NO.     S-  F-  Z   .   *   /   +   -
ECSR(004)    37  37 037 037 037 037 037 037
SCORE        74  85  89  87  84  74  90  97
HEAD         07  07 004 006 007 004 007 007
TAIL         14  12 005 010 017 010 014 012
AV. SCORE  0 0 8700920089508800000091009450

K 8800 0   1 0 88      6
K 8800 0   1 0 88      0
L 9100 >   1 0 91      2

WORD NO.     1   2   3   4   5   6   7   8   9   .   *   /   +   -
ECSR(004)    37  37 037 037 037 037 037 037 037 037 037 037 037 037
SCORE        95  96  80  86  85  88  84  87  91  87  84  74  90  97
HEAD         05  04 007 004 004 004 004 005 006 006 007 004 007 007
TAIL         1   05 010 005  05 007 005 010 010 010 017 010 014 012
AV. SCORE  9 0 9550840088508800895087508900895087508600000089009250


                         --------          03 ROUND          --------

K 9250   2 0185        2   -
L 9450 6 2 0185        2   -

WORD NO.     P   M   2   3   4   5   6   7   8   9   S-  0   F-  2   1
ECSR(012)    37  37 037 037  37 037 037 037 037 037 037 037 037 037 037
SCORE        75  75  90  79  83  86  88  84  92  86  79  84  92  85  87
HEAD         15  16 016 012 016 012 016 016 012 012 014 016 012 013 016
TAIL         22  22 020 014 020 013 022 020 014 016 017 020 016 016 022
AV. SCORE  5 6656669166893389339166910089669366916689008966936691339066

K 8900   2 0178        2   F-
K 9200   2 0184        2   7
K 9350 5 2 0187        7   2
K 8950 0 2 0179        2   .
K 9100   2 0182        2   +
K 8950   2 0179        7   6
K 8950   2 0179        2   9
L 9500 5 2 0195        2   1

WORD NO.     .   *   /   +   -
ECSR(01.)    37  37 037 037 037
SCORE        84  9.  77  98  92
HEAD         11  11 012 011 012
TAIL         15  17 016 014 016
AV. SCORE  9 35933389009600940 0


                         --------          04 ROUND          --------

K 9250 0 2 0185        2   -
K 9366 0 3 0281        2   -   F-
L 9400 8 3 0282        2   1   .

WORD NO.     P   M   2   3   4   5   6   7   8   9   S-  0   F-  2   1
ECSR(016)    37  37 037 037 037 037 037 037 037 037 037 037 037 037 037
SCORE        84  79  97  75  88  88  93  88  92  90  70  89  97  86  92
HEAD         21  16 016 022 017 022 022 016 022 022 022 016 022 017 017
TAIL         3   4  020 030 020 023 025 020 025 032 030 020 025 020 022
AV. SCORE  . 759 2594758850925091759300925092759225000092759400920093500

K 9200 0   2 0184      2   7
K 9350     2 0187      7   2
K 9166 0   3 0275      2   -   5
K 9366 0   3 0281      2   -   6
K 9166 0   3 0275      2   -   9
K 9333     3 028       2   1   .
L 9600 6   3 0288      2   1   +

WORD NO.     P   M   2   3   4   5   6   7   8   9   S-  0   F-  2   1
ECSR(014)    37  37 037 037 037 037 037 037 037 037 037 037 037 037 037
SCORE        82  77  97  84  89  86  92  89  86  88  86  89  86  86  91
HEAD         15  16 016 014 016 016 016 016 014 015 014 016 014 016 016
TAIL         22  22 020 017 020 017 022 020 022 020 017 020 022 020 022
AV. SCORE  9 5 91259825930094259350950094259350940093509425935093509475
```

```
                                          --------              05 ROUND        --------

¥ 0000 9 04 048         Z   1   +   6

WORD NO.      S- PFM  Z   .   +   -   *   /   =
ECSR(022)     37  37 037 037 037 037 037 037 037
SCORE         84  95  87  93  81  94  92  85  85
HEAD          25  23 023 024 022 022 022 023 025
TAIL          31  26 031 030 030 025 032 032 031
AV. SCORE 9 6 9300934094409220948094409300928O

K 9366 0  3 0281        2   -   F-
K 9400 0 04 0376        Z   1   -   F-
K 9475 0 04 0379        Z   1   +   1
K 9350 0 02 0187        Z   2
K 9425 0 04 0377        Z   1   +   7
K 9366 0 03 0281        2   -   8
K 9425 0  4 0377        Z   1   +   4
K 9333 0  3 028         Z   1   *
¥ 0000 9  4 0485        Z   1   +   2

WORD NO.      S-  F-  Z   .   +   -   *   /   =
ECSR(020)     37  37 037 037 037 037 037 037 037
SCORE         71  97  84  90  90  95  99  81  76
HEAD          22  22 022 022 021 022 021 022 022
TAIL          3.  25 030 030 030 025 032 032 026
AV. SCORE 9 0 96409380950095009600968093209220

                                          --------              06 ROUND        --------

K 9500 0 05 0475        Z   1   +   6  PFM
K 9500 0 05 0475        Z   1   +   2   .
K 9400 0 04 0376        Z   1   -   F-
K 9475 0 04 0379        Z   1   +   1
K 9600 0 05 048.        Z   1   +   2   -
K 9425 0 04 0377        Z   1   +   7
¥ 0000 + 05 0482        7   1   +   2   F-

WORD NO.      2   3   4   5   6   7   8   9   Z   1   .   +   -   *   /   =
ECSR(025)     37  37 037 037 037 037 037 037 037 037 037 037 037 037 037 037
SCORE         84  81  90  85  90  84  84  83  90  86  77  73  81  78  86  89
HEAD          25  25 026 025 026 026 026 025 025 030 025 025 026 025 026 025
TAIL          32  30 030 030 034 032 034 032 032 034 030 034 034 036 032 031
AV. SCORE 9 0 9-8395339450953394339433941695339433931600009383933394669516

K 9425 0 04 0477        Z   1   +   4
L 9680 6 05 0484        Z   1   +   2   *

WORD NO.      P   M   2   3   4   5   6   7   8   9   S-  0   F-  Z   1
ECSR(030)     37  37 037 037 037 037 037 037 037 037 037 037 037 037 037
SCORE         74  77  82  98  82  77  90  87  82  89  91  82  79  86  96
HEAD          34  32 034 034 034 034 032 034 032 033 034 034 034 034 032
TAIL          36  36 036 036 036 036 034 036 034 036 036 036 036 036 034
AV. SCORE 9 839350943397009433935095669516943395509583943393839500966б

K 9500 0 05 0475        Z   1   +   2   +

                                          --------              07 ROUND        --------

K 9550 0 06 0573        Z   1   +   2   *   9
K 9583 0 06 0575        Z   1   +   2   *   S-
K 9516 0 06 0571        Z   1   +   2   F-  =
K 9566 0 06 0574        Z   1   +   2   *   6
K 9600 0 05 048         Z   1   +   2   -
¥ 9700 5 06 0582        7   1   +   2   *   3

WORD NO.     S- SF-  Z   .   =
ECSR(036)     37  37 037 037 037
SCORE         43  34  45  15  16
HEAD          35  36 036 036 036
TAIL          37  37 037 037 037
AV. SCORE 0 000 0000000000000

K 9533 0 06 0572        Z   1   +   2   F-  Z
K 9533 0 06 0572        Z   1   +   2   F-  4
¥ 9616 2 06 058         7   1   +   2   *   1

WORD NO.      .   =
ECSR(034)     37  37
SCORE         79  79
HEAD          34  34
TAIL          36  36
AV. SCORE 9 149414

K 9533 0 06 0572        Z   1   +   2   F-  6

                                  ********    FINAL RESULT    ********

Q 9700 0 06 0582        Z   1   +   2   *   3
```

```
****
****       No. 2: speaker SM
****
SIMA  29

0*******1*******2*******3*******4*******5*******6*******7*******0*******1*******2*******3*******4*******5*******6*******
/CYUEAX.CU-SCAVZEESAHU.CI-SUYOCBI.  <---- first candidate
-SIEIEEPSG/CACUDUICUREPSH.CEICCDBP  <---- second candidate
6300021210Z.0521350101610930031009  <---- degree of confidence
0402288680480040282048000902026809

                                 --------          01 ROUND        --------

L 0000 * 00 0 00

WORD NO.   028 10  018 038 058 078 088 048 068 098   <------   predicted words
ECSR(000)  041  41 041 041 041 041 041 041 041 041
SCORE       91  94  00  91  89  88  00  95  85  93
HEAD        01  01 000 001 001 001 000 001 001 000
TAIL       011  03 000 011 011 011 000 011 014 011
AV. SCORE 910 9400000091008900800000009500850093000

                                 --------          02 ROUND        --------

L 9300 > 02 0186       098 GT4  <------  partial sentence

WORD NO.   038 -1  -2  -3  -4  -5  -6  -7  -8  -9  10  HA  028 10
ECSR(011)  041 041 041 041 041 041 041 041 041 041 041 041 041 041
SCORE       94  89  85  00  00  00  00  81  00  81  84  00  84  88
HEAD        13  12 013 011 011 011 011 013 011 012 012 011 013 013
TAIL        2.  33 021 011 011 011 011 021 011 030 021 011 020 016
AV. SCORE 935 9166973300000000000000089000000890090000000008509133

L 9100   02 0182       028 GT4

WORD NO.   -1  -2  -3  -4  -5  -6  -7  -8  -9  10  HA  028 10
ECSR(011)  041 041 041 041 041 041 041 041 041 041 041 041 041
SCORE       89  85  00  00  00  00  81  00  81  84  00  84  88
HEAD        12  13 011 011 011 011 013 011 012 012 011 013 013
TAIL        3.  21 011 011 011 011 021 011 030 021 011 020 016
AV. SCORE 9.34e90000000000000000008766000876688660000875090000

L 9400 3 01 0 94       10

WORD NO.   018 028 GT4
ECSR(003)  041 041 041
SCORE       9.  95  95
HEAD       004  03 003
TAIL       014 011 011
AV. SCORE 9133494669450

L 9100 > 02 0152       038 GT4

WORD NO.   038 -1  -2  -3  -4  -5  -6  -7  -8  -9  10  HA  028 10
ECSR(011)   41 041 041 041 041 041 041 041 041 041 041 041 041 041
SCORE       94  89  85  00  00  00  00  81  00  81  84  00  84  88
HEAD        14  12 013 011 011 011 011 013 011 012 012 011 013 013
TAIL        2.  30 021 011 011 011 011 021 011 030 021 011 020 016
AV. SCORE 925 9 3389000000000000000000876600087668866000087509000

L 9500 > 02 019.       048 GT4

WORD NO.   038 -1  -2  -3  -4  -5  -6  -7  -8  -9  10  HA  028 10
ECSR(011)   41  41 041 041 041 041 041 041 041 041 041 041 041 041
SCORE       94  89  85  00  00  00  00  81  00  81  84  00  84  88
HEAD        13  12 013 011 011 011 011 013 011 012 012 011 013 013
TAIL        2.  30 021 011 011 011 011 021 011 030 021 011 020 016
AV. SCORE 945.9300916600000000000000903300090339133000069509266

                                 --------          03 ROUND        --------

L 9350 1 04 0374       098 GT4 038 104

WORD NO.   NC4
ECSR(020)  041
SCORE       93
HEAD        02
TAIL        22
AV. SCORE 934.

L 9450 1 04 0378       048 GT4 038 104

WORD NO.   NC4
ECSR(020)  041
SCORE       93
HEAD        22
TAIL        22
AV. SCORE 942.

L 9466 > 03 0284       10  028 GT4

WORD NO.   038 -1  -2  -3  -4  -5  -6  -7  -8  -9  10  HA  028 10
ECSR(011)  041 041 041 041 041 041 041 041 041 041 041 041 041 041
SCORE       94  89  85  00  00  00  00  81  00  81  84  00  84  88
HEAD        13  12 013 011 011 011 011 013 011 012 012 011 013 013
TAIL        2.  33 021 011 011 011 011 021 011 030 021 011 020 016
AV. SCORE 944 93259225000000000000000091250000912592000000090409300

L 9450 > 02 0185       10  GT4

WORD NO.   038 -1  -2  -3  -4  -5  -6  -7  -8  -9  10  HA  028 10
ECSR(011)  041 041 041 041 041 041 041 041 041 041 041 041 041 041
SCORE       94  89  85  00  00  00  00  81  00  81  84  00  84  88
HEAD        13  12 013 011 011 011 011 013 011 012 012 011 013 013
TAIL        2.  33 021 011 011 011 011 021 011 030 021 011 020 016
AV. SCORE 9425926691330000000000000090000000900091000000089259233

L 9300 / 03 0279       048 GT4 -1

WORD NO.   NI  GE  KA  SO  MO  KI  DO
ECSR(022)  041  41 041 041 041 041 041
SCORE       87  86  92  98  91  94  92
HEAD        23  23 031 032 023 031 032
TAIL        34  34 040 040 040 040 040
AV. SCORE 915 91259275942592509325927 5
```

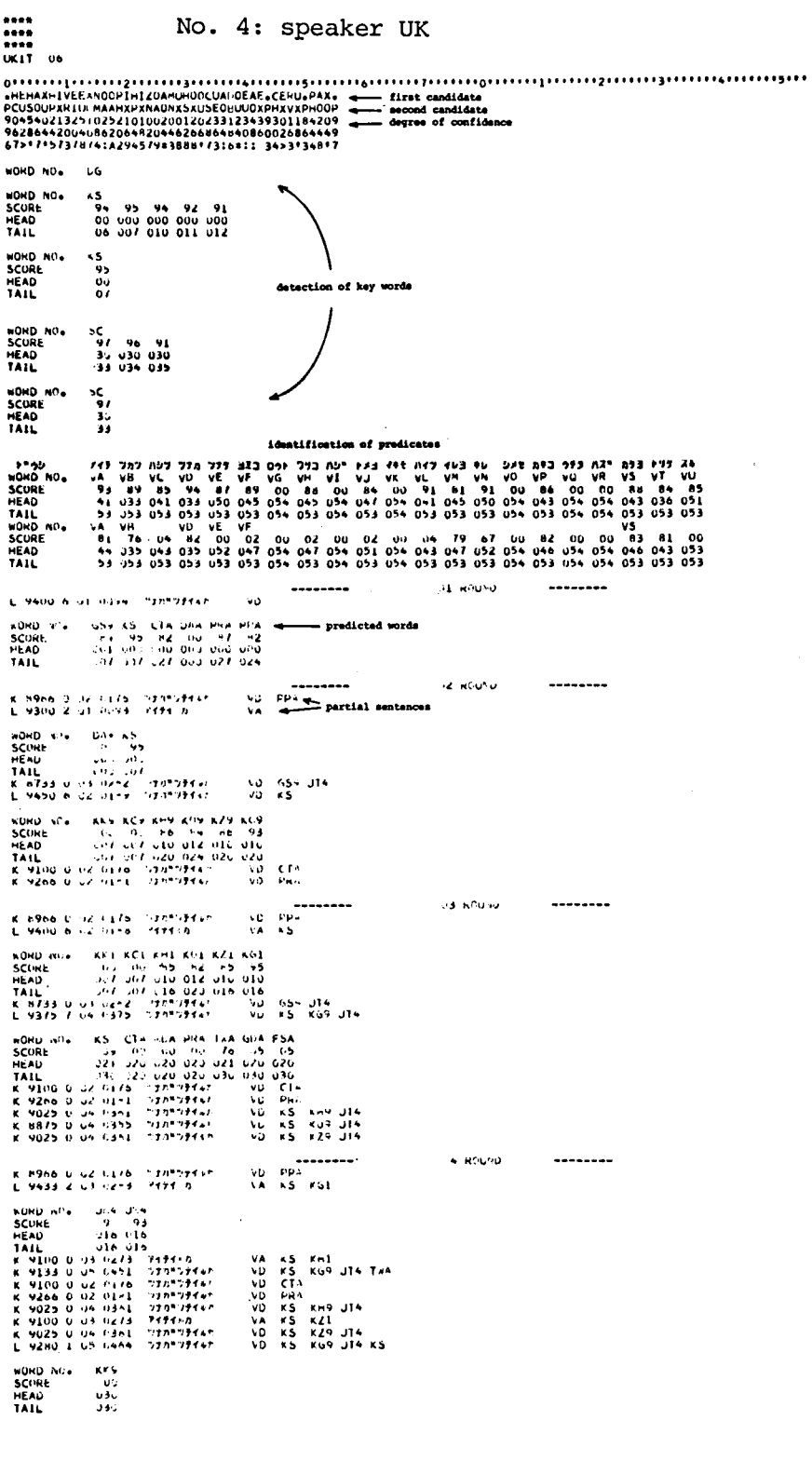
```
                                        --------          04 ROUND        --------

L 9340 / 05 0467      098 GT4 038 104 NC4

WORD NO.    NI  GE  KA  SU  MO  KI  DO
ECSR(022)  041 041 041 041 041 041 041
SCORE       88  87  87  86  92  92  83
HEAD       023 023 026 022 023 026 023
TAIL       034 034 040 030 040 040 030
AV. SCORE 925 923918923169231692669166

L 9420 / 05 0471      048 GT4 038 104 NC4

WORD NO.    NI  GE  KA  SU  MO  KI  DO
ECSR(022)  041 041 041 041 041 041 041
SCORE       88  87  87  86  92  92  83
HEAD       023 023 026 022 023 026 023
TAIL       034 034 040 030 040 040 030
AV. SCORE 931693009250928493833933923 3

L 9440 / 05 0472      10  028 GT4 038 104

WORD NO.    018 NC4
ECSR(020)  041 041
SCORE        94  93
HEAD       021 020
TAIL       033 022
AV. SCORE 94289416

L 9425 / 04 0377      10  GT4 038 104

WORD NO.    018 NC4
ECSR(020)  041 041
SCORE        94  93
HEAD       021 020
TAIL       033 022
AV. SCORE 94169400

# 9425 0 04 0377      048 GT4 -1  SU

                                        --------          05 ROUND        --------

L 9400 / 05 0472      10  GT4 038 104 NC4

WORD NO.    NI  GE  KA  SU  MO  KI  DO
ECSR(022)  041 041 041 041 041 041 041
SCORE       88  87  87  86  92  92  83
HEAD       023 023 026 022 023 026 023
TAIL       034 034 040 030 040 040 030
AV. SCORE 930.928392339266936693169216

L 9428 / 07 0665      10  028 GT4 038 104 018 NC4

WORD NO.    NI  GE  KA  SU  MO  KI  DO
ECSR(030)  041 041 041 041 041 041 041
SCORE       00  00  92  98  00  94  92
HEAD       030 030 031 032 030 031 032
TAIL       030 030 040 040 030 040 040
AV. SCORE 0 0.0.009400947500094259400

L 9416 / 06 0565      10  028 GT4 038 104 NC4

WORD NO.    NI  GE  KA  SU  MO  KI  DO
ECSR(022)  041 041 041 041 041 041 041
SCORE       88  87  87  86  92  92  83
HEAD       023 023 026 022 023 026 023
TAIL       034 034 040 030 040 040 030
AV. SCORE 932893149271930093859342925 7

L 9416 / 06 0565      10  GT4 038 104 018 NC4

WORD NO.    NI  GE  KA  SU  MO  KI  DO
ECSR(030)  041 041 041 041 041 041 041
SCORE       00  00  92  98  00  94  92
HEAD       030 030 031 032 030 031 032
TAIL       030 030 040 040 030 040 040
AV. SCORE 0.0000093859471000094149385

# 9425 0 04 0377      048 GT4 -1  SU

                                        --------          06 ROUND        --------

# 9425 0 08 0754      10  028 GT4 038 104 018 NC4 KI
# 9475 0 08 0758      10  028 GT4 038 104 018 NC4 SU
# 9414 0 07 0659      10  GT4 038 104 018 NC4 KI
# 9471 0 07 0663      10  GT4 038 104 018 NC4 SU
# 9425 0 04 0377      048 GT4 -1  SU

                                    ********    FINAL RESULT    ********

Q 9475 - 08 0758      10  028 GT4 038 104 018 NC4 SU
```

No. 3: speaker YS

```
****
****
****
YOSI  06

0*******1*******2*******3*******4*******5*******6*******7*******0*******1*******2*******3*******4*******5*******6*******
*CEHAACIPEDGOOCIPEHLAUSOCUFA*EPOEAAI*C/X*FA*   <------ first candidate
PSUSEUSUPXXNAASUHI EEIHUSIVUPGFEAEEXPHIUPHOP   <------ second candidate
92612135091011330003343130100207390410O0169   <------ degree of confidence
96028460600266292242006420220816020262420869

WORD NO.    GG

WORD NO.    KS
SCORE        92  94  94  90  90  89  89
HEAD        003 003 000 000 000 020 020
TAIL        006 007 010 011 012 031 032

WORD NO.    KS  KS
SCORE        94  89
HEAD        000 020
TAIL        007 031
                                          detection of key words

WORD NO.    SC
SCORE        92  93  91  97  97  96
HEAD        003 003 003 026 026 026
TAIL        006 007 010 030 031 032

WORD NO.    SC  SC
SCORE        97  93
HEAD        026 003
TAIL        030 007
                                          identification of predicates

 ﾄﾞｳｼ        ｱｲｸ ﾜｶﾂ ﾊﾂﾌﾟ ﾜｽﾙ ﾜﾗｳ ﾖﾐｺ ﾛｳﾄ ﾜﾘｺ ﾊｼﾞ ﾄﾒﾙ ﾏﾈｷ ﾊｲﾂ ｲﾚｺ ﾈﾚ ｼﾒｾ ｶﾈｺ ｳﾁｽ ﾊｽﾞ ｶｹﾙ ﾄﾘｸ ﾒﾙ
WORD NO.    VA  VB  VC  VD  VE  VF  VG  VH  VI  VJ  VK  VL  VM  VN  VO  VP  VQ  VR  VS  VT  VU
SCORE        94  95  87  91  90  00  00  00  00  00  00  92  81  88  00  00  00  00  86  00  00
HEAD        041 030 026 034 045 053 053 053 053 053 053 041 046 050 053 053 053 053 044 053 053
TAIL        052 052 052 052 052 053 053 053 053 053 053 052 052 052 053 053 053 053 052 053 053
WORD NO.    VA  VB- VC- VD  VE                                                      VS
SCORE        91  87  86  81  76  00  00  00  00  00  00  83  01  26  00  00  00  00  77  00  00
HEAD        044 034 041 036 047 053 053 053 053 053 053 043 050 052 053 053 053 053 047 053 053
TAIL        052 052 052 052 052 053 053 053 053 053 053 052 052 053 053 053 053 053 052 053 053

                                 --------          01 ROUND         --------
 L 9400 4 01 0344  ｱｲﾃｲｶ           VA

 WORD NO.    GP* AD* DA* KS   <------------ predicted words
 SCORE        86  00  00  95
 HEAD        001 000 000 000
 TAIL        023 000 000 007

                                 --------          02 ROUND         --------
 L 9450 6 02 0149  ｱｲﾃｲｶ           VA  KS <-- partial sentence

 WORD NO.    KK1 KC1 KH1 KO1 KZ1 KG1
 SCORE        86  91  91  88  85  94
 HEAD        012 010 010 012 007 007
 TAIL        015 015 015 017 012 015

                                 --------          03 ROUND         --------
 L 9433 2 03 0243  ｱｲﾃｲｶ           VA  KS  KG1

 WORD NO.    JA4 JA4
 SCORE        95  96
 HEAD        015 015
 TAIL        015 015
 K 9233 0 03 0277  ｱｲﾃｲｶ           VA  KS  KK1
 L 9333 2 03 0250  ｱｲﾃｲｶ           VA  KS  KC1

 WORD NO.    JA4 JA4
 SCORE        95  96
 HEAD        015 015
 TAIL        015 015
 K 9333 0 03 0250  ｱｲﾃｲｶ           VA  KS  KH1
 K 9133 0 03 0274  ｱｲﾃｲｶ           VA  KS  KZ1

                                 --------          04 ROUND         --------
 ¥ 0000 1 04 0379  ｱｲﾃｲｶ           VA  KS  KG1 JA4

 WORD NO.    W5
 SCORE        89
 HEAD        015
 TAIL        016
 K 9233 0 03 0277  ｱｲﾃｲｶ           VA  KS  KK1
 ¥ 0000 1 04 0376  ｱｲﾃｲｶ           VA  KS  KC1 JA4

 WORD NO.    W5
 SCORE        89
 HEAD        015
 TAIL        016
 K 9333 0 03 0250  ｱｲﾃｲｶ           VA  KS  KH1
 K 9133 0 03 0274  ｱｲﾃｲｶ           VA  KS  KZ1
 K 9325 0 04 0373  ｱｲﾃｲｶ           VA  KS  KG1 JN4
 K 9250 0 04 0370  ｱｲﾃｲｶ           VA  KS  KC1 JN4

                                 --------          05 ROUND         --------
 ¥ 0000 0 05 0468  ｱｲﾃｲｶ           VA  KS  KG1 JA4 W5
 K 9233 0 03 0277  ｱｲﾃｲｶ           VA  KS  KK1
 K 9300 0 05 0465  ｱｲﾃｲｶ           VA  KS  KC1 JA4 W5
 L 9333 2 03 0250  ｱｲﾃｲｶ           VA  KS  KH1

 WORD NO.    JA4 JA4
 SCORE        95  96
 HEAD        015 015
 TAIL        015 015
 K 9133 0 03 0274  ｱｲﾃｲｶ           VA  KS  KZ1
 K 9325 0 04 0373  ｱｲﾃｲｶ           VA  KS  KG1 JN4
 K 9250 0 04 0370  ｱｲﾃｲｶ           VA  KS  KC1 JN4
```

```
                                                --------          07 ROUND          --------
K 9233 0 03 0277  ｱｲｻｲﾙｶ          VA- KS  KG1
K 9000 0 04 0380  ﾊｲﾂｻｲﾙｶ         VL  KS  KG9 JN4
K 9033 0 03 0271  ｱｲｻｲﾙｶ          VA- KS  KK1
K 9200 0 03 0276  ｱｲｻｲﾙｶ          VA- KS  KH1
K 9033 0 03 0271  ｱｲｻｲﾙｶ          VA- KS  KO1
# 0000 0 06 0559  ﾊｲﾂｻｲﾙｶ         VL  KS  KC9 JN4 DG8 JA4
K 8925 0 04 0357  ﾊｲﾂｻｲﾙｶ         VL  KS  KH9 JN4
K 8975 0 04 0359  ﾊｲﾂｻｲﾙｶ         VL  KS  KO9 JN4
L 9450 6 02 0149  ｷﾚ              VN  KS

WORD NO.     KK9 KC9 KH9 KO9 KZ9 KG9
SCORE         82  95  84  87  84  86
HEAD         013 010 010 012 011 011
TAIL         022 012 014 022 012 014

                                                --------          08 ROUND          --------
K 9233 0 03 0277  ｱｲｻｲﾙｶ          VA- KS  KG1
K 9000 0 04 0380  ﾊｲﾂｻｲﾙｶ         VL  KS  KG9 JN4
K 9033 0 03 0271  ｱｲｻｲﾙｶ          VA- KS  KK1
L 9300 1 01 0353  ﾊｼﾞﾂｻｲﾙｶ        VC-

WORD NO.     KS
SCORE         95
HEAD         002
TAIL         007
K 9200 0 03 0276  ｱｲｻｲﾙｶ          VA- KS  KH1
K 9033 0 03 0271  ｱｲｻｲﾙｶ          VA- KS  KO1
K 9000 0 04 0380  ｷﾚ              VN  KS  KG9 JN4
L 9475 1 04 0379  ｷﾚ              VN  KS  KC9 JN4

WORD NO.     DG8
SCORE         89
HEAD         019
TAIL         022
K 8975 0 04 0359  ﾊｲﾂｻｲﾙｶ         VL  KS  KO9 JN4
K 8975 0 04 0359  ｷﾚ              VN  KS  KO9 JN4

                                                --------          09 ROUND          --------
K 9233 0 03 0277  ｱｲｻｲﾙｶ          VA- KS  KG1
K 9000 0 04 0380  ﾊｲﾂｻｲﾙｶ         VL  KS  KG9 JN4
K 9033 0 03 0271  ｱｲｻｲﾙｶ          VA- KS  KK1
L 9400 6 02 0188  ﾊｼﾞﾂｻｲﾙｶ        VC- KS

WORD NO.     KK9 KC9 KH9 KO9 KZ9 KG9
SCORE         85  98  90  87  87  92
HEAD         010 013 010 012 011 011
TAIL         015 015 015 015 015 015
K 9200 0 03 0276  ｱｲｻｲﾙｶ          VA- KS  KH1
K 9033 0 03 0271  ｱｲｻｲﾙｶ          VA- KS  KO1
K 9000 0 04 0380  ｷﾚ              VN  KS  KG9 JN4
K 9283 0 06 0557  ｷﾚ              VN  KS  KC9 JN4 DG8 JG4
L 9300 2 01 0393  ﾂﾏｶﾞﾂｻｲﾙｶ       VD

WORD NO.     GS9 KS
SCORE         85  95
HEAD         001 002
TAIL         017 007
K 8975 0 04 0359  ｷﾚ              VN  KS  KO9 JN4

                                                --------          10 ROUND          --------
K 9233 0 03 0277  ｱｲｻｲﾙｶ          VA- KS  KG1
L 9600 1 04 0454  ﾊｼﾞﾂｻｲﾙｶ        VC- KS  KC9 JD4

WORD NO.     Z88
SCORE         94
HEAD         016
TAIL         022
K 9033 0 03 0271  ｱｲｻｲﾙｶ          VA- KS  KK1
K 9275 0 04 0371  ﾊｼﾞﾂｻｲﾙｶ        VC- KS  KG9 JD4
K 9200 0 03 0276  ｱｲｻｲﾙｶ          VA- KS  KH1
K 9033 0 03 0271  ｱｲｻｲﾙｶ          VA- KS  KO1
K 9200 0 04 0368  ﾊｼﾞﾂｻｲﾙｶ        VC- KS  KH9 JD4
K 9283 0 06 0557  ｷﾚ              VN  KS  KC9 JN4 DG8 JG4
L 9400 6 02 0148  ﾂﾏｶﾞﾂｻｲﾙｶ       VD  KS

WORD NO.     KK9 KC9 KH9 KO9 KZ9 KG9
SCORE         85  91  84  86  00  85
HEAD         010 010 010 012 007 011
TAIL         017 014 014 017 007 014
K 9025 0 04 0361  ﾊｼﾞﾂｻｲﾙｶ        VC- KS  KZ9 JD4

                                                --------          11 ROUND          --------
K 9233 0 03 0277  ｱｲｻｲﾙｶ          VA- KS  KG1
L 9533 2 06 0572  ﾊｼﾞﾂｻｲﾙｶ        VC- KS  KC9 JD4 Z88 JA4

WORD NO.     A2  A3
SCORE         97  85
HEAD         024 026
TAIL         030 032
K 9033 0 03 0271  ｱｲｻｲﾙｶ          VA- KS  KK1
K 9275 0 04 0371  ﾊｼﾞﾂｻｲﾙｶ        VC- KS  KG9 JD4
K 9200 0 03 0276  ｱｲｻｲﾙｶ          VA- KS  KH1
K 9033 0 03 0271  ｱｲｻｲﾙｶ          VA- KS  KO1
K 9200 0 04 0368  ﾊｼﾞﾂｻｲﾙｶ        VC- KS  KH9 JD4
# 0000 0 06 0557  ｷﾚ              VN  KS  KC9 JN4 DG8 JG4
K 9250 0 04 0370  ﾂﾏｶﾞﾂｻｲﾙｶ       VD  KS  KC9 JT4
K 9025 0 04 0361  ﾊｼﾞﾂｻｲﾙｶ        VC- KS  KZ9 JD4

                                                --------          12 ROUND          --------
K 9233 0 03 0277  ｱｲｻｲﾙｶ          VA- KS  KG1
# 9328 0 07 0653  ﾊｼﾞﾂｻｲﾙｶ        VC- KS  KC9 JD4 Z88 JA4 A3
K 9033 0 03 0271  ｱｲｻｲﾙｶ          VA- KS  KK1
K 9275 0 04 0371  ﾊｼﾞﾂｻｲﾙｶ        VC- KS  KG9 JD4
K 9200 0 03 0276  ｱｲｻｲﾙｶ          VA- KS  KH1
K 9033 0 03 0271  ｱｲｻｲﾙｶ          VA- KS  KO1
K 9200 0 04 0368  ﾊｼﾞﾂｻｲﾙｶ        VC- KS  KH9 JD4
# 9542 0 07 0668  ﾊｼﾞﾂｻｲﾙｶ        VC  KS  KC9 JD4 Z88 JA4 A2
K 9250 0 04 0370  ﾂﾏｶﾞﾂｻｲﾙｶ       VD  KS  KC9 JT4
K 9025 0 04 0361  ﾊｼﾞﾂｻｲﾙｶ        VC- KS  KZ9 JD4

                                        ********    FINAL RESULT    ********

Q 9542 0 07 0668  ﾊｼﾞﾂｻｲﾙｶ        VC  KS  KC9 JD4 Z88 JA4 A2
```

No. 4: speaker UK



```
                                        --------          5 ROUND          --------
K 8966 0 02 0176   ﾂｶﾞﾜﾃｲﾙ        VD  PPA
¥ 0000 1 04 0376   ｱｲﾃｲｶ          VA  KS  KG1 JA4

WORD NO.     W5
SCORE        9.
HEAD         022
TAIL         022
K 9100 0 03 0273   ｱｲﾃｲｶ          VA  KS  KH1
K 9133 0 05 0451   ﾂｶﾞﾜﾃｲﾙ        VD  KS  KG9 JT4 TVA
K 9100 0 02 0176   ﾂｶﾞﾜﾃｲﾙ        VD  CTA
K 9266 0 02 0101   ﾂｶﾞﾜﾃｲﾙ        VD  PRA
K 9025 0 04 0361   ﾂｶﾞﾜﾃｲﾙ        VD  KS  KH9 JT4
K 9100 0 03 0273   ｱｲﾃｲｶ          VA  KS  KZ1
K 9025 0 04 0361   ﾂｶﾞﾜﾃｲﾙ        VD  KS  KZ9 JT4
L 9325 6 04 0373   ｱｲﾃｲｶ          VA  KS  KG1 JA4

WORD NO.     CTA VDA PRA TAA GDA FSA
SCORE         00  94  00  65  65  65
HEAD         016 017 016 017 017 017
TAIL         016 027 016 030 030 030

                                        --------          6 ROUND          --------
K 9250 0 05 0455   ｱｲﾃｲｶ          VA  KS  KG1 JA4 TAA
# 0000 0 05 0456   ｱｲﾃｲｶ          VA  KS  KG1 JA4 W5
K 9100 0 03 0273   ｱｲﾃｲｶ          VA  KS  KH1
K 9133 0 05 0451   ﾂｶﾞﾜﾃｲﾙ        VD  KS  KG9 JT4 TAA
K 9100 0 02 0175   ﾂｶﾞﾜﾃｲﾙ        VD  CTA
K 9266 0 02 0101   ﾂｶﾞﾜﾃｲﾙ        VD  PRA
K 9025 0 04 0361   ﾂｶﾞﾜﾃｲﾙ        VD  KS  KH9 JT4
K 9100 0 03 0273   ｱｲﾃｲｶ          VA  KS  KZ1
K 9025 0 04 0361   ﾂｶﾞﾜﾃｲﾙ        VD  KS  KZ9 JT4
L 9400 1 05 0467   ｱｲﾃｲｶ          VA  KS  KG1 JN4 MDA

WORD NO.     SC
SCORE        97
HEAD         035
TAIL         035

                                        --------          7 ROUND          --------
K 9250 0 05 0456   ｱｲﾃｲｶ          VA  KS  KG1 JN4 TAA
K 9100 0 03 0273   ｱｲﾃｲｶ          VA  KS  KH1
K 9133 0 05 0451   ﾂｶﾞﾜﾃｲﾙ        VD  KS  KG9 JT4 TAA
K 9100 0 02 0176   ﾂｶﾞﾜﾃｲﾙ        VD  CTA
L 9266 1 02 0101   ﾂｶﾞﾜﾃｲﾙ        VD  PRA

WORD NO.     SC
SCORE        94
HEAD         035
TAIL         035
K 9025 0 04 0361   ﾂｶﾞﾜﾃｲﾙ        VD  KS  KH9 JT4
K 9100 0 03 0273   ｱｲﾃｲｶ          VA  KS  KZ1
K 9025 0 04 0361   ﾂｶﾞﾜﾃｲﾙ        VD  KS  KZ9 JT4
L 9400 = 06 0566   ｱｲﾃｲｶ          VA  KS  KG1 JN4 MDA SC

WORD NO.     JN4 J06 018 028 036 048 058 068 078 088 098
SCORE         95  87  00  87  87  89  88  89  90  00  00
HEAD         034 034 033 034 033 034 034 034 034 033 033
TAIL         035 040 033 040 041 041 041 041 040 033 033

                                        --------          8 ROUND          --------
K 9250 0 05 0455   ｱｲﾃｲｶ          VA  KS  KG1 JN4 TAA
L 9300 1 05 0744   ｱｲﾃｲｶ          VA  KS  KG1 JN4 MDA SC  076 HN4

WORD NO.     JN4
SCORE        95
HEAD         041
TAIL         041
K 9250 0 06 0740   ｱｲﾃｲｶ          VA  KS  KG1 JN4 MDA SC  05  HN4
K 9133 0 05 0451   ﾂｶﾞﾜﾃｲﾙ        VD  KS  KG9 JT4 TAA
K 9100 0 02 0176   ﾂｶﾞﾜﾃｲﾙ        VD  CTA
K 9166 0 04 0275   ﾂｶﾞﾜﾃｲﾙ        VD  PRA SC
K 9250 0 06 0740   ｱｲﾃｲｶ          VA  KS  KG1 JN4 MDA SC  064 HN4
K 9100 0 03 0273   ｱｲﾃｲｶ          VA  KS  KZ1
¥ 9342 2 07 0659   ｱｲﾃｲｶ          VA  KS  KG1 JN4 MDA SC  JN4

WORD NO.     W2  W3
SCORE        90  96
HEAD         035 036
TAIL         035 041
K 9275 0 08 0742   ｱｲﾃｲｶ          VA  KS  KG1 JN4 MDA SC  064 HN4

                                        --------          9 ROUND          --------
K 9250 0 05 0458   ｱｲﾃｲｶ          VA  KS  KG1 JN4 TAA
# 9322 0 09 0839   ｱｲﾃｲｶ          VA  KS  KG1 JN4 MDA SC  076 HN4 JN4
K 9250 0 06 0740   ｱｲﾃｲｶ          VA  KS  KG1 JN4 MDA SC  056 HN4
K 9133 0 05 0451   ﾂｶﾞﾜﾃｲﾙ        VD  KS  KG9 JT4 TAA
# 9437 0 08 0755   ｱｲﾃｲｶ          VA  KS  KG1 JN4 MDA SC  JN4 W3
K 9166 0 03 0275   ﾂｶﾞﾜﾃｲﾙ        VD  PRA SC
K 9250 0 06 0740   ｱｲﾃｲｶ          VA  KS  KG1 JN4 MDA SC  066 HN4
K 9100 0 03 0273   ｱｲﾃｲｶ          VA  KS  KZ1
# 9342 0 07 0659   ｱｲﾃｲｶ          VA  KS  KG1 JN4 MDA SC  JN4
K 9275 0 08 0742   ｱｲﾃｲｶ          VA  KS  KG1 JN4 MDA SC  064 HN4

                                  ********   FINAL RESULT   ********

0 9437 0 08 0755   ｱｲﾃｲｶ          VA  KS  KG1 JN4 MDA SC  JN4 W3
```

No. 5: speaker NG