STUDIA PHONOLOGICA XII (1978)

A Pre-Matching Method for a Real Time Spoken Word Recognition System and a Learning Procedure of Speaker Differences

Sei-ichi NAKAGAWA and Toshiyuki SAKAI

SUMMARY

If we enlarge the vocabulary size of the word recognition system to about several hundreds, we are afraid that the recognition time becomes not only very long by increasing an amount of processing but also the correct rate of recognition decreases. To cope with these weak points, we adopted the method which reduced candidate words in the vocabulary by means of pre-matching using both local and global features of a spoken word. That is, to eliminate the most unlike group of candidates using the measurements of both features from the vocabulary list was tried to reduce the recognition time, and this operation also eliminated the misleading candidates to make increase the correct rate of recognition. Furthermore, to add the measurement to the final judgement made increase the correct rate. Moreover, in order to absorb the influence of speaker differences, we added the capability of learning to the system.

In an experiment on name recognition using 100 Japanese-city names, the system recognized the names correctly at the rate of 83% for unspecific speakers and 93% after learning, using a mini-computer in real time. The number of candidate words was reduced to one tenth by pre-matching.

I. INTRODUCTION

In automatic word recognition, if the whole word input pattern is regarded as a point in the pattern space, the recognizer can avoid the problem of coarticulation. In that case, it can also use the linguistic information consulting with the word dictionary. 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 focussed on the following goals:

(1) enlargement of the vocabulary size.

Sei-ichi NAKAGAWA (中川聖一) : Assistant, Department of Information Science, Kyoto University.
Toshiyuki SAKAI (坂井利之) : Professor, Department of Information Science, Kyoto University.

(2) reduction of the amount of computation and storage.
(3) development for a general word recognition scheme using phoneme recognition.
(4) normalization or learning of speaker differences.
(5) extension to recognition of connected words.

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, we believe that a matching method using DP(dynamic programming) is one of the best algorithms suited for automatic word recognition. We 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.

In this paper, we describe our recognition method on the subjects of (1) through (4) mentioned above.

So far, we have experimented on the recognition of spoken digits in isolation by a real-time spoken word recognition system on a mini-computer[1),2)]. If we enlarge the vocabulary size of the system to a few hundred words, we are afraid that the recognition time becomes not only very long depending upon the processing value but also the correct rate of recognition decreases. Although some techniques have been tried to reduce computation time[3)], we propose a new pre-matching method for shortening of recognition time.

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

1. *Normalization approach*[4),5),6)]..... 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.

2. *Learning approach*[1),7),8)]..... 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.

Although our system employes both learning procedures[1)], in this paper, we describe a non-supervised learning procedure.

II. SYSTEM OVERVIEW

We reconstructed a subpart (Phoneme Recognizer and Word Identifier) of the LITHAN speech understanding system[9)~12)] to a real-time spoken word recognition system on a mini-computer[1),2)]. Fig. 1 shows a block diagram of this system.

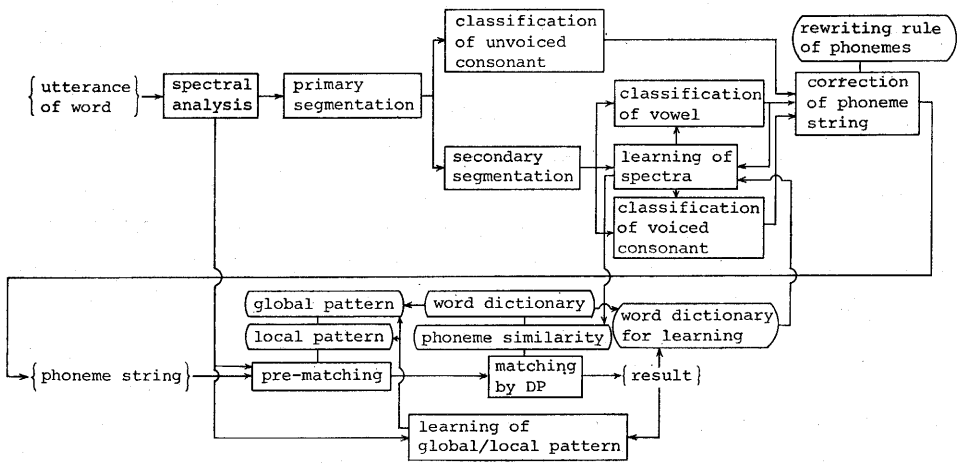


Fig. 1 Configuration of a real time spoken word recognition system.

Speech signals are 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; 40 Hz), 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 facter $2^{1/4}$, from 210 Hz to 5660 Hz.

The Phoneme Recognizer divides input speech into segments of utterances which are hypothesized to consist of a continuum of a single phoneme; it then assigns one phoneme category to each segment.

Primary segmentation is performed on such analyzed speech, now represented by a sequence of short time spectra (we 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. If a part is irregular, it is smoothed by using rewriting or phonological rules. The output of this process 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. From segments classified into voiced groups,

a portion of voiced consonants is detected as one of the following: 1) long transient; 2) having weak energy with concave speech level; and the undetected portions are regarded as vowels (secondary segmentation).

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). 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 are recognized by Euclidean distance. These phoneme recognition procedures can all be achieved for each frame within a sampling interval 10ms.

By rewriting rules, a recognized phoneme sequence in the voiced parts is smoothed and merged. Then, output of the phoneme classification process makes 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 (Word Recognizer).

The first three constituents of the j-th segments in a sequence will be denoted by J, l and $\mathrm{p}(0 \leqq \mathrm{p} \leqq 1.0)$, 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 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 equation:

$$\mathrm{S}(I,k,c\,;J,l,\mathrm{p}) = \max \left\{ \begin{array}{l} \mathrm{S}(I,J) \\ \mathrm{c}\times\mathrm{S}(k,J) \\ \mathrm{p}\times\mathrm{S}(I,J)+(1-\mathrm{p})\times\mathrm{S}(I,l) \\ \mathrm{p}\times\mathrm{c}\times\mathrm{S}(k,J)+(1-\mathrm{p})\times\mathrm{c}\times\mathrm{S}(k,l) \end{array} \right\}$$

where S(*,*) is the similarity (measure) between two phonemes given in *a priori*. By using this measure, we can evaluate the matching score between an element string of a word and a recognized string. We make the following restrictions except special cases with respect to the matching. These could be regarded as reasonable restrictions, judging from the performance of the Phoneme Recognizer.

(1) A vowel and the syllabic nasal in the word dictionary can be associ-

ated with three or less segments in a recognized phoneme string.

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

(3) Three or more successive elements cannot be associated with one segment.

If an element in a word associates with a segment of a recognized string, the similarity is directly celculated by the above equation. 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 a recognized phoneme string and an element string; we then say that the similarity with the highest score is the likelihood for that word. A recognized string is recognized as the word which gained the highest likelihood. This calculation is carried out efficiently by the use of dynamic programming (DP-matching). Fig. 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 vertical axis denotes a string of elements of a lexical entry.

The process for shortening of recognition time lies between Phoneme Recognizer and Word Recognizer. Before the DP-matching, an uttered word is checked about the measurement of both local and global features of a spoken word to eliminate the most unlike group of candidates from the vocabulary list (we call this pre-matching). Besides benefits of the shortening of recognition time, the correct rate of recognition would be improved by adding this matching result to DP-matching result. The spectral patterns at the head and tail positions of spoken words were employed for the local features. Both the number of phonemes, and

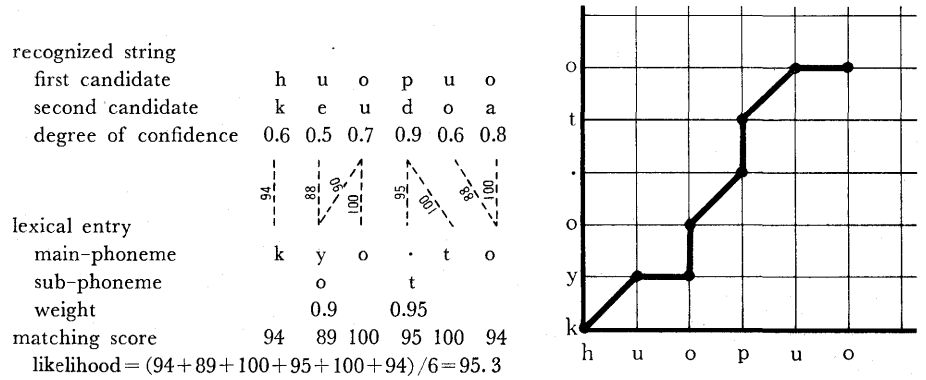


Fig. 2 Graphic representation of an example of word matching and matching score.

presence fo five vowels and voiced and unvoiced consonants in the recognized phoneme string, and the contour of energy envelope were emplyed for the global features.

The spectral patterns of vowels, weighting coefficients and binary patterns for pre-matching are learned for each speaker with respect to his spoken word. And also, the phoneme similarity matrix and spectral patterns of voiced consonants are modified for each speaker by using the learned spectral patterns of vowels.

This system was implemented on a mini-computer (MELCOM-70, cycle time=0.8μs, core memory=24K words). The algorithms were programmed with an assembly language. This program occupied about 7.3K words and the work area about 8.7K words. The matching time required between a segment string and an element string in a lexical entry was about 50ms. This system can illustrate a recognition result on a graphic display within 500ms for a 100-word vocabulary.

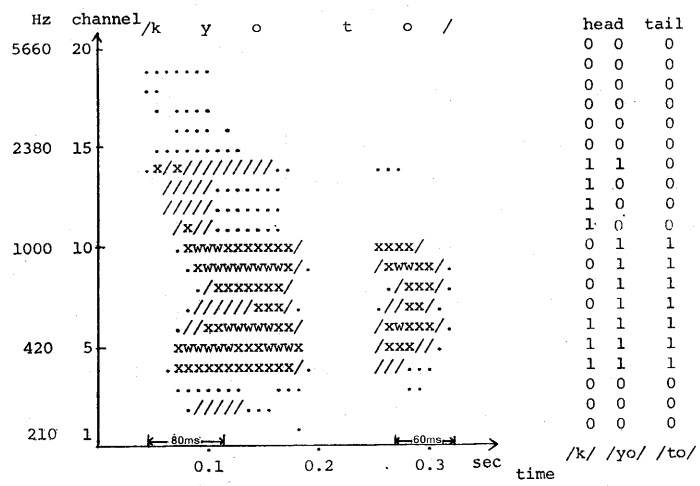
III. PRE-MATCHING PROCEDURE

Pre-matching should be a simple algorithm because this aims to shorten the recognition time. This procedure pre-selects candidate words for an uttered word from the vocabulary and passes only these words to DP-matching procedure. We employed both local and global features of a spoken word for pre-matching. These features have not been used at the Phoneme Recognizer or Word Recognizer (DP-matching), therefore, we can expect that pre-matching also brings the improvement of recognition performance by following two reasons.

(1) Indistinguishable words which are similar to an uttered word on a recognized segment string are rejected.
(2) The correct rate of recognition is improved by adding these new measures of pre-matching to DP-matching score.

III-1 LOCAL PATTERN OF SPOKEN WORD

The spectral binary patterns of head and tail positions of a spoken word are employed for local features. In the head position, eight frames (80ms) are divided into two intervals automatically at the most changing point on the basis of spectral changes. That is, the system computes the amount of spectral change between adjacent frames from third frame to sixth frame. Of course, spectral values of each frame are normalized in order to avoid the variation of speech power. The spectral patterns are averaged over all frames in each divided interval for every channel and then are transformed into a binary pattern at a certain threshold value. Thus, we obtain a spectral binary pattern of 40 bits. In almost cases, the



output value of channel: w>x>/>·>blank

Fig. 3 Local spectral pattern.

front of these corresponds to a consonant of CV (or CYV) syllables and the end part a vowel. If the initial phoneme of a word is a vowel or a fricative, both front and end parts correspond to the phoneme. However, we expect that the end part never contains the following CV syllable, because man requires the duration time of more than 80ms to utter any CV syllable and vowel. Therefore, we can make these reference patterns from all CV syllables (about 100 syllables) spoken in isolation of Japanese.

In the tail position, the spectral patterns of the last six frames (60ms) are averaged and transformed into a binary pattern in the same mannar as the head position. We obtain a spectral binary pattern of 20 bits.

It has the advantage that these positions are exactly detected independent on the speed of an utterance and also are insensitive on contexts. Fig. 3 illustrates an example of these positions and patterns.

III-2 Global Pattern of Spoken Word

(a) Number of phonemes in a spoken word

Whatever the speed of an utterance is, the number of phonemes in a spoken word is constant. Therefore we can derive from the number of recognized phonemes what kind of words was uttered. However we should note that the segmentation of our system is incomplete.

(b) Kind of phonemes in a spoken word

We can guess an uttered word by the kind of phonemes contained in a recognized phoneme string. Although if the order of appearance of phonemes is taken into consideration, the exacter guess would be possible,

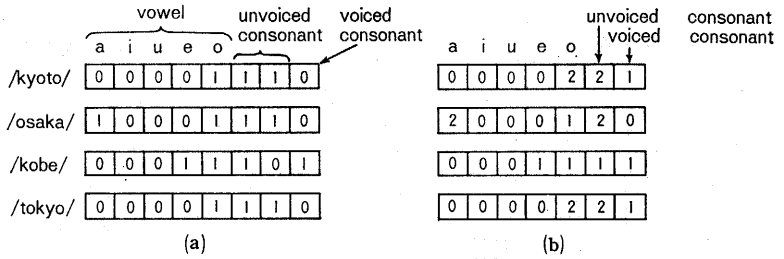


Fig. 4 Phonemic pattern.

the algorithm to guess would become sophisticated by reason of the incompleteness of segmentation and recognition of Phoneme Recognizer. Therefore we transform the recognized phoneme string into the following eight bit pattern (phonemic pattern).

vowel (a, i, u, e, o)..... 1 bit for each vowel.

unvoiced consonant (p, t, k, s, c, h)..... 2 bits.

voiced consonant (m, n, ŋ, b, d, g, r, z)..... 1 bit.

If these phonemes appear as the first candidate phoneme of recognized phoneme string, the corresponding bits are set 1, if not, 0 (if two unvoiced consonants appear→11, one→10, zero→00). The reference pattern is translated from the word dictionary. Examples are shown in Fig. 4(a).

Fig. 4 (b) shows the phonemic pattern which takes the number of phonemes into consideration. For example, if there are two /a/ in a word, the corresponding bit pattern is 0011 and if there are three unvoiced consonants, the bit pattern is 0111. The syllabic nasal and semivowels are regarded as one of voiced consonants for convenience's sake. Thus, this pattern consists of 28 bits.

(c) Contour of energy envelope

The global contour of energy envelope of a spoken word is almost constant among speakers. Where the energy is defined as the root of square sum of 20-channel outputs. The duration time of a word is divided into 14 intervals linearly. The system checks for each interval if the slope of energy envelope is positive or negative and if the value of energy level is larger than the half of the largest value in all frames. For each

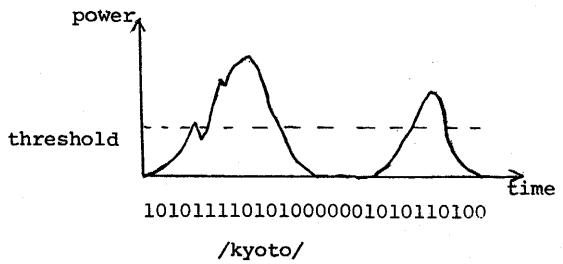


Fig. 5 Envelope pattern of speech power.

interval, two bits are set according to the status mentioned above as follows: positive and larger→11, positive and smaller→10, negative and larger→01, and negative and smaller→00. Thus we obtain a binary pattern of 28 bits. Only the reference pattern of this feature can not generate automatically, that is, we must prepare the reference pattern for every word individually. Fig. 5 illustrates an example of this pattern.

III-3 Matching Procedure

Pre-matching should be performed very quickly, because it aims to reduce the recognition time of a spoken word. Therefore we adopted the simple bit pattern and simple algorithm. As described above, features for pre-matching are represented by the number of phonemes in a spoken word and binary patterns of 96 bits (6 computer-words); such as spectral patterns in the head and tail positions (40 bits and 20 bits), a phonemic pattern of 8 bits and a contour pattern of energy envelope of 28 bits. These binary patterns were compared independently with each reference pattern on the basis of Hamming distance. Let θ be the difference between the number of recognized phonemes and the number of phonemes in the n-th lexical word. Let α, β, γ, and δ be Hamming distance between the binary patterns of a spoken word and the binary reference patterns of the n-th lexical word, respectively. The procedure consists of following 6 steps.

(1) Check of the number of phonemes.
When the number in the n-th lexical word is smaller than 9, if $\theta \geqq 4$, its lexical word is rejected. When the number is greater than 10, if $\theta \geqq 5$, it is rejected.

(2) Check of the spectral binary pattern in the head position.
If $\alpha \geqq 20$, its lexical word is rejected.

(3) Check of the spectral binary pattern in the tail position.
If $\beta \geqq 12$, its lexical word is rejected.

(4) Check of the phonemic pattern.
If $\gamma \geqq 5$, its lexical word is rejected.

(5) Check of the contour pattern of energy envelope.
If $\delta \geqq 12$, its lexical word is rejected.

(6) Check of the total patterns.
If $\alpha+\beta+\gamma+\delta \geqq 36$, its lexical word is rejected.

IV Learning of Speaker Differences

We tried a non-supervised learning method for learning of speaker differences. This method is performed in parallel with recognition stage of a spoken word. Therefore, this has the following advantages.

(1) This method automatically learns the speaker differences without any burden of a speaker or an operator.
(2) Extracted samples for learning are most suitable, because they have the same contexts as recognized words.
(3) This method can follow in the changes of reference patterns of a spcaker depending on time or environments.
(4) This method can be taken other learning methods together.

IV-1 LEARNING OF VOWEL SPECTRUM

Vowel spectra are different from speaker to speaker. Therefore, the system must normalize or learn these speaker differences. Only one reference spectral pattern may be insufficient for each vowel, because vowel spectra are influenced by contexts or nasalization. In the initial step of reference patterns and lexical word, each vowel is represented as only one way for every speakers, because we have following questions:

(1) Should each vowel pattern of any speaker be divided into two classes?
(2) Is the way of division into two classes the same concerning vowel spectra in a context for every speakers?
(3) Is there a systematic way of division for all contexts when we divide the reference patterns for each vowel into two classes in advance?

Therefore, our system generates two reference patterns automatically for each vowel from one reference pattern by using learning samples, although we have a basic question how many classes for each vowel are necessary.

A system must extract correct learning samples to avoid mislearning and it is also desired to extract them automatically as possible in order not to require many loads to a speaker or an operator. Our system adopted the following approaches taking non-supervised learning into consideration.

(1) Each vowel spectrum is divided into two classes on the stage of phoneme recognition.
(2) Only reliable spectra which are decided by the score of result of phoneme recognition are used as learning samples.
(3) Whether the extracted samples by (2) are used for learning or not is dependent on the score of result of word recognition. That is, when the score is enough high and vowels are included in its word lexicon for learning, such only vowels are adopted as vowels for learning.

Here, the lexicon for learning is a vowel list included in its original lexicon, but it does not include devocalized vowels, because the spectra

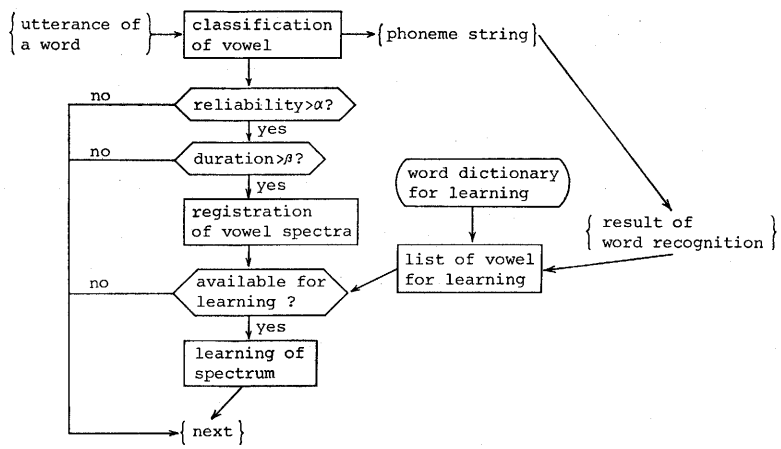


Fig. 6 Learning of vowel spectra.

of devocalized vowels are unstable in all most cases and are scarcely recognized as its vowel. We describe in detail the learning procedure below. This procedure is performed by following three steps as shown in Fig. 6.

step 1: For spectra (or frames) recognized as a vowel in the stage of phoneme recognition, if

(1) the degree of confidence of the vowel is larger than a threshold α, and

(2) the duration of the vowel is larger than a threshold β,

these spectra are registered into a data area for learning.

step 2: When an input utterance is recognized as a word, the score of recognition result should be beyond a threshold γ in order to be learned. Where, let Δ be the score of difference between first best candidate (recognition result) and second best candidate. Then if either first or second condition is satisfied, the following vowels might be learned.

(1) If $\Delta \geqq \delta$ (a threshold), vowels included in the lexicon of first candidate for learning might be learned.

(2) If $\Delta < \delta$ (a threshold), vowels included commonly in both the lexicons of first best candidate and second best candidate for learning might be learned.

step 3: If spectra of the vowels which are selected by step 2 have been registered in the data area for learning, the spectra are used for learning of a class out of 10 classes.

The procedure mentioned above is a non-supervised learning method. If we want to adopt a supervised learning method, the operator should

teach the correct word as a recognition result to the system in order to satisfy unconditionary the condition (1) of step 2.

Now, we will explain a learning method of vowel spectra and coefficients of linear discriminant functions. Let $\boldsymbol{V}_{ik}$ (i$\in$ {/a/, /i/, /u/, /e/, /o/}, k$\in$ {first class, second class for each vowel}) be the learning sampl for vowel i and class k, N_{ik} be the total number of learned samples for them, and $\boldsymbol{V}_{ik}^{N_{ik}}$ be the reference spectrum of N_{ik}-th step. Then,

$$\boldsymbol{V}_{ik}^{N_{ik}+1}=\frac{1}{\alpha+N_{ik}+1}\cdot\{(\alpha+N_{ik})\cdot\boldsymbol{V}_{ik}^{N_{ik}}+\boldsymbol{V}_{ik}\}$$

For a reference coefficient as similar above,

$$\boldsymbol{w}_{ik_1,jk_2}^{N_{ik_1}+jk_2}=\frac{1}{\alpha'+N_{ik_1}+N_{jk_2}}\{\alpha'\boldsymbol{w}_{ik_1,jk_2}^{0}+(N_{ik_1}+N_{jk_2})\cdot(\boldsymbol{V}_{ik_1}^{N_{ik_1}}-\boldsymbol{V}_{jk_2}^{N_{jk_2}})\}$$

where both $\boldsymbol{V}_{ik}^{0}$ and $\boldsymbol{w}_{ik_1,jk_2}^{0}$ are initial patterns, and α and α' are constant.

IV-2 Learning of Local and Global Binary Pattern

Although learning of binary patterns can obtain from the learning of original patterns since the original patterns are transformed into binary patterns, our system directly learns the binary patterns. The procedure is the following as shown in Fig. 7.

step 1: When an input utterance is recognized as a word, the score of recognition result should be beyond a threshold γ and Δ should be beyond a threshold δ (see step 2 of section IV-1) in order to be learned. Then, the input binary pattern is used as a learning sample.

step 2: Each bit of reference binary pattern of the recognized word is renewed by the majority decision of three bits of input binary pattern, temporary binary pattern and reference pattern.

step 3: The temporary binary pattern is replaced by the input binary pattern.

Where, the intial temporary binary pattern is the copy of the initial

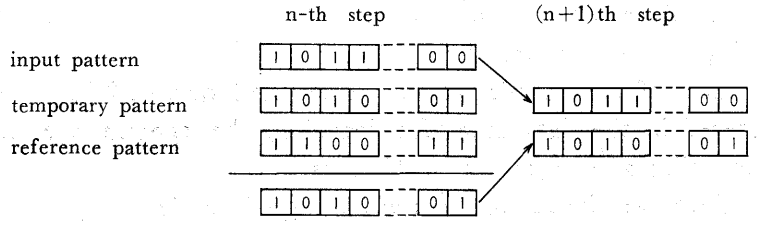


Fig. 7 Learning of binary pattern.

reference binary pattern. Roughly speaking, the new reference pattern is the average of the last three learning samples.

V. Difference of Local and Global Features Between Words

We investigated to which extent the local and global features are effective for pre-selection. Eight male adults uttered all Japanese syllables and 100 Japanese-city names. Table 1 shows the vocabulary of city

Table 1 List of 100 Japanese-city names.

sa-poro	aomori	akita	morioka	sendai
yamagata	hukusima	mito	ucunomiya	maebasi
urawa	ciba	tokyo	yokohama	nigata
toyama	kanazawa	hukui	kohu	nagano
gihu	sizuoka	nagoya	cu	ocu
kyoto	osaka	kobe	nara	wakayama
to-tori	macue	okayama	hirosima	yamaguci
tokusima	takamacu	macuyama	koci	hukuoka
saga	nagasaki	kumamoto	oita	miyazaki
kagosima	naha	asahikawa	hakodate	kusiro
otaru	obihiro	abasiri	kawasaki	macumoto
hamamacu	sakai	himezi	simonoseki	kitakyusyu
wa-kanai	nemuro	muroran	tomakomai	hirosaki
kamaisi	iwaki	koriyama	aizuwakamacu	takasaki
omiya	kawaguci	hunabasi	haciozi	yokosuka
odawara	nagaoka	ogaki	simizu	atami
toyohasi	yo-kaici	hikone	maizuru	higasiosaka
toyonaka	amagasaki	nisinomiya	akasi	yonago
izumo	kurasiki	kure	hukuyama	ube
imabari	nihama	kurume	sasebo	be-pu

Table 2 Confusable words.

tokyo/kyoto	nagasaki/kawasaki/amagasaki/takasaki	urawa/nara/naha
wakayama/okayama	hukusima/tokusima	sizuoka/hukuoka
ucunomiya/nisinomiya	kobe/kure/kurume/ube	yamaguci/kamaisi/kawaguci
kohu/ocu/koci	nigata/nihama	takamacu/hamamacu nagano/yonago

Table 3 Distribution of difference of number of phonemes between words.

difference	ratio	difference	ratio
0	17.7%	5	4.0 (96.8)
1	25.9 (43.6)	6	2.1 (98.9)
2	21.7 (65.3)	7	0.7 (99.6)
3	16.9 (82.2)	8	0.3 (99.9)
4	10.6 (92.8)	9	0.1 (100.0)

names. This vocabulary includes many confusable words as shown in Table 2. These utterances were used for the construction of spectral binary patterns and envelope pattern of speech power, respectively. Tables

Table 4 Distribution of Hamming distance between binary patterns of words.

a : phonemic pattern (a), b : phonemic pattern (b)
c : spectral binary pattern at head position
d : spectral binary pattern at tail position
e : envelope pattern of speech power

distance	a	b	c	d	e
0	4.1%	1.3%	4.6%	9.0%	1.0%
1	13.4 (17.5)	1.0 (2.3)	0.3 (4.9)	6.0 (15.0)	0.1 (1.1)
2	24.1 (41.6)	3.9 (6.2)	0.2 (5.1)	2.6 (17.6)	0.1 (1.2)
3	25.8 (67.4)	8.6 (14.8)	0.1 (5.2)	5.8 (23.4)	0.6 (1.8)
4	19.3 (86.7)	14.2 (29.0)	0.7 (5.9)	2.7 (26.1)	1.1 (2.9)
5	9.7 (96.4)	16.5 (45.5)	1.4 (7.3)	2.4 (28.5)	2.4 (5.3)
6	2.9 (99.3)	18.3 (63.8)	1.2 (8.5)	5.1 (33.6)	3.2 (8.5)
7	0.6 (99.9)	14.1 (77.9)	1.7 (10.2)	6.6 (40.2)	4.9 (13.4)
8	0.1 (100.0)	11.5 (89.4)	1.6 (11.8)	7.0 (47.2)	7.7 (21.1)
9		5.8 (95.2)	2.7 (14.5)	8.1 (55.3)	9.4 (30.5)
10		3.3 (98.5)	2.5 (17.0)	8.0 (63.3)	10.7 (41.2)
11		1.0 (99.5)	2.6 (19.6)	3.5 (66.8)	11.7 (52.9)
12		0.4 (99.9)	2.3 (21.9)	4.9 (71.7)	11.4 (64.3)
13		0.1 (100.0)	2.3 (24.2)	10.5 (82.2)	10.8 (75.1)
14			6.0 (30.2)	11.5 (93.7)	8.7 (83.8)
15			4.5 (34.7)	5.4 (99.1)	6.4 (90.2)
16			1.4 (36.1)	0.9 (100.0)	4.3 (94.5)
17			6.4 (42.5)		2.8 (97.3)
18			1.9 (44.4)		1.6 (98.9)
19			6.0 (50.4)		0.7 (99.6)
20			7.3 (57.7)		0.3 (99.9)
21			6.6 (64.3)		0.1 (100.0)
22			9.1 (73.4)		
23			5.6 (79.0)		
24			2.8 (81.8)		
25			4.2 (86.0)		
26			2.4 (88.4)		
27			1.7 (90.1)		
28			3.2 (93.3)		
29			1.6 (94.9)		
30			1.6 (96.5)		
31			0.8 (97.3)		
32			0.6 (97.9)		
33			0.4 (98.3)		
34			0.4 (98.7)		
35			0.8 (99.5)		
36			0.5 (100.0)		

3 and 4 show the distributions of the difference of binary patterns between words on all pair of 100 Japanese-city names. The total number of pairs is $100C_2+100$(themselves)$=5050$. The value in the parentheses denotes the cummulative distribution. From Table 3, we can guess, if the segmentation of our system is performed completely, the system can reject 82 words on the average from this vocabulary by using this feature.

VI. Experiment

We applied our system to the speech recognition of 100-city names in Japan. The lexical entry consists of 3.5 syllables on the average. The 100 Japanese-city names were uttered five times in isolation by each of five male adults. We say these in order set 1, set 2, ..., set 5 for convenience' sake. The reference spectral patterns of phonemes and similarity matrix were calculated from VCV contexts which were included in 2450 words spoken by other 10 male adults.

V-1 Effectiveness of Pre-Matching

In order to investigate the effectiveness of pre-matching, we made experiments to recognize the spoken words of set 1. Table 5 shows the results.

The column (a) shows the correct rate of recognition in the case without pre-matching, that is, with using only DP-matching.

The column (b) shows the correct rate in the case without DP-matching, that is, with using only pre-matching.

Table 5 Recognition results of 100 Japanese-city names for set 1 (without learning).
a: without pre-matching (only DP matching).
b: without DP-matching (only pre-matching).
c: with pre-selection.
d: with pre-selection and pre-matching score.
e: within the top 2 choices.
f: average number of pre-selected words from 100 words.
g: rejected number of input words by pre-selection in 100 utterances.

speaker	a	b	c	d	e	f	g
ST	55.0%	70.0%	71.0%	80.0%	86.0%	9.23	8
FK	63.0	71.0	73.0	76.0	83.0	10.20	10
HR	88.0	86.0	90.0	92.0	94.0	9.96	4
MK	70.0	76.0	78.0	86.0	92.0	8.82	6
MG	76.0	80.0	83.0	84.0	90.0	10.46	5
average	70.4	76.6	79.0	83.6	89.0	9.73	6.6

The column (c) shows the correct rate in the case with the pre-selection by pre-matching.
The column (d) shows the correct rate in the case with the pre-selection and with the final decision by adding (the pre-matching score = (96-Hamming distance)) × 0.3 to DP-matching score.
The column (e) shows the correct rate within the top 2 choices in the case (d).
The column (f) shows the average number of pre-selected words from 100 words.
The column (g) shows the rejected number of input words by the pre-selection in 100 utterances, that is, these input words were misrecognized by the stage of pre-matching.

From these results, we can conclude that the pre-selection by pre-matching reduced the candidate words from 100 words to about 10 words. In the other words, the recognition time was improved a factor of 10, since the times of phoneme recognition and pre-matching can be ignored in comparison with DP-matching.

And also, the correct rate was improved from about 70% to about 83% by taking the pre-matching into consideration. However, a few correct word (6.6%) was rejected by the pre-selection. Therefore, we cannot expect that the correct rate is beyond 94%, even if Phoneme Recognizer and Word Recognizer are improved. From this fact, (c) and (d), we find that more than half of the misrecognized input words were recognized as the second best in the stage of DP-matching.

Table 6 shows the performance of pre-matching at each step. The spectral binary pattern at the head position of a spoken word and the envelope pattern of speech power are effective in particular.

Table 7 shows the performance of local and global features. These features have the almost same performance, that is, about a factor of

Table 6 Performance of pre-matching.

feature	number of pass	ratio of pass	ideal ratio (Table 3, 4)	threshold
without pre-matching	100			
number of phonemes	74.0	74%	82%	4(5)
spectral pattern at head position	40.2	60	50	20
spectral pattern at tail position	26.8	67	67	12
phonemic pattern	21.4	80	87	5
envelope pattern of speech power	12.4	58	53	12
all features	9.7	78		36

Table 7 Performance of local and global features.

(a) local feature

speaker	recognition rate	number of pass	number of reject
ST	71%	37.8	5
FK	67	37.3	5
HR	88	35.1	0
MK	78	33.6	3
MG	82	38.9	3
average	77.2	36.5	3.2

(b) global feature

speaker	recognition rate	number of pass	number of reject
ST	64%	29.3	4
FK	64	32.3	5
HR	88	30.9	4
MK	79	31.9	3
MG	83	32.0	2
average	75.6	31.3	3.6

3～4, respectively.

Spoken Digit Recognition

Next, we experimented on the recognition of spoken 10 digits. The results are shown in Table 8. Five male adults uttered 10 digits, each of which was uttered two times by each speaker. Because /u/ in 'roku' [=6] was often devocalized, several words of 'roku' was rejected. Therefore, we made a new spectral binary pattern of the tail position of 'roku'. The symbol * in Table 8 denotes the result of this experiment. In this case, there was no rejected word. Comparing with city names, the effectiveness of pre-selection on 10 digits decreased. This is caused by the following reason. The longer the average length of words in a vocabulary becomes, the better the effectiveness of pre-selection becomes.

Table 8 Recognition results of 10 digits (without learning).

a : without pre-matching (only DP matching).
b : without DP-matching (only pre-matching).
c : with pre-selection.
d : with pre-selection and pre-matching score.
f : average number of pre-selected words from 10 words.
g : rejected number of input words by pre-selection in 10 utterances.
* in d and g denotes results after adjustment of spectral binary pattern of tail position of 'roku'.

speaker	a	b	c	d	d*	f	g	g*
NG	100%	100%	100%	100%	100%	2.8	0	0
UK	95	85	95	100	100	2.8	0	0
AR	90	95	90	95	100	2.7	1	0
NK	85	90	85	85	90	3.2	1	0
TR	85	85	85	95	100	2.9	1	0
average	91	91	91	95	98	2.9	0.6	0

V-2 EFFECTIVENESS OF LEARNING

Next, we experimented on the recognition of the spoken words of sets 4 and 5 in order to investigate the effectiveness of the learning procedure of speaker differences. The sets 1, 2 and 3 were used for learning samples. The spectral patterns for phoneme recognition and binary patterns for pre-matching were learned by the non-supervised learning. The learning was performed by three steps. The first step used only set 1 for learning samples. The second step used set 2 in succession, that is, the learning was performed by using sets 1 and 2. The third step used set 3 in succession.

The experimental results are shown in Table 9. The contents of columns d, f and g are the same as those of Table 5. About 60% of input words were used for the non-supervised learning. Note that the binary patterns are not modified by the first step, because more than two samples for each word is necessary to modify them. Therefore, the contents of (f) and (g) are almost the same as those in the case without learning, since only the recognized phoneme number and kind of recognized phonemes slightly change. The correct rate was improved by the learning procedure from 85.0% to 93.3%. Although the rejected number decreased from 5.8 to 1.8, the number of pre-selected words increased from 9.8 to 14.4 in the stage of the pre-matching procedure. This fact shows, we guess, that the difference between binary patterns of a word spoken by two speakers is larger than the difference between binary patterns of two different words spoken by a speaker.

Table 9 Reconigtion results of 100 Japanese-city names for sets 4 and 5 (with learning).
d: with pre-selection and pre-matching score.
f: average number of pre-selected words from 100 words.
g: rejected number of input words by pre-selection in 100 utterances.

speaker	set	without learning			learning by set 1			learning by sets 1 and 2			learning by sets 1, 2 and 3		
		d	f	g	d	f	g	d	f	g	d	f	g
MK	4	86.0%	8.6	4	88.0%	9.4	3	95.0%	12.1	1	94.0%	12.7	2
	5	85.0	8.8	5	85.0	9.6	5	89.0	11.2	7	92.0	12.5	2
MG	4	88.0	10.9	5	89.0	10.8	4	94.0	15.5	2	95.0	16.5	1
	5	81.0	10.6	6	88.0	10.8	4	92.0	15.4	1	92.0	15.7	2
average		85.0	9.8	5.0	87.5	10.2	4.0	92.5	13.4	2.8	93.3	14.4	1.8

VI. CONCLUSION

To eliminate the most unlike group of candidates using the measurements of both local and global features of a word from vocabulary list reduced the recognition time, and this operation also made increase the correct rate. On experiment of 100 Japanese-city names recognition, the system recognized them with the correct rate of 83% for unspecific speakers and 93% after learning the speaker differences. The recognition time of a spoken word was about 0.3～0.6sec on a mini-computer (MELCOM-70, cycle time=0.8μs, core memory=24 K words). The program size was about 7.3K steps (7.3K words). The work area was about 8.7K words including 600 words for similarity matrix, 1200 words for reference patterns on phoneme recognition, 800 words for the word dictionary of 100 city names and 600 words for reference patterns on pre-matching. The system can treat a vocabulary size of several hundred words and recognize them within one second.

ACKNOWLEDGEMENT

The authors wish to thank Mr. H. Shirakata and M. Utsumi for their cooperation.

REFERENCES

1) S. Nakagawa and T. Sakai: A Real Time Spoken Word Recognition System with Various Learning Capabilities of the Speaker Differences, Jour. of IECEJ Vol. 61, No. 6 (1978) (in Japanese).
2) T. Sakai and S. Nakagawa: On-Line, Real-Time Spoken Words Recognition System with Learning Capability of the Speaker Differences, Studia Phonologica, X (1976).
3) D. R. Reddy: Speech Recognition by Machine: A Review, Proceedings of the IEEE, Vol. 64, No. 4 (1976).
4) L. J. Gerstman: Classification of Self-Normalized Vowels, IEEE Trans. Vol. AU-16, No. 1 (1968).
5) H. Fujisaki et al.: Analysis, Normalization and Recognition of Sustained Japanese Vowels, Jour. of ASJ, Vol. 26, No. 3 (1970).
6) H. Wakita: Normalization of Vowels by Vocal-Tract Length and Its Application to Vowel Identification, IEEE Trans. Vol. ASSP-25, No. 2 (1977).
7) S. Saito and M. Kohda: Spoken Word Recognition Using the Restricted Number of Learning Samples, Conference Record of ICASSP (1976).
8) B. T. Lowerre: Dynamic Speaker Adaption in the Harpy Speech Recognition System, Conference Record of ICASSP (1977).
9) T. Sakai and S. Nakagawa: A Classification Method of Spoken Words in Continuous Speech for Many Speakers, Information Processing in Japan, Vol. 17 (1977).
10) S. Nakagawa and T. Sakai: A word Recognition Method from a Classified Phoneme String in the LITHAN Speech Understanding System, Conference Record of ICASSP (1978).

11) T. Sakai and S. Nakagawa: A Speech Understanding System of Simple Japanese Sentences in a Task Domain, Jour. of IECEJ, Vol. E-60, No. 1 (1977).

12) T. Sakai and S. Nakagawa: Speech Understanding System LITHAN and Some Applications, Proceedings of 3rd IJCPR, Coronado (1976).

(Aug. 31, 1978, received)