A WORD RECOGNITION METHOD FROM A CLASSIFIED PHONEME STRING IN THE LITHAN SPEECH UNDERSTANDING SYSTEM

Sei-ichi Nakagawa and Toshiyuki Sakai

Department of Information Science
Kyoto University
Kyoto, Japan

Abstract

In this paper, we describe an optimum matching method between a classified phoneme string and a phoneme string of a lexical entry in a word dictionary. This method is performed by using a phoneme similarity matrix and a dynamic programming technique. A classified segment consists of the first candidate, second candidate, reliability and duration. The effect of coarticulation is normalized in this matching procedure.

We also describe a word spotting method in continuous speech by modifying this method.

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. Therefore, for limited speakers, it is fairly easy to recognize spoken words in a limited vocabulary. 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 this paper, we describe an optimum matching between two phoneme strings in the LITHAN speech understanding system[1,2].

2. Acoustic Processor and Phoneme Classification

The system first analyzes input speech by the 20 channel 1/4-octave filterbank. 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(/h/,/p,t,k/), voiceless-nonplosive(/s/,/c/,/h/) 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: /s/, /c/, /h/, /p,t,k/. 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.

The decided stationary and quasi-stationary parts are regarded as presenting portions for vowels. The most stationary frames are used for partially non-supervised learning of the spectrum patterns of vowels[3]. 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 phonemes. 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 phonemes, 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.

This phoneme string is translated into a word. It is based on matching technique of a recognized phoneme string against a phoneme string given by an entry in a word dictionary. This matching uses a phoneme similarity matrix and dynamic programming (DP).

3. Word Classification from Phoneme String

3.1 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.

CH1285-6/78/0000-0726$00.75©1978IEEE

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 in the word dictionary. To evaluate a degree of matching between two phonemes, we introduce a concept of similarity between two phonemes.

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 linear transformation of the distance. If either i or j is unvoiced, $S(i,j)$ is derived from the confusion matrix. The resulting similarity matrix is shown in Table 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 1 Phoneme similarity matrix

In \ Out	*a*	*i*	*u*	*e*	*o*	*m*	*n*	*ŋ*	*p*	*t*	*k*	...
a	100	36	51	69	75	51	55	58	5	5	5	⋮
i	36	100	83	73	56	72	74	85	50	50	50	⋮
u	51	83	100	74	80	81	83	89	5	5	5	⋮
e	69	73	74	100	69	59	69	74	5	5	5	⋮
o	75	56	80	69	100	64	65	75	5	5	5	⋮
m	61	82	91	69	74	100	92	86	5	5	5	⋮
n	65	84	93	79	75	92	100	88	5	5	5	⋮
ŋ	68	95	99	84	85	86	88	100	85	85	85	⋮
*	5	5	5	5	5	5	5	5	100	100	100	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

3.2 Word Dictionary

Some phonemes in a word are often influenced by phoneme environments, while other phonemes are sometimes devocalized. 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.

Table 2 shows a part of the word dictionary. The special symbols (+ and -) indicate changes of restrictions for DP matching. 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 2 Example of entries in the Word Dictionary

Word	Symbol	Phoneme Representation	Maximum Duration	Minimum Duration
ichi	1	i ·/c(1.0) c i/c (1.0)	350ms	100ms
ni	2	n i	300	100
san	3	s a N	550	200
yon	4	y/g(0.9) o N	450	150
go	5	g o	300	100
roku	6	r/p(0.85) o ·/k(0.95) k u/*(1.0)	450	100
nana	7	n a/N(0.85) n/a(0.85) a/N(0.85)	550	200
hachi	8	h a/N(0.85) ·/c(1.0) c 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

3.3 Lexical Matching Procedure

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 for the second candidate ((difference of the both reliability)/(sum of the both reliability)), and the segment duration. The first three constituents of the j-th segment in a string 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 equation:

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

We simply denote $S(I,k,c;J,l,p)$ as $S_0(i,j)$. Any phoneme has a corresponding significant coefficient on the matching procedure (vowel=3, semi-vowel=1, voiced consonant=1, unvoiced consonant=2, devocalized vowel=1, etc.). This coefficient is the reflective of *a priori* knowledge on the performance of Phoneme Recognizer and the significance of a phoneme in Japanese.

We make the following restrictions with respect to the matching between an input string and an element string in the word dictionary. 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 consonant 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 (+).
(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.

Now, we consider how to calculate the likelihood for a spoken word in isolation. 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 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 equation:

$$L(i,j)=\max\{L_1(i,j),\ L_2(i,j),\ L_3(i,j),\ L_4(i,j),\ L_5(i,j),\ L_6(i,j)\},$$

where

$$L_1(i,j)=L^*(i-1,j)+w(i)\times S_0(i,j)$$
$$L_2(i,j)=L(i-1,j-1)+w(i)\times S_0(i,j)$$
$$L_3(i,j)=L^*(i-1,j-1)+w(i)\times\{S_0(i,j-1)+S_0(i,j)\}/2$$
$$L_4(i,j)=L(i-1,j-2)+w(i)\times\{S_0(i,j-1)+S_0(i,j)\}/2$$
$$L_5(i,j)=L^*(i-1,j-2)+w(i)\times\{S_0(i,j-2)+S_0(i,j-1)+S_0(i,j)\}/3$$
$$L_6(i,j)=L(i-1,j-3)+w(i)\times\{S_0(i,j-2)+S_0(i,j-1)+S_0(i,j)\}/3$$

where $w(i)$ represents a significant coefficient of the i-th main phoneme in the lexical entry. 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. The boundary (or initial) conditions are the following:

$$L(1,j)=0 \quad : j\geq 4$$
$$L(1,1)=w(1)\times S_0(1,1)$$
$$L(1,2)=w(1)\times\{S_0(1,1)+S_0(1,2)\}/2$$
$$L(1,3)=w(1)\times\{S_0(1,1)+S_0(1,2)+S_0(1,3)\}/3$$

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:

$$L(i,j)=\max\{L_1(i,j), L_2(i,j), L_3(i,j), L_4(i,j)\}$$
$$L(1,3)=0$$

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 as $L(i_0,j_0)/\sum_{i=1}^{i_0}w(i)$. Input utterance is recognized as a word which has the highest likelihood out of all words. Fig.1 shows the graphic representation of word matching.

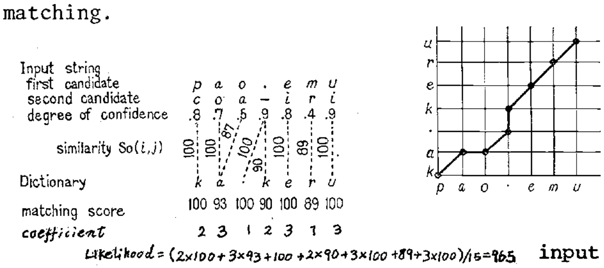

Fig. 1 Graphic representation of word matching and matching score

3.4 Normalization of Coarticulation

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 the previous section, 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:

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

Now let us consider matched combinations such as illustrated in Fig.2. For the cases (a) and (b) in the figure, the above equation 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.2, if we regard the association between x_i and $\{y_{j-1}, y_j, y_{j+1}\}$ as valid association, we can assume that y_{j-1} or y_{j+1} is a transient segment between either 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 phoneme between two successive phonemes.

$$S_0(i-1,j-1)>S_0(i-1,\bar{i})$$
$$S_0(i+1,j+1)>S_0(i+1,\bar{i}),$$

where both a and b in $S_0(a,\bar{b})$ denote the a-th and b-th elements in the entry. On the other hand, if that association is regarded as invalid association, 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:

$$S(x_i; y_{j-1}, y_j, y_{j+1})= S_0(i,j-1)+k\{S_0(i-1,j-1)-S_0(i-1,\bar{i})\}+S_0(i,j)+S_0(i,j+1)+k\{S_0(i+1,j+1)-S_0(i+1,\bar{i})\}\}/3$$

where k is a constant value. Fig.2 (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:

$$S(x_i; y_{j-1}, y_j)=\{S_0(i,j-1)+k\{S_0(i-1,j-1)-S_0(i-1,\bar{i})\}+S_0(i,j)+k\{S_0(i+1,j)-S_0(i+1,\bar{i})\}\}/2$$

4. Augmented Matching Methods for Continuous Speech

4.1 Sequential Matching Method

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.

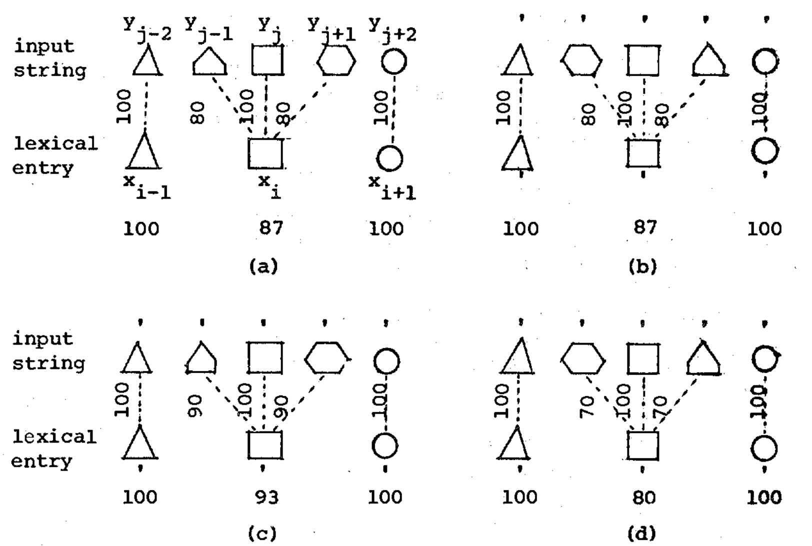

Fig. 2 Graphic model of normalization of coarticulation.
(a) and (b) : before normalization
(c) and (d) : after normalization
where, we assume that S(□,⌂)=S(□,⬡)=S(△,⌂)=S(○,⬡)=80, S(△,□)=S(○,□)=60, S(△,⬡)=S(□,○)=40, k=0.5.

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

We call this method a 'sequential matching method'(one end-point free method), the formal version of which is shown in Fig.3. The last phoneme $x'_{I'}$ of the preceding word is assumed to have been associated with the segment y_r. The concatenation $x'_{I'}$, $x_1 x_2 \cdots x_I$ of the element $x'_{I'}$, and the next word $x_1 x_2 \cdots x_I$ 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:

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

where $r+[(I+1)/2] \leq t \leq \min(r+I+5, r+2I-4)$, and $L(I,t_1)/\sum_{i=I'}^{I} w(i)$ 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 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}.

4.2 Word Spotting Method[4]

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'[5,6]. 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 ending point of dynamic programming at arbitrary points, that is, $L(I,j)/\sum_{i=0}^{I} w(i)$ indicates the likelihood of the presence of the word ending at segment j of the recognized phoneme string. This function equals the following modification in the sequential matching method:

$$L(1,j) = w(1) \times \max\{S_0(1,j),\ [S_0(1,j-1)+S_0(1,j)]/2,\ [S_0(1,j-2)+S_0(1,j-1)+S_0(1,j)]/3\} \quad ; j \geq 4$$

We call this method 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.4. An example of matching by this method is shown in Fig.5. The utterance is "Keisanki cyūo-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 (threshold=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].) The key word "sōchi" was detected at four locations, although the number of its utterance was one.

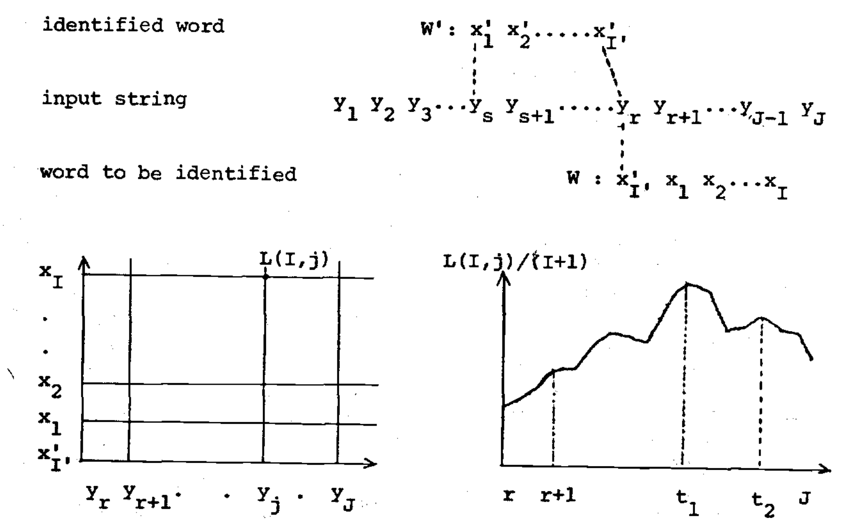

Fig. 3 Sequential matching method (one end-point free method).

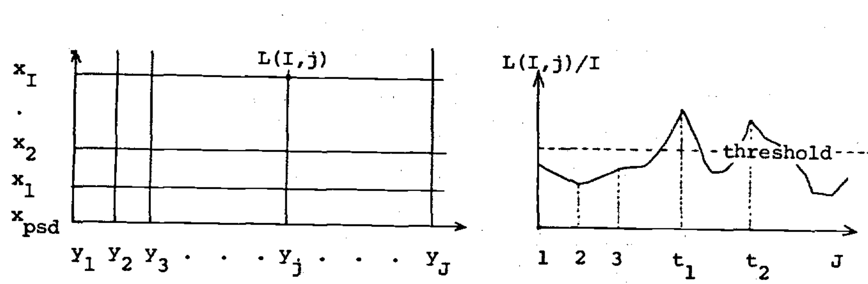

Fig. 4 Direct matching method (both end-points free method).

```
input                 kesaN ki cyūono zikitēpu sō  ci saNbaNkarakesaN kigazō e d ē  tayoNorō doseyo
segment number        0.......1.......2.......3.......4.......5.......6.......7.......0.......1.......
first candidate       /pucau.pi.cuuoro-puciges.csou.ci.caobaopao-pucaN.pidazouu.bgeu.cabogoroogusee/
second candidate      -cisegpcuphiNuuu.hihuducppcuopsbpsoNrorcor/cisNuphuŋoduee-ubuNphonNuubeudocug-
                      61  6  11612642    13 24916 29229 3 73 2  11 1313 31 14 342 1 9471 22113 243 6
reliability (x100)    0662000066242602080606809806092898048400002082266486062000444698446646044444440

             main-pho.  / k e i s a N : k i       s o . c i
             sub-pho.   k   e       k   k           c   c
                                        1           1   1
lexicon      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
sōchi        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. 5 Detection of "keisanki[=computer]" and "sōchi [=device]" by direct matching method.

5. Experimental Results

5.1 Isolated Spoken Digits Recognition

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 recognition method was tested for ten isolatedly uttered digits. The lexicon is the same ones as shown in Table 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 spectra 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. The performance of Phoneme Recognizer was about 76% for vowels, 13% for semi-vowels, 8% for voiced consonants, and 64% for unvoiced consonants(/p,t,k/ was regarded as an identical phoneme) in *Experiment 1.* The recognition rates of *Experiments 1, 2 and 3* were 96.4%, 97.4% and 97.6%, respectivrely.

5.2 Recognition of Three Successive Vowels

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 reference patterns were common to all speakers. The recognition rate without normalization was 93.6%, while normalization improved the rate to 95.6%.

5.3 Key Words Detection in Continuous Speech

The LITHAN speech understanding system was applied to restricted utterances with vocabulary of about 100 words which concerned with operational commands and queries of the status of a computer network. According to the results tested on a sample 200 sentences spoken by 10 male adults at a normal speed, 64% of the sentences and 93% of the output words were correctly recognized by using common reference patterns to all speakers. The System firstly tried to detect three key words ("dengen", "keisanki", "sōchi") in continuously uttered speech of 200 sentences containing 1983 words. Except two, all of the 352 key words were correctly detected. In these procedures, the number of detected places as key word existing locations was almost as 2.5 times as those where key words actually appeared.

6. Conclusion

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.

References

1. T.Sakai and S.Nakagawa : A Speech Understanding System of Simple Japanese Sentences in a Task Domain, Jour. of IECEJ, Vol. 60E, No.1 (1977)
2. T.Sakai and S.Nakagawa : Speech Understanding System LITHAN and Some Applications, 3rd IJCPR, Coronado (1976)
3. T.Sakai and S.Nakagawa : A Classification Method of Spoken Words in Continuous Speech for Many Speakers, Information Processing in Japan, Vol. 17 (1977)
4. T.Sakai and S.Nakagawa : A Word Identification Method in Continuous Speech, Reports of the 1975 Spring Meeting of ASJ (in Japanese)
5. J.K.Baker : Stochastic Modeling for Automatic Speech Recognition, Ed. by D.R.Reddy, Academic Press (1975)
6. R.W.Christiansen and C.K.Rushforth : Detecting and Locating Key Words in Continuous Speech Using Linear Predictive Coding, IEEE Trans. Vol. ASSP-25, No.5 (1977)

The detailed descriptions of the LITHAN speech understanding system are shown in "A Machine Understanding System for Spoken Japanese Sentences" by Sei-ichi Nakagawa, Doctral Thesis, Kyoto University (1976).