A New Word Pre-Selection Method Based on an Extended Redundant Hash Addressing for Continuous Speech Recognition

著者	Ito Akinori, Makino Shozo
journal or publication title	IEEE International Conference on Acoustics, Speech, and Signal Processing, 1993. ICASSP-93
volume	1993
number	2
page range	299-302
year	1993
URL	http://hdl.handle.net/10097/46625

doi: 10.1109/ICASSP.1993.319296

A NEW WORD PRE-SELECTION METHOD BASED ON AN EXTENDED REDUNDANT HASH ADDRESSING FOR CONTINUOUS SPEECH RECOGNITION

Akinori ITO † *Shozo MAKINO* ‡

† Education Center for Information Processing, Tohoku University
‡ Research Center for Applied Information Sciences, Tohoku University
Sendai 980 Japan

1. INTRODUCTION

Several speech recognition systems for dictation or dialogue have been developed [1][2]. These systems have top-down prediction of words for the recognition using grammatical information. Although the use of grammar makes the recognition process efficient, it is not sufficient for reducing computational load against increment of the vocabulary size. To reduce the recognition time and to realize an on-line continuous speech recognition system with a large size vocabulary, it is important to develop an efficient and accurate bottom-up word pre-selection method.

In this paper, a new word pre-selection method called "extended redundant hash addressing method" is proposed. This method is based on isolated word recognition method using the redundant hash addressing principle proposed by T.Kohonen et.al[4]. The redundant hash addressing method can not be used for continuous speech recognition because the word boundary in continuous speech is not known. We have extended Kohonen's method to continuous speech recognition. The proposed method spots words from the input speech using the information about locations of features in reference patterns. Combination of this method and an ordinary parser makes a speech recognition system faster. At last several experimental results are shown to compare this method with the continuous-DP matching method which is a traditional word spotting method. The proposed method gives comparable performance to that with the continuous-DP matching and is five times faster than it.

2. WORD RECOGNITION BASED ON THE REDUNDANT HASH ADDRESSING METHOD

The redundant hash addressing method recognizes an isolated spoken word based on reference to a dictionary using "features" extracted from an input phoneme sequence. Two or three consecutive phonemes (called *bigrams* or *trigrams*) is used as features. Moreover, the hash addressing technique is used to look up a dictionary entry effectively. First, cyclic trigrams are extracted from each entry in the dictionary. Each trigram is associated with the pointer to the dictionary entry. These pointers are registered in a hash table. Next, the trigrams are extracted from a phoneme sequence recognized from the input speech. The extracted trigram votes to word candidates using hash search. Finally, all word candidates are scored based on the number of votes by the extracted trigrams. The word candidate with the highest score is selected as a recognition result. The redundant hash addressing method is very fast because the input sequence is not compared with all entries in the dictionary; only the words having similar sub-sequences to the input sequence are tested.

Though this method is very fast, it has some disadvantages. As this method uses the trigrams extracted from words in the dictionary without considering phoneme recognition errors, the recognition score is not higher than that using the DP-matching-based word recognition.

3. THE EXTENDED REDUNDANT HASH ADDRESSING METHOD

In this section, we propose the extended redundant hash addressing method. This method has the following improvements on the redundant hash addressing: (1) utilization of confusion matrix of the phoneme recognition to make dictionary reference more accurate, and (2) activation point matching for spotting words using the references from trigrams.

In order to make reference from a trigram to the dictionary more accurate, we propose the method which generates *extended trigrams* from an original trigram extracted from the dictionary entry. The extended trigrams are generated using the phoneme recognition error probabilities in the confusion matrix. The generated extended trigrams are used to search the entry. Figure 1 shows an example of the generation of the extended trigrams. First, the trigram is extracted from a dictionary entry. Next, phoneme candidates are generated using the confusion matrix. For substitution error, several phonemes with high substitution probability is selected for each phoneme in the original trigram; then several extended trigrams with high probabilities are picked up from all combinations of these phonemes. In the same fashion,

0-7803-0946-4/93 $3.00 © 1993 IEEE

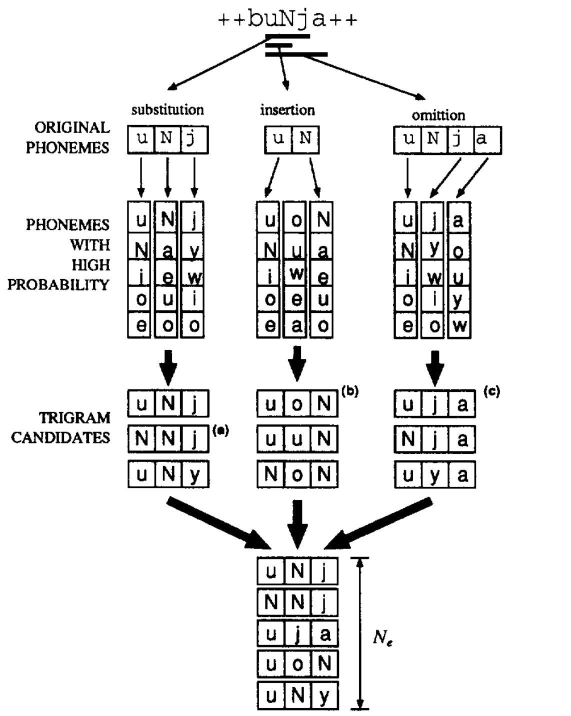
++buNja++
substitution
insertion
omittion
ORIGINAL PHONEMES
u N j
u N
u N j a
PHONEMES WITH HIGH PROBABILITY
u N j / N a y / i e w / o u i / e o o
u o N / N u a / i w e / o e u / e a o
u j a / N y o / i w u / o i y / e o w
TRIGRAM CANDIDATES
u N j / N N j (a) / u N y
u o N (b) / u u N / N o N
u j a (c) / N j a / u y a
u N j / N N j / u j a / u o N / u N y
N_e

Figure 1: An example of the generation of extended trigrams

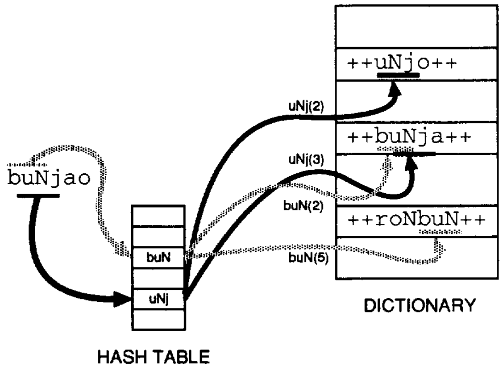

Figure 2: Dictionary look-up of the extended redundant hash addressing method

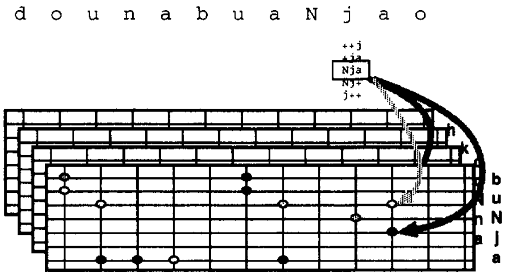

Figure 3: Determine dictionary entries from a trigram

the extended trigrams are generated for phoneme omission and insertion. Finally, a certain number (N_e) of the extended trigrams are generated from the original trigram. These trigrams are registered to the hash table instead of the original trigram. They have scores based on the probabilities of phoneme substitution, omission and insertion. Let $P_s(p_1,p_2)$ be the substitution probability from phoneme p_1 to p_2, $P_i(p_1)$ be the insertion probability of phoneme p_1 and $P_o(p_1)$ be the omission probability of phoneme p_1. The score of the trigram is calculated as sum of logarithm of these probabilities. For example, score of the trigram (a) in figure 1 is calculated as

$$\log P_s(/u/,/N/) + \log P_s(/N/,/N/) + \log P_s(/j/,/j/)$$

similarly, (b) and (c) are calculated as

$$\log P_s/u/,/u/) + \log P_i(/o/) + \log P_s(/N/,/N/)$$

and

$$\log P_s(/u/,/u/)+\log P_o(/N/)+\log P_s(/j/,/j/)+\log P_s(/a/,/a/)$$

respectively.

Figure 2 shows the dictionary reference using the extended redundant hash addressing method. Not only the dictionary entry, but also the location of the extended trigram in the entry is determined from the trigram.

The following five cases are considered for the location of the extracted trigrams from the input sequences: The center of extracted trigram is (1) located at one phoneme after the word-initial, (2) at the word-initial, (3) at the word-medial, (4) at the word-final and (5) at one phoneme before the word-final. Once the trigram is extracted from the input sequence, several dictionary entries are determined through the hash table. Figure 3 shows an example how an entry is retrieved using the trigrams extracted from the input sequence. From one trigram, the following information can be known: the dictionary entry: w, the location of the trigram in the entry: j and the location of the trigram in the input sequence: i. We call the set (i,j,w) an *activation point.*

Figure 4 shows an example of the activation point matching. In this example, a word "*buNja*(field)" is matched to the input sequence. The five trigrams shown in the upper part of the figure are extracted at each phoneme location of the input sequence. These trigrams determine scores and locations in a word candidate (expressed as the color of a circle in the figure). A word candidate is spotted from an input sequence by connecting activation points and by accumulating these scores.

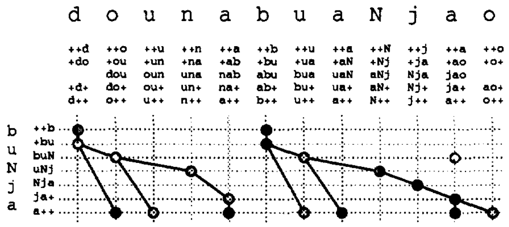
d o u n a b u a N j a o
++d ++o ++u ++n ++a ++b ++u ++a ++N ++j ++a ++o
+do +ou +un +na +ab +bu +ua +aN +Nj +ja +ao +o+
dou oun una nab abu bua uaN aNj Nja jao
+d+ do+ ou+ un+ na+ ab+ bu+ ua+ aN+ Nj+ ja+ ao+
d++ o++ u++ n++ a++ b++ u++ a++ N++ j++ a++ o++
b ++b
u +bu
buN
N uNj
j Nja
ja+
a a++

Figure 4: Activation point matching

Connections with the least penalty between activation points are checked to spot words from the input sequence. To search the optimum connection to the present activation point (i_0, j_0, w), all preceding activation points have to be checked; but this check requires large computation amount and it lose the advantage of this word pre-selection method. To make the search faster, we restricted the activation points for connection to the last activated points within the same i. Under this restriction, the sub-optimum connections are calculated using a dynamic programming. An algorithm to connect activation points is shown in Figure 5. This algorithm connects the present activation point (i_0, j_0, w) to a proper preceding activation point. In this algorithm, $SCORE(t)$ represents the score of a trigram t, J is the length of the input sequence, $TSCORE(i, j, w)$ is the accumulated score of an activation point (i, j, w), $L(j, w)$ is the location of the last activated point in the input sequence and $PEN((i_1, j_1), (i_2, j_2))$ is the penalty score between two activation points to be connected. In the following experiments, PEN is defined as follows:

$$PEN((i_1, j_1), (i_2, j_2)) = (|i_2 - i_1| + |j_2 - j_1| - 2) \times P$$

where P is a constant lower than a score of any trigram ($P < 0$). A word is spotted when the activation point reaches to the position of the final trigram in the word.

In this algorithm, candidates of the activation point for connection are checked within the range of $j = 1 ... j_0 - 1$. This range can be $j = min(j_0 - K, 1) ... j_0 - 1$ to avoid a connection between too distant activation points and to reduce the calculation time. The effect of this restriction is investigated in the experiment later.

4. EXPERIMENTS

Word spotting experiments were carried out to compare the extended redundant hash addressing method to the continuous-DP matching. Input sequences were generated by computer simulation based on the confusion matrix. They were 136 sentences from an scientific article and the assumed phoneme recognition rate is 85%.

At first the effectiveness of the extended trigrams was investigated. To check the effect of number of generated trigrams (N_e), word spotting tasks were carried out for various N_e. The dictionary size for the experiment was 847. The result of this experiment is shown in Figure 6. The word detection performances of the extended RHA method were compared with that of the continuous-DP matching. According to this experiment, it was found that the word detection performance of the extended RHA method was comparable to that of the continuous-DP matching and $N_e = 10$ gave the best performance for word detection.

In the next experiment, the relation between the size of the dictionary and the recognition speed was investigated. Process times (without time for preprocessing) of the two methods were compared for 364, 500, 600, 700 and 847 words dictionary. Figure 7 shows the result of the experiment. This result showed that the extended RHA method was about five times faster than the continuous DP matching.

Finally, the effect of the restriction for the connection range was investigated. Figure 8 shows the

```
$L(j, w) \leftarrow$ none   for all $1 \leq j \leq J$
begin
    $maxscore \leftarrow$ VERY_SMALL_VALUE;
    for $j \leftarrow 1$ to $j_0 - 1$ do begin
        $score \leftarrow SCORE(t) +$
                 $TSCORE(L(j, w), j, w) +$
                 $PEN((i_0, j_0, w), (L(j, w), j, w))$;
        if $score > maxscore$ then begin
            $maxscore \leftarrow score$;
        end
    end
    $TSCORE(i_0, j_0, w) \leftarrow maxscore$;
    $L(j_0, w) \leftarrow i_0$
end
```

Figure 5: An algorithm to connect activation points

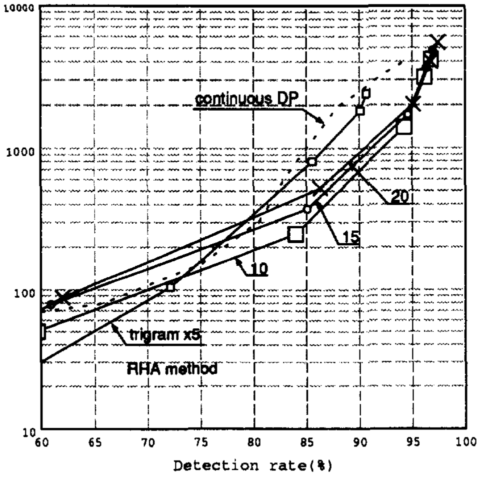

Figure 6: Word detection performance for various N_e

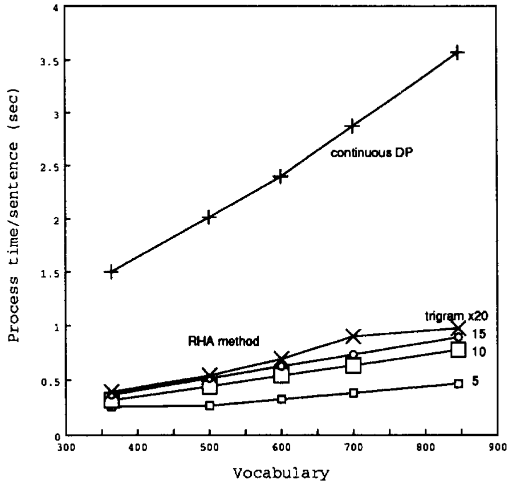

Figure 7: Process times for various size of dictionary

word detection performance for various K ("maximum gap" in the figure). In this experiment, N_e was set to 10. From this figure, it is clear that $K = 6$ gave as good performance as $K = \infty$ where $K = \infty$ means no restriction for the activation point matching. Figure 9 shows that the process time at $K = 6$ is about 14% shorter than the process time at $K = \infty$. When K is set to 6, the processing time is about five times faster than that of continuous DP matching.

5. CONCLUSION

A new word pre-selection method "extended RHA method" for continuous speech recognition was proposed. This method extends the redundant hash addressing method to word spotting from continuous speech. Moreover, the improvement of the trigram extraction makes matching score more accurate than the redundant hash addressing method. The word spotting experiments showed that the proposed method gave comparable word spotting performance to the continuous-DP matching and that the proposed method was about five times faster than the continuos-DP matching.

6. REFERENCES

[1] T. Hanazawa, K. Kita, S. Nakamura, T. Kawabata and K. Shikano: *ATR HMM-LR Continuous Speech Recognition System*, Proc. ICASSP90, pp.53-56 (1990-4)

[2] M. Okada: *A Unification-Grammar-directed One-Pass Search Algorithm for Parsing Spoken Language*, Proc. ICASSP91, pp.721-724 (1991-5)

[3] S.Makino, A.Ito, M.Endo and K.Kido: *A Japanese Text Dictation System Based on Phoneme Recognition and a Dependency Grammar*, Proc. ICASSP91 (1991-5) ,Vol.E 74, No.7 (1991-7)

[4] T.Kohonen, H.Riittinen, E.Reuhkala and S.Haltsonen: *On-Line Recognition of Spoken Words from a Large Vocabulary*, INFORMATION SCIENCES 33, pp.3-30 (1984)

[5] S.Tanaka, A.Ito, M.Makino, T.Sone and K.Kido: *A high speed method for detection of Bunsetsu-unit in the Japanese text dictation system*, IEICE Technical Report SP-70 (1990-12, in Japanese)

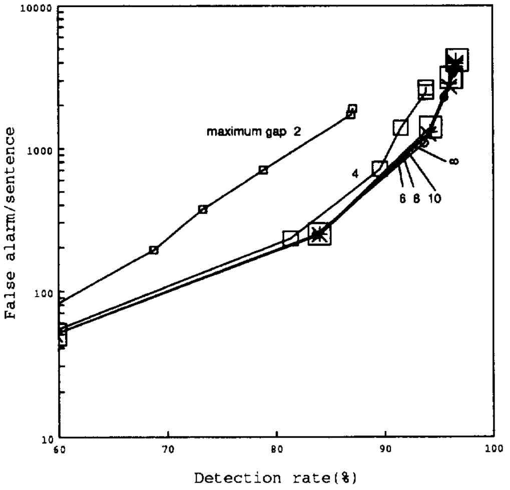

Figure 8: Word detection performance for various K

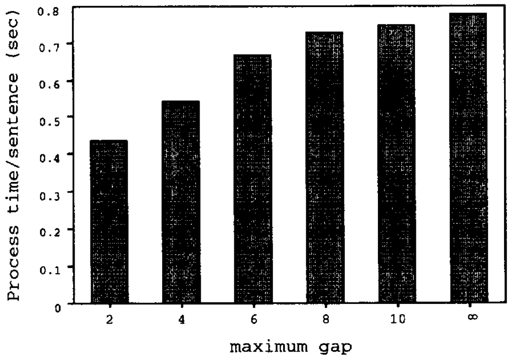

Figure 9: Process times for various K