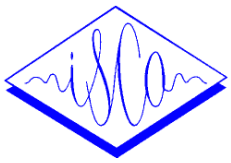

First International Conference on
Spoken Language Processing (ICSLP 90)
Kobe, Japan
November 18-22, 1990

A RECOGNITION TIME REDUCTION ALGORITHM FOR LARGE-VOCABULARY SPEECH RECOGNITION

J. M. Koo, C. K. Un, H. S. Lee, H. R. Kim and M. W. Koo

Communications Research Laboratory
Department of Electrical Engineering
Korea Advanced Institute of Science and Technology
P.O. Box 150, Chongyangni, Seoul, Korea

ABSTRACT

We propose an efficient pre-classification algorithm extracting candidate words to reduce the recognition time in a large-vocabulary recognition system and also propose the use of spectral and temporal smoothing of the observation probability to improve its classification performance. The proposed algorithm computes the coarse likelihood score for each word in a lexicon using the observation probabilities of speech spectra and duration information of recognition units. With the proposed approach we could reduce the computational amount by 74% with slight degradation of recognition accuracy in a 1160-word recognition system based on the phoneme-level HMM.

I. INTRODUCTION

As the speech recognition technology progresses, many algorithms have been proposed for large-vocabulary speech recognition. Among these algorithms, the one based on hidden Markov modeling(HMM) which requires much less recognition time than other template-matching based approaches is known to be a viable method for the practical usage. Particularly, the HMM-based approaches using sub-word unit models are widely used due to its ability of easy construction of word models from the sub-word models. Although the HMM based algorithms have an advantage in recognition time, it becomes difficult to recognize utterances in real time as the size of vocabulary grows. To alleviate this problem, several time reduction algorithms have been proposed[1][2]. One example is the two-pass algorithm in which candidate words are first selected for the second-stage classification[3].

In this paper, a pre-classification algorithm based on the detection probability and duration information of the recognition units are proposed to reduce the total recognition time. The computational gain is considered when the proposed algorithm is used as the first-stage classifier of a large vocabulary recognition system utilizing the phoneme level HMM. Also, its classification performance is examined and compared to that of the first-stage classifier based on the vowel classification.

II. DESCRIPTION OF ALGORITHM

The pre-classification algorithm consists of two phases; training and classification phases. In the training phase, the probability distribution of speech spectra and the durational information are examined for each recognition unit. In this work, the probability distribution of speech spectra is estimated by a non-parametric method to reduce computation. For this estimation, input speech spectra are vector quantized(VQ), and the relative frequencies of VQ codewords in recognition units are computed. Assuming that the number of the recognition units is R and the number of VQ codewords is M, the j-th VQ codeword observation probability of i-th recognition unit, $f_i(j)$, is obtained for $1 \leq i \leq R$, $1 \leq j \leq M$. Also, the minimum and maximum durations of each recognition units, $d_{\min}(i)$ and $d_{\max}(i)$, are obtained for $1 \leq i \leq R$ during the training phase.

In the classification phase, the coarse likelihood score is computed for every word $w_k (1 \leq k \leq V)$ in a lexicon of size V, and a part of them are chosen as candidate words for the second-stage recognition according to their likelihood scores. When an input speech is uttered, the feature is extracted and encoded to a VQ index by vector quantization process for every frame of the input speech. Then, the input speech is represented by a series of VQ indices $\mathbf{O}=\{O_1, O_2, \ldots, O_T\}$. From this series of indices, the detection probability of every recognition unit, $\mathbf{P}_i=\{p_i(1), p_i(2), \ldots, p_i(T)\}$, is obtained for $1 \leq i \leq R$ where $p_i(t)=f_i(O_t)$. Then, the coarse likelihood score L_k for k-th word w_k which is composed of n recognition units $\{r_1, r_2, \ldots, r_n\}$ is computed as follows. First, a possible starting point s_i and an ending point e_i of the recognition unit $r_i (1 \leq i \leq n)$ are computed as

$$s_i = min\left(\sum_{j=1}^{i-1} d_{\min}(r_j), T\right), \tag{1}$$

$$e_i = min\left(\sum_{j=1}^{i} d_{\max}(r_j), T\right), \tag{2}$$

where $min(x, y)$ is equal to x if $x \leq y$, or y otherwise. The coarse likelihood score L_k for word w_k is computed as

$$L_k = \prod_{t=1}^{T} \max_{r_i \, s.t \, s_i \leq t \leq e_i} (p_{r_i}(t)), \tag{3}$$

where $\max_c(x)$ chooses the maximum value of x under the condition c. According to these values $L_k (1 \leq k \leq V)$, a part of vocabulary are chosen as candidate words and the second-stage classification is performed for those words.

As we explained above, the time reduction algorithm requires the vector quantization process. If the proposed algorithm is used as a preprocessor of an HMM-based speech recognition system which requires vector quantization, the computation can be reduced further by sharing the VQ process.

Like the recognition system based on the discrete HMM, the performance of the time reduction algorithm largely depends on the parameter estimation procedure of training phase. That is, if the observation probability of speech spectra is estimated from a small amount of the training data, the performance of the pre-classification becomes degraded significantly. To alleviate this problem in the training phase, we adopt a spectral smoothing method which smoothes the probability distribution by the fuzzy mapping concept[4]. If a VQ codeword, O_t which shows a high observation probability for a recognition unit r_i is observed, the codewords observed near the observation time t will also show a high observation probability for the same unit. To accommodate this tendency in the classification phase, we propose a temporal smoothing method which smoothes the recognition unit observation probability $p_i(t)$ as

$$\hat{p}_i(t) = (c_1 \cdot p_i(t-1) + c_2 \cdot p_i(t) + c_3 \cdot p_i(t+1))/(c_1 + c_2 + c_3), \tag{4}$$

where c_1, c_2 and c_3 are constants. The temporal smoothing compensates for the fact that each output codeword is treated independently in (3). We used the above two smoothing methods sequentially, and observed their contributions to the classification performance by computer simulation.

III. CONSIDERATION OF COMPUTATIONAL GAIN

We now compare the amount of computation of the recognition system based on the phoneme-level HMM with the proposed time reduction algorithm to that of the recognition system without the proposed algorithm. Since the feature extraction and vector quantization procedure can be shared, and the time consumed is negligible as compared to that of the classification procedure in a large vocabulary recognition system, we will consider only the time required for classification. We used a three-state phoneme HMM with

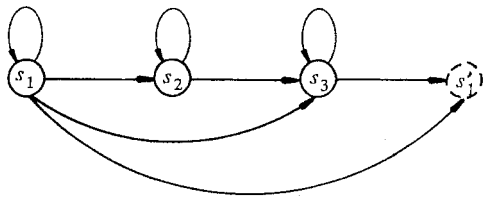

Fig. 1. Left-to-right HMM for a phoneme

eight transitions as shown in Fig. 1. Consider a word which consists of n phonemes and whose length is T. To compute the Viterbi score, $[8(T-n)-12]\cdot n$ multiplications, the same number of additions and $3(T-n-1)\cdot n$ comparisons are required. For simplicity, if we assume that the time for all operations are equal to time τ and every word in the lexicon has the same number of phonemes and the same length, then the time required to recognize a word by full search of the lexicon of size V is $(19nT-19n^2-27n)\cdot\tau\cdot V$. To choose the candidate words by the proposed time reduction algorithm, we should compute $\mathbf{P}_i\ (1\leq i\leq R)$ once and L_k for every word $w_k\ (1\leq k\leq V)$. Since $V\gg R$, the computation required for calculating $\mathbf{P}_i$ can be neglected, and the number of operations required to compute L_k are less than $n\cdot T$ comparisons and T multiplications. But, for simplicity, let the computation time required to perform the first-stage classification be $(n+1)\cdot T\cdot\tau\cdot V$. Then the ratio of the computation time of the full search method to that of the first-stage classification is

$$G = 19\frac{n}{n+1} - 19\frac{n^2}{(n+1)T} - 27\frac{n}{(n+1)T} \qquad (5)$$

We applied the proposed algorithm to a recognition system with $V=1160$, $R=44$. The average length of input words was 84.4 frames and the average number of phonemes in a word was 9.32. Substituting these values in (5), we get the computational overhead $G^{-1}=6\%$ approximately. That is, the time for the first-stage classification is only 6% of the time required for performing the Viterbi score computation for every word in a lexicon. If we choose ν words as candidate words from the lexicon by the pre-classification algorithm, the total time required to recognize an input speech is approximately equal to $(\nu\cdot 100/V+6)\%$ of the time required when the first-stage classification is removed. Through our computer simulation, we confirmed that the real amount of recognition time reduction was almost the same as our estimation.

IV. SIMULATION RESULTS

The performance of the proposed time reduction algorithm was tested in a speaker-independent isolated word recognition system. The recognition unit was 44 Korean context-independent phoneme-level HMM and the size of the lexicon was 1160 which are used for an automatic telephone number information query system. For training, we used a speech data base consisting of 75 phonetically balanced Korean words uttered by 5 male speakers. The test data base of size 1160 words was constructed from the utterances of another male speaker. All input utterances were low-pass filtered with cut-off frequency of 4.5 KHz and digitized with the sampling frequency of 10 KHz. End points were detected manually, and phonemes were hand-segmented for the training data. But, the end points were detected automatically in the test procedure. Twelve-order linear predictive coding(LPC) cepstral coefficients and differenced LPC cepstral coefficients were obtained as the features in every 10 ms. We used two separate codebooks of size 256 for each feature, and treated them independently. The Viterbi algorithm was used to compute the likelihood of a model and a test utterance in the second-stage.

We compared the classification performances of the first-stage classification algorithms of different smoothing methods. Table 1 shows their inclusion rate according to its candidate word selection range for the second-stage classification. One can see from Table 1 that the fuzzy smoothing and the temporal smoothing improves the classification performance substantially, and that if we select 20% of the vocabu-

lary as candidate words, about 4% of input utterances are not included in the candidate word list. In this case, we can save approximately 74% of computation as compared to the system without the pre-classification stage. The recognition accuracy of the large vocabulary recognition system according to the different selection range are shown in Table 2. Since the recognition accuracy is not different from that of the full search case (selection range = 1.0) when more than 30% of the vocabulary are selected as the candidate words (selection range > 0.3), only three selection range were considered. Even when we set the selection range to be 0.1, the recognition accuracy degrades very slightly. In addition to this fact, considering the small average ranks (about 4% of the vocabulary size) in Table 1, we can assert that the proposed coarse likelihood score computation coincides with the likelihood score computed by the Viterbi algorithm very well. Note that the perplexity of the recognition system is 1160 and the mode of the recognition is speaker-independent.

Table 1. Average inclusion rate(%) for different selection ranges, average and maximum rank of spoken words in a candidate word list.

Algorithm	Selection range						avg.	max.
	0.1	0.2	0.3	0.4	0.5	0.6	rank	rank
Basic	83.3	92.1	95.3	97.3	98.4	99.2	67.3	1025
Spectral	89.7	95.8	97.9	98.3	99.2	99.7	44.6	855
Temporal	90.6	95.9	97.9	98.3	98.7	99.7	40.5	854

Table 2. Recognition accuracy(%) of the large vocabulary recognition system for different selection ranges. (the value in parenthesis shows the accuracy of recognizing the first two candidates)

Selection range	0.1	0.2	0.3	1.0
Recognition accuracy	59.05 (71.81)	59.31 (72.59)	59.48 (72.84)	59.48 (72.67)

We also compared the classification performance to that of the first-stage classification algorithm based on the classification of vowel class. By recognizing the sequence of vowel classes in an input utterance, they choose the candidate words for the second-stage classification. The algorithm chooses 20% of the vocabulary as the candidate words with the inclusion rate of 97% in a speaker-dependent mode. Although the classification performance is comparable to the proposed algorithm, the total computation time required is much longer than the proposed algorithm, since the vowel classification procedure requires the extraction of additional features such as formants.

V. CONCLUSION

In this paper, we proposed an efficient pre-classification algorithm based on the observation probability of speech spectra and duration information of recognition units for a large vocabulary recognition system. We also proposed two smoothing methods to improve the classification performance. We applied the proposed algorithm to a 1160-word recognition system based on the phoneme-level HMM and observed that we could reduce the computational amount by 74% with slight degradation of the recognition rate.

References

[1] L. Bahl et al. "Matrix fast match : A fast method for identifying a short list of candidate words for decoding," Proc. ICASSP 89, Paper S6.24.

[2] L. Fissore, G. Micca and R. Pieraccini, "Very large vocabulary isolated utterance recognition : A comparison between one pass and two pass strategies," Proc. ICASSP 88, Paper S5.6.

[3] T. Kaneko and N. R. Dixon, "A hierarchical decision approach to large vocabulary discrete utterance recognition," IEEE Vol. ASSP-31, pp. 1061-1066, Oct. 1983.

[4] J. M. Koo and C. K. Un, "Fuzzy smoothing of HMM parameters in speech recognition," Electron. Lett., Vol. 26, No. 11, pp. 743-744, May 1990.