EUROSPEECH '97
5th European Conference on Speech
Communication and Technology
Rhodes, Greece, September 22-25, 1997

FAST ALGORITHM FOR SPEECH RECOGNITION USING SPEAKER CLUSTER HMM

M. Yamada *Y. Komori* *T. Kosaka* *H. Yamamoto*
Media Technology Laboratory, Canon Inc.
890-12 Kashimada, Saiwai-ku, Kawasaki-shi, Kanagawa 211 Japan
E-mail:masayuki@cis.canon.co.jp

ABSTRACT

This paper describes a high speed algorithm for a speech recognizer based on speaker cluster HMM. The speaker cluster HMM, which enables to deal with variety among speakers, have been reported to show good performance. However, the computation amount grows in proportion to the number of clusters, when the speaker cluster HMM is used in speaker independent recognition, where the recognition processes must be run in parallel using every speaker cluster HMM. To reduce the computation, we introduced the multi-pass search for searching on the broad space covering lexical and speaker variation. Furthermore, the output probability recalculation is introduced to reduce the state output probability computation. We had some experiments on 1000 word speaker independent continuous telephone speech recognition. The result in the case where 7 speaker clusters are used shows about 30% of computation reduction.

1. INTRODUCTION

To achieve a higher recognition rate in speaker independent speech recognition, it is a successful way to use as detailed acoustic models as possible. Speaker cluster dependent HMM is such a model which enables to deal with variety among speakers[1]. The gender dependent HMM is popularly used as a simple implementation of the speaker cluster HMM[2, 3, 4].

When the speaker cluster models are used in a speaker independent recognizer, the computation grows in proportion to the number of clusters, because the recognizer doesn't know which cluster the speaker belongs to actually. The recognition processes must be run in parallel using every speaker cluster HMM. Hence, some systems with the gender dependent HMMs have steps for gender identification prior to the speech recognition[2], others utilizes the multi-pass search strategy[3]. In [5], the beam search is adopted to prune the search space. It becomes an important issue to reduce the processing time as the number of the speaker clusters increases.

This paper proposes a fast algorithm to deal with the speaker cluster HMM in the speaker independent speech recognition without cluster pre-selection. In the algorithm, the multi-pass search was used to reduce the lexical search space. Furthermore, the output probability recalculation algorithm[7] is introduced to reduce the computation amount for state output probability computation.

First, a speaker cluster HMM which we consider is described. The problem of the speaker cluster HMM used in speaker independent recognition without supervised cluster selection is also described. In section 3., our new method is proposed. Finally, our experimental results on 1000 word speaker independent continuous telephone speech recognition are shown.

2. SPEAKER CLUSTER HMM

Clustering speakers on training data, and designing HMMs on each cluster effectively improves recognition rate. For example, gender dependent HMM, which is a simple implementation of the speaker cluster HMM, is reported to show good performance in the large vocabulary task[2, 3, 4]. More detailed speaker clustering technique is also reported[1].

The speaker cluster HMM we use is organized into hierarchical tree-structure[1]. The root cluster in the hierarchy consists of all speakers in training data, that is speaker independent. The second clusters are gender dependent. As the hierarchy tree is traced to the leaf, the node on the tree represents the more detailed cluster.

To exploit the speaker cluster HMM in a speaker independent recognition, where information about the speaker is not explicitly given, the following two strategies are available.

1. Cluster pre-selection — After selecting the cluster where the speaker belongs, run recognition process on the selected cluster. As a means of the selection, result of the speech recognition using the speaker independent HMM can be utilized.
2. Probability competition — After running speech recognizers in parallel using every speaker cluster HMM, choose the result of the best score. When this algorithm is carried out with some kind of pruning technique, it can be implemented as in the following two ways.
   (a) After getting result for each speaker cluster, compare the results and pick the result of the best score.

(b) Search the best path directly in the (*speaker cluster*) × (*linguistic candidate*) space.

The cluster pre-selection has a risk that the mis-selection of a cluster can not be recovered and it harms recognition performance. On the other hand, the probability competition requires heavier computation.

The algorithm we propose here is based on the probability competition. But it requires lighter computation.

3. PROPOSED ALGORITHM

In the proposed algorithm, the multi-pass search is introduced to reduce lexical search space. We use the tree-trellis based search[8] as an instance of the multi-pass search. Furthermore, the output probability recalculation algorithm[7] is applied to reduce the computation amount for state output probability computation.

3.1. Tree-trellis based search on speaker cluster HMM

To reduce the lexical search space, we make use of the tree-trellis based search which was proposed for getting N-best results.

The tree-trellis based search consists of the following two passes: forward Viterbi search and backward best-first A* search. The result of the forward search is a lattice holding scores of partial paths. The forward result is used as heuristics in the backward A* search. Our idea is to share a common heuristics among every recognition process running on each speaker cluster HMM. In our implementation, the forward result using the root (speaker independent) cluster HMM is used as the common heuristics.

The A* condition for the optimality is no longer satisfied by changing HMMs between the forward and backward search. However, keeping enough N-best stack size is considered to save the search error.

The backward search can run a) separately on each speaker cluster HMM or b) simultaneously on every cluster using best-first criterion.

The Figure 1 shows a brief image of the algorithm described here. The (a) in the figure represents the case when the backward searches run separately, while the (b) represents the other case.

3.2. Output probability recalculation

When the speaker cluster HMMs are used, the amount of the output probability computation also increases depending on the number of the clusters. Although the search strategy described above denotes that the output probability for the non-root cluster HMM is only needed in the backward search, it directory affects the delay of the response to compute the probability in the backward step. On the other hand, computing it in the forward step requires heavy computation, because we must compute probability for all HMM states, not knowing which state appears in the backward search.

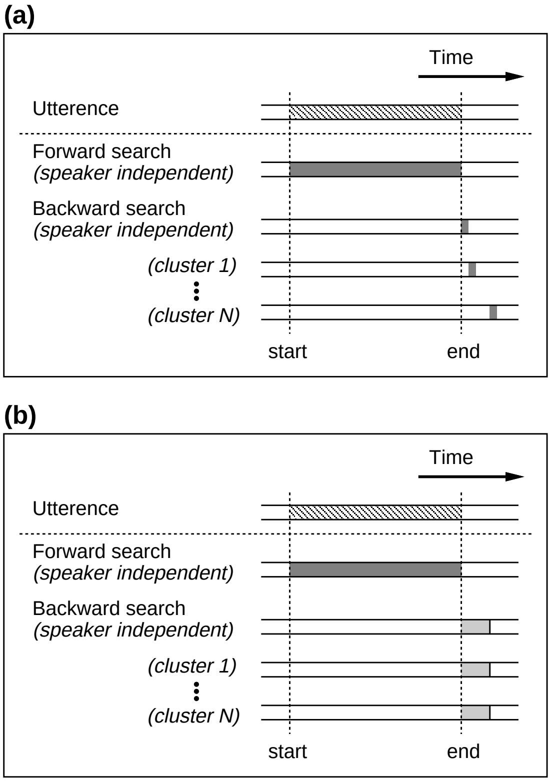

Figure 1. Brief image of the search algorithm

Therefore, we applied the output probability recalculation algorithm[6, 7] to the speaker cluster HMMs.

The algorithm consists of the following three steps: 1) rough and fast output probability estimation, 2) selection of the states which have high estimated values, 3) output probability recalculation on the selected states. Because the rigid probability computation is held only on the selected states, it runs very fast. Here, we used scalar quantization and independent dimension multi-mixture computation for the probability estimation reported previously[7].

When applying this algorithm to the speaker cluster HMMs, the results of the state selection can be shared among every speaker cluster HMM. In other words, the estimation and the selection is carried out only for the root cluster HMM. Only the recalculation for the selected states is carried out for every speaker cluster HMM.

In this way, computation amount for the output probability is effectively reduced.

4. EXPERIMENT

We performed experiments on 1000 word speaker independent continuous telephone speech recognition. The algorithms are evaluated by recognition rate and time spent.

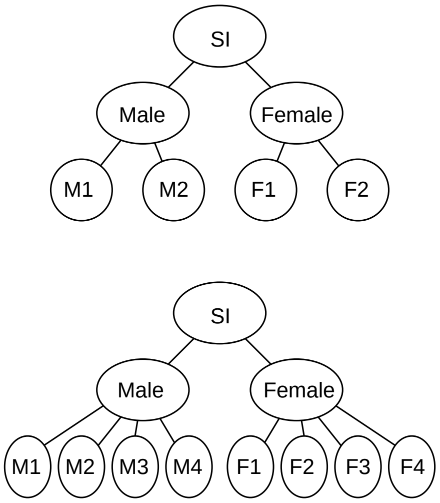

Figure 2. Cluster tree organization

4.1. Conditions

The HMM was shared state triphone consisting of left-to-right 3 states. The number of states was about 780 for each speaker cluster HMM set. Each state has 6 mixture components. The HMM parameters was adapted for telephone speech using MCMS-PMC[9]. The speaker cluster tree were organized into 3 layers: root (speaker independent), gender dependent and leaves. The leaves were designed using k-means clustering for each gender. The number of the leaves was 4 or 8 (Figure 2). From the cluster trees, we picked up some clusters for each experiment. Stack depth for the tree-trellis based search was 10 for forward and 30 for backward. The beam search was also used. The inter-word phoneme models are expanded into triphones in the backward step while monophones are used in the forward step. The other experimental conditions are shown in the Table 1.

4.2. Results and Discussion

At first, we had backward search run independently on each speaker clusters (Figure 1(a)). The results are shown in the Table 2 and the Figure 3. As the baseline, we had run recognition simply in the (*speaker cluster*) × (*linguistic candidate*) space[5].

The Table 2 shows that the proposed algorithm doesn't harm recognition rate in spite of the reduction of the recognition time shown in the Figure 3.

The Figure 3 shows that the proposed algorithm reduces the processing time effectively. The reduction rate amounts about 25% and 30% for 3 and 7 clusters, respectively. However, increase of the time spent for the backward search is also found. This increase seems to be caused by the multiple backward search held independently on every speaker clusters.

Then we had another experiment where backward search ran simultaneously on every speaker clusters using best-first criterion (Figure 1(b)). As shown in Figure 4, the search speed can be improved. However, serious degradation of the recognition rate was found, caused by the discrepancy against the A* search condition.

Table 1. Experimental condition

training	ASJ, ATR and Canon 404 speakers, about 143000 utterances
test	50 sentence × 10 speakers average duration : 2.7 sec/word
grammar	written by BNF, vocabulary: 1004 word perplexity: 30.2

Table 2. Experimental result (rate)

cluster	rate(%) proposed / baseline
root	79.6
root+gender	83.4 / 83.2
root+leaf(4)	83.0 / 82.4
root+gender+leaf(4)	84.4 / 83.6
root+leaf(8)	83.0 / 82.2

In general, speaker independent HMM outputs relatively lower score than some of the other speaker cluster HMMs. Furthermore, we had word boundary context expansion in the backward step which also causes increase of the score. In fact, we can get better recognition rate expanding the word boundary context in advance (Table 3). However, the word boundary expansion increases the search space for the forward step. This trade-off is shown in Figure 5.

Nevertheless, either of the proposed methods showed better performance than baseline in both rate and speed, when seven clusters (root, gender and four leaves) were used.

5. CONCLUSION

In this paper, we proposed a fast algorithm for a speaker independent speech recognizer using speaker cluster HMMs without cluster pre-selection. The algorithm utilizes tree-trellis based search and the output probability recalculation for sharing some parts of the procedures among the speaker clusters. The algorithm is evaluated on 1000 word speaker independent continuous telephone speech recognition. When the 7 clusters are used, 30% of the computation was saved without loss of recognition rate.

ACKNOWLEDGMENT

The authors wish to thank Dr. Hideyuki TAMURA, Head of the Media Technology Laboratory at Canon Inc., Mr. Minoru FUJITA, the manager of the Intelligent Media Division, Mr. Tsuyoshi YAGISAWA, the manager of our department, for giving us the opportunity of this study.

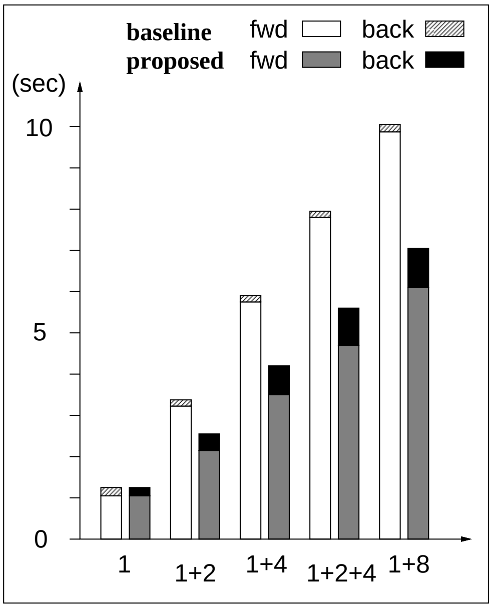

Figure 3. Computation time for forward and backward step in baseline/proposed algorithm

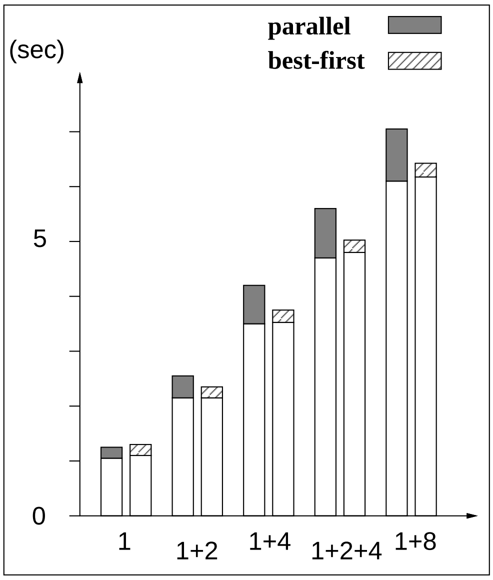

Figure 4. Computation time for parallel/best-first backward search

REFERENCES

[1] Kosaka T. et al. : "Tree-structured speaker clustering for speaker adaptation", Proc. ICASSP94, vol. 1, pp. 245-248 (1994).

[2] Gauvain J.L. et al. : "Developments in large vocabulary dictation : the LIMSI Nov94 NAB system", Proc. of the ARPA Spoken Language Systems Technology Workshop, 1995.

[3] Normandin Y. et al. : "CRIM's November 94 continuous speech recognition system", Proc. of the ARPA Spoken Language Systems Technology Workshop, 1995.

Table 3. Recognition rate when word boundaries are expanded in forward / backward

cluster	rate(%) backward / forward
root	79.6 / 80.6
root+gender	83.4 / 84.6
root+leaf(4)	83.0 / 83.6
root+gender+leaf(4)	84.4 / 84.8
root+leaf(8)	83.0 / 84.0

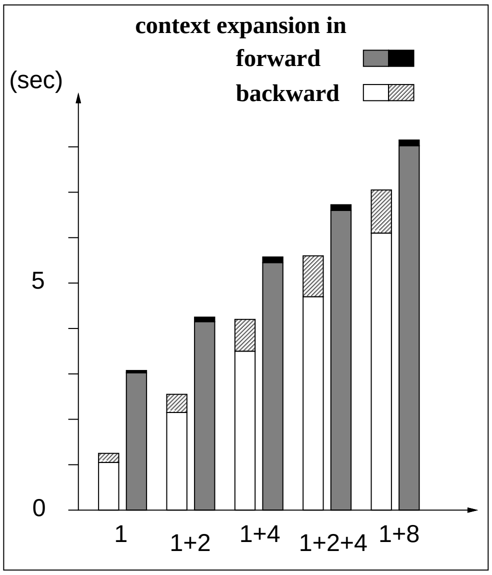

Figure 5. Expand word boundary phoneme context in forward/backward search

[4] Woodland P.C. et al. : "The 1994 HTK large vocabulary speech recognition system", Proc. ICASSP95, vol. 1, pp. 73-76 (1995).

[5] Yamaguchi K. et al. : "Speaker-consistent parsing for speaker-independent continuous speech recognition", Proc. ICSLP94, vol. 2, pp. 791-793 (1994).

[6] Komori Y. et al. : "An efficient output probability computation for continuous HMM using rough and detail models", Proc. Eurospeech95, vol. I, pp. 1087-1090 (1996).

[7] Yamada M. et al. : "Fast output probability computation using scalar quantization and independent dimension multi-mixture", Proc. ICASSP96, vol. II, pp. 893-896 (1996).

[8] Soong F. et al.: "Tree-Trellis Based Fast Search for Finding the N Best Sentence Hypotheses in Continuous Speech Recognition", Proc. ICASSP91, pp.705-708 (1991).

[9] Yamamoto H. et al.: "Fast speech recognition algorithm under noisy environment using modified CMS-PMC and improved IDMM+SQ", Proc. ICASSP97, pp.847-850 (1997).