Proceedings of the 2005 Finnish Signal Processing Symposium - FINSIG'05
August 25, 2005
Kuopio, Finland

Speaker Clustering in Speech Recognition

Olga Grebenskaya, Tomi Kinnunen, Pasi Fränti

University of Joensuu
Department of Computer Science
P.O. Box 111, FIN-80101 Joensuu
FINLAND
{ogreben,tkinnu,franti}@cs.joensuu.fi

ABSTRACT

The paper presents a combination of speaker and speech recognition techniques aiming to improve speech recognition rates. This combination is done by clustering the speaker models created from the training material. Speaker model is a codebook obtained by Vector Quantization (VQ) approach. We propose metaclustering algorithm to group codebooks into clusters and calculate the centroid codebooks. The last are thought as cluster representatives and used to determine the closest cluster on the recognition stage. We present the results of clustering under two conditions. First one keeps codebook size fixed and varies the number of clusters while the second one examines the impact of different cluster number on recognition results while codebook size is fixed.

1. INTRODUCTION

The performance of speech recognition systems has significantly increased since an invention of stochastic models for acoustic modelling, namely *Hidden Markov Models* (HMMs). A wide research is concentrated now on the ways of improvement the recognition results. One of the challenging problems is speaker variability: every speaker has his/her own individuality and pronounces words in his/her own manner. Among these characteristics, we can consider speaking rates, pitch and vocal tract characteristics. To overcome the problem stated above the adaptation ([3], [5]) and clustering ([1], [4], [6], [7]) approaches can be applied.

In clustering, we need to decide what our cluster representative is and how to measure the distance between it and units being clustered. In our case it is obvious that the cluster representative should characterize the corresponding speaker group. There are a number of studies done in speaker clustering and the methods proposed can be divided into two groups: *model-based* and *non-parametric* cluster representations. The examples of the former case are GMMs, HMMs or HMNets based clustering ([4], [7]). The later is represented by speaker grouping using e.g. estimated vocal tract (VT) parameters ([6]). An interesting example of speaker clustering is studied in [1]. It uses the eigenvoice ideas to get speaker specific weight vectors and cluster them in a bottom-up manner. The methods, mentioned above, give 5-11% of relative word error reduction.

In this paper, we concentrate on speaker clustering technique, which makes use of the recent speaker recognition achievements. The VQ based speaker identification provides us with fast speaker identification, so it is possible to use it to improve speech recognition results. Moreover, we can use the distance between speaker codebooks as a measure of the speaker closeness in the acoustic space. Based on it the clustering can be done.

We propose a method of speaker grouping based on speaker models, i.e. codebooks, and will refer to it as *metaclustering* since the objects for clustering are the results of clustering by themselves.

The rest of this paper is organized as follows. Section 2 describes how metaclustering is involved in speech recognition training and testing. The results of experiments done with the proposed algorithm are given in Section 3 and discussed in Section 4. The concluding remarks are presented in the last section.

2. CODEBOOK SPEAKER CLUSTERING

2.1. Training procedure

In order to use clusters in speech recognition we need to obtain them on the training stage. This procedure is outlined in Fig. 1.

The speech material from every speaker is used to extract individual feature vectors (FV sets) and train codebooks on them (using VQ). As a result, we get N speaker codebooks (CBs), or models. These codebooks reflect speaker individualities and the next step is to group them into M ($M<N$) clusters so that similar speakers appear in the same group. This is done by metaclustering algorithm.

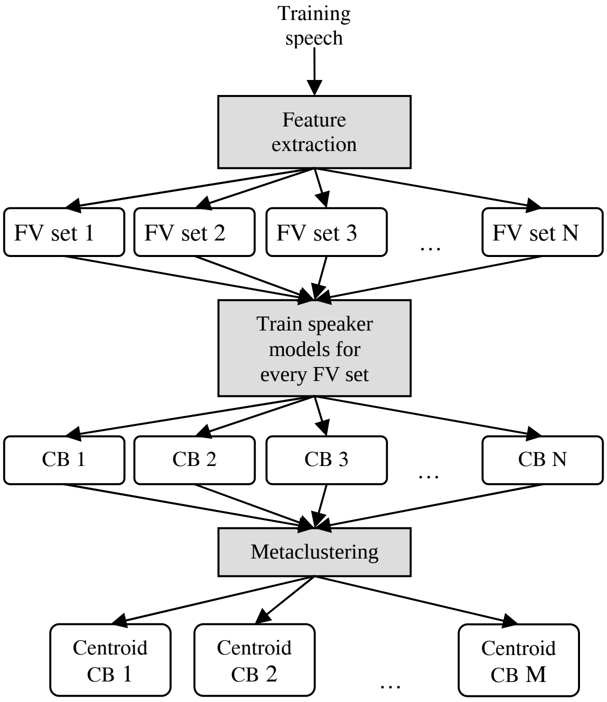

Fig. 1. Metaclustering involved in training process. *N* is a number of speakers in the training set, *M* is a number of clusters

The following symmetric distance measure is used to evaluate the similarity between speakers:

$$D(\mathbf{X},\mathbf{Y}) = dist(\mathbf{X},\mathbf{Y}) + dist(\mathbf{Y},\mathbf{X}), \qquad (1)$$

where ***X***, ***Y*** are codebooks and *dist(**X**,**Y**)* is calculated as a sum of minimum distances between every vector in ***X*** and every vector in ***Y***. It is shown in Fig. 2.

```
function dist($X$,$Y$)
{
    sum := 0;
    FOR every vector $x_i$ in $X$ DO
    {
        d := EuclideanDist($x_i$, $y_1$);
        FOR every vector $y_j$ in $Y$ DO
        {
            IF (d > EuclideanDist($x_i$, $y_j$))
            {
                d := EuclideanDist($x_i$, $y_j$);
            }
        }
        sum := sum + d;
    }
return sum;
}
```

Fig. 2. Calculation of the distance between two codebooks.

The metaclustering itself is illustrated in Fig. 3.

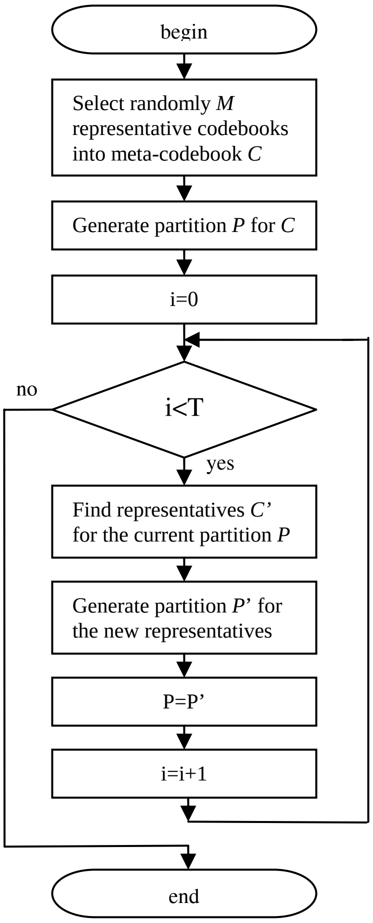
begin
Select randomly M representative codebooks into meta-codebook C
Generate partition P for C
i=0
no
i<T
yes
Find representatives C' for the current partition P
Generate partition P' for the new representatives
P=P'
i=i+1
end

Fig. 3. Metaclustering algorithm flowchart. *T* is the maximum number of iterations.

The first step is initialization. We select randomly *M* codebooks from the whole set and call them 'centroids'. For these centroid codebooks the partition *P* is formed. It is done by assigning every codebook from the set to the closest centroid. The closeness is evaluated as a minimum *D(**X**,**Y**)*, where ***X*** is a codebook from the set and ***Y*** is a centroid. The next step is an iterative centroid recalculation. This is done by pooling all vectors from codebooks belonging to one cluster together and performing any clustering procedure on this merged vector set. We use the *Randomized local search* algorithm [2] for obtaining the centroid codebook from the pool of vectors. The result of the metaclustering is a set of *M* centroid codebooks and we treat these codebooks as cluster representatives.

The final HMMs for speech recognition were obtained applying speaker adaptation techniques to the models trained for all speakers, i.e. cluster independent ones. The data from a cluster was used to adapt those models. Such an approach allows to overcome the problem of a not significant speech data when data from the cluster are not enough to train HMMs well.

2.2. Recognition procedure

The speech recognition procedure goes as follows:

- Feature extraction
- Using extracted feature vectors determine the appropriate cluster.
- Use HMMs belonging to the chosen cluster to perform speech recognition.

The determination of the cluster is simply a speaker identification procedure: the cluster is chosen so that the distance between feature vectors, extracted from the testing speech, and cluster representative (i.e. centroid codebook) is minimal.

The impact of the speaker clustering on speech recognition will be discussed in the next section.

3. EXPERIMENTS

3.1. Test setup

For the experimental part, we built triphones-based speech recognizer using HTK. All the experiments were done on the TIMIT speech database. The development set consists 3696 sentences spoken by 462 speakers. Training of speaker models as well as training of HMMs for speech recognition were done on this speech material. For testing purposes the TIMIT *core set* was used. It includes 192 sentences coming from 24 speakers. The training and testing sets are non-overlapping. MFCC feature vectors (dimensionality 39) with first and second derivatives included were used for both HMMs training and speaker clustering. We also used bigram language model trained from all TIMIT sentences to improve speech recognition results. Speaker modelling and identification were performed by *Sprofiler* speaker recognition software developed in the University of Joensuu. The baseline system *word error rate* (WER) was 6.63%.

3.2. Experimental results

The speaker clustering approach was tested for different number of clusters and codebook sizes. In the first part, we studied the influence of the number of speaker clusters on the recognition accuracy. In the second part, we kept the number of speaker clusters fixed and varied the size of speaker codebooks.

3.2.1. Number of clusters

To study the effect of the number of clusters on recognition rate, we fixed the speaker codebook size to 64. Cluster-dependent set was obtained by performing an adaptation of speaker independent HMMs using data from the cluster. The adaptation was done using MLLR, MAP and MAP+MLLR approaches. The results are given in Table 1.

Table 1. Recognition accuracy for varying number of speaker clusters.

Number of clusters	MLLR	MAP	MLLR+MAP
2	7.9%	6.69%	8.8%
4	6.31%	6.69%	6.5%
8	6.63%	7.07%	6.63%
16	6.82%	6.88%	7.46%
32	7.07%	6.63%	7.1%
64	6.88%	7.27%	8.03%
128	7.84%	7.14%	9.24%
256	7.78%	6.44%	10.71%

For 4 - 32 clusters, the accuracy is close to the baseline results and for 2, 128 and 256 clusters, MLLR and MAP+MLLR are inferior to baseline. For four clusters, MAP reduced WER from 6.63% to 6.31%.

3.2.2. Codebook size

Next, we varied the speaker codebook size from 4 to 256 and kept the number of speaker clusters fixed to 4. Methods for obtaining cluster dependent HMM sets were kept the same as in the previous test. The results are shown in Table 2.

Table 2. Recognition accuracy for varying codebook size (number of speaker clusters = 4).

Speaker codebook size	MLLR	MAP	MLLR+MAP
4	7.01%	6.82%	6.82%
8	6.44%	6.18%	6.63%
16	6.69%	6.69%	6.76%
32	6.44%	6.63%	6.37%
64	6.31%	6.69%	6.5%
128	6.76%	6.44%	6.37%
256	6.37%	6.5%	6.37%

The best result of 6.18% WER was obtained in the case of MAP adaptation and codebook size equal to 8, i.e. it gave 6.8% relative WER reduction compare to the baseline.

4. DISCUSSION

From the first experimental part, we can see that the best result obtained is about 5% relative WER reduction. For the second test setup, this value turned out to be 6.8%. It is quite interesting to analyze the data assigned to each cluster. The division into four clusters in a case of codebook size 8 is shown in Fig. 4.

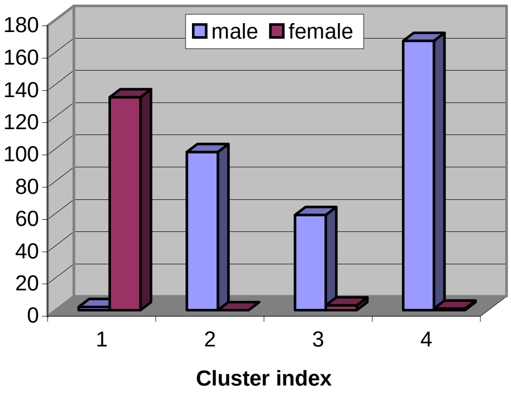

Fig. 4. Male/female division in a case of 4 clusters and codebook size of 8.

In this case, the metaclustering algorithm divides speakers into male/female groups with almost 100% probability. For the case of two clusters, we repeated the metaclustering algorithm ten times. We found out that in six cases it assigned almost all speakers in a single cluster. The smallest mean squared error was obtained in one of these six cases. The clustering approach is based on the suboptimal *k*-means algorithm, which strongly depends on initial solution. It explains why partitions differ from each other between repetitions.

5. CONCLUSIONS

The VQ-based clustering approach presented in the paper can be though as one of the ways of involving speaker information in speech recognition. The best relative WER reduction obtained was 6.8%. We did not observe dependency between the number of clusters and the recognition performance. However, for more than 64 clusters, MLLR showed performance degradation. Small codebooks with the size of four also turned out to give high error rate due to poor speaker representation. Speaker clustering itself can be also considered as a kind of speaker adaptation where HMM parameters are not changed, but “the best” model set is found. VQ speaker representation provides us with the fast speaker identification process and hence allows to adapt speaker recognition system rapidly.

REFERENCES

[1] P. Faltlhauser and G. Ruske, “Robust Speaker Clustering in Eigenspace”. In *Proc. SRU2001*, 57-60, 2001.

[2] P. Fränti and J. Kivijärvi, “Randomized Local Search Algorithm for the Clustering Problem”. *Pattern analysis and Applications*, 3(4), 358-369, 2000.

[3] J.-L. Gauvain and C.-H. Lee, “Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains”. *IEEE Transactions on Speech and Audio Processing*, vol. 2, 291-298, 1994.

[4] T. Kosaka, S. Matsunaga and S. Sagayama, “Speaker-Independent Speech Recognition Based on Tree-Structured Speaker Clustering”. *Computer Speech and Language*, 10(1), 55-74, 1996.

[5] C.J. Leggeter and P.C. Woodland, “Speaker Adaptation of HMMs Using Linear Regression”. Technical Report, Cambridge University Engineering Department, 1994.

[6] M. Naito, L. Deng and Y. Sagisaka, “Speaker Clustering for Speech Recognition Using Vocal-Tract Parameters”. *Speech Communication*, 36(3), 305-315, 2002.

[7] A. Sankar, F. Beaufays and V. Digalakis, “Training Data Clustering for Improved Speech Recognition”. In *Proc. of the European Conference on Speech Communication and Technology*, vol. 1, no. 2, pp. 503-506, 1995.