Designing a SC-HMM System as a Tool for Future Research in Multi-stream Continuous Speech Recognition

Vitor Pera, CEFAT-FEUP

ABSTRACT

This paper describes work that was done during a 9 months stay at ELIS (Gent). The objective was to develop in-house speech recognition software tools that could be used as a starting point for future research in multi-stream continuous speech recognition (CSR). It was decided to go for a baseline system based on context-independent SC-HMMs, and this paper describes how the system was developed, what its actual performance is, and how the system could evolve in the future.

INTRODUCTION

The basic idea was to try and build a system that was simple in concept, requiring not too much code, and that is yet able to attain an acceptable performance on tasks like Resource management (RM) or Wall Street Journal (WSJ). This way, we would have full control over our software, so that we can adapt it to our needs when performing multi-stream research later. Moreover we would have a good baseline system for comparison of multi-stream over single-stream recognition at an appropriate level of performance. The software was always tested on RM, and at the end of my stay in Gent, a word error rate of 6,72% was achieved on the Feb89 test set. Since my return at CEFAT, I have further improved the training strategy, and the expectations are that using this new strategy, a final performance between 5 and 6% is within reach.

The paper chronologically describes the evolution of the system. Section 1 outlines its status at the time of my arrival in Gent, and describes the first experiments that were carried out on RM at ELIS. Section 2 discusses the significant improvements that were introduced during my stay in Gent. Section 3 finally deals with the improvements that were reached since then, and which were the result of a continuing collaboration with ELIS. The paper ends with a discussion on future work and a formulation of conclusions.

1. THE INITIAL SYSTEM

The first software modules were developed in CEFAT. Using this software it was possible to perform the following operations:

- mel-scale cepstral analysis of the speech signal;
- generation of a codebook of diagonal covariance gaussians to cover the acoustic space;
- Baum-Welsh training of the transition an emission probabilities of a SC-HMM acoustical models;
- Viterbi phone recognition on the basis of a unigram phone language model.

That code was tested on the TIMIT database. In order to move from phone recognition to word recognition, the following tasks had to be carried out:

- make the interface modules compatible with the ELIS data formats, and augment them to support new databases (e.g. RM) with other labelling conventions;
- change the procedures for generating the codebook, so that it is no longer a collection of sub-codebooks derived for individual phones;
- introduce a language model, which in the case of RM was the standard word-pair grammar that is supplied with the speech data on the CDs.

After all these modifications were carried out, the system was trained and tested on the RM database. In an attempt to reduce the training time, the first tests were run with a small configuration comprising 512 diagonal covariance gaussians, 51 context-independent three-state left-to-right phone models without jumps, and 64 mixtures per state. The models were seeded and 5 Baum-Welsh training iterations were performed. As could be expected, the performance was very poor. The word error rate for the Feb89 test set was no less than 21,80%.

2. SYSTEM DEVELOPMENT AT ELIS

Given the very poor performance achieved with this initial system, it was clear that drastic improvements had to be introduced. The most important changes are listed below.

- The training module that was originally developed at CEFAT did only update the transition probabilities and the mixture weights on the states. It did not adapt the codebook vectors. Consequently, the first thing that was done was to modify the code so that a maximum likelihood adaptation of *all* the free model parameters could be achieved. It was clear from the first tests that this modification had a very significant impact on the system performance.
- In the original software the codebook was initialized by clustering the speech frames of a single mono-phone, and by creating a limited size codebook for each mono-phone. The latter were then collected in one big codebook. Obviously, this is a sub-optimal procedure and a new initialisation procedure based on k-means clustering of the entire data set (using a binary tree splitting strategy) was implemented.
- A garbage model was introduced, obligatory at the sentences extremes and optionally between consecutive words;
- To reduce the computational load and the memory requirements a little bit, the Baum-Welsh paradigm was substituted by a Viterbi training;
- Two other techniques for reducing the computational load were implemented as well, namely, beam-searching (both during training and recognition), and a mechanism for saving emission likelihoods;

Preliminary experiments were carried out in order to fix some of the parameters and to make decisions with regard to the training strategy.

2.2. Fixing the design parameters

In the acoustic front-end, the signal is first pre-emphasised with a factor 0,95. Then it is segmented into overlapping 25ms frames. The frame rate is 10ms. For each frame, 12 MFCCs and a log-energy term are extracted. First- and second-order derivatives are then added so as to obtain a 39-dimensional feature vector per frame.

The 51 mono-phones were modelled by a three-state left-to-right HMM. A *silence* phone was added to the model inventory, and modelled by a one-state HMM.

2.3. The training procedure

The following algorithm was adopted for the training of the system:

Initialisation

generate a codebook of 1K gaussians

seed the models (use 1K mixtures/state)

Training

while the codebook size is less than 8K

run Viterbi iterations as long as the error rate keeps decreasing

perform one or both of the following operations

- halve the number of gaussians used by each state
- double the number of gaussians and the number of mixtures per state

2.4. Experimental results

The systems that were tested are referred to by two numbers: the number of mixtures per state and the number of gaussians in the codebook.

The initialisation yielded a 1K/1K system with a word error rate of 21,78%. After 2 iterations the error rate reached a minimum of 12.7%. Reducing the number of mixtures reduced the error rate to 9.66%. Increasing the codebook size to 2K then resulted in a 512/2K system that was trained until an error rate of 9.3%. Doubling the codebook again resulted in a 512/4K system and an error rate of 8.13%. The evolution of the word error rate as a function of the iteration number is depicted on Figure 1.

Further doubling the codebook size to 8K caused an even more significant improvement. On Figure 2, one observes the error rates obtained with 3 systems differing in the number of mixtures per state. The two 256/8K systems only have a

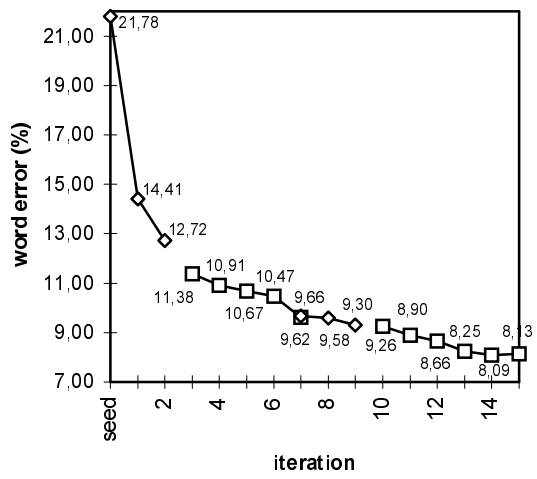

Figure 1 - Word error curves (systems are given by number of mixtures per state / codebook size): (diamonds) 1K/1K; (squares) 512/1K; (diamonds) 256/2K; (squares) 512/4K

different number of mixtures on the garbage model. The best recognition rate obtained after 20 iterations was equal to 6,72%.

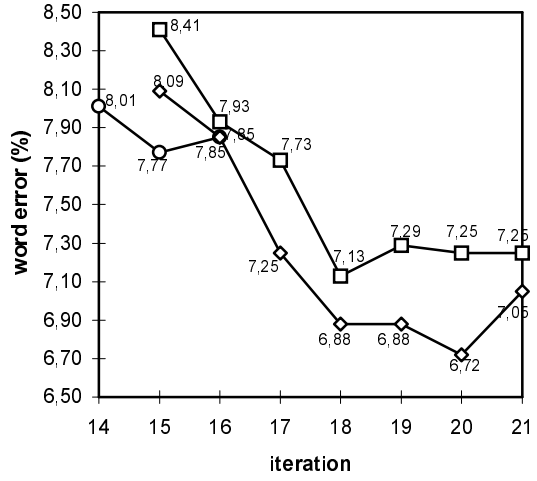

Figure 2 - Word error curves: (squares) 512/8K; (circles) 256/8K; (diamoonds) 256/8K but with a 1K mixtures garbage model

Note that all word error rates were obtained without any optimisation of the word penalty (it was 0) nor the language model exponent (it was 1). So further reductions of the error rate should be possible.

3. DEVELOPMENTS AT CEFAT

Although the training procedure outlined in section 2 already yielded an acceptable performance, it was decided to try and optimise it even further in an attempt to find the best balance between accuracy / number of free parameters / training time. Therefore we adopted a strategy which also enables the creation of models comprising different numbers of mixtures per state. The strategy can be described as follows:

Initialisation

generate a codebook of 512 gaussians
seed the models (use 512 mixtures/state)

Training

while the codebook size is smaller than 8K

perform Viterbi iterations as long as the error rate keeps decreasing

for each model state, eliminate the mixture weights with value smaller than a fraction the largest weight for that state

perform a few Viterbi iterations with the reduced models

split the gaussians that occur with a minimum weight in at least one mixture

As can be derived from Figure 3, the performance for a given system size is now significantly better than before. E.g., the word error rate of 7,53% (at iteration 23) compares to the 9,30% that was obtained with a 256/2K system using the old training procedure.

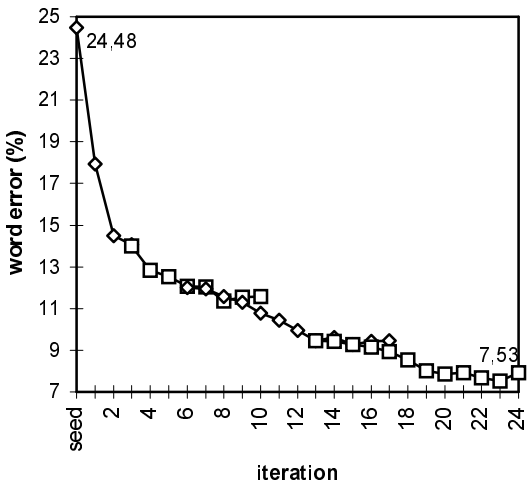

Figure 3 - Word error curves : (diamonds) 512/512; (squares) 256/512; (diamonds) 232/1K; (squares) 318/2K

It is also interesting to examine the number of mixtures that is retained on the different states. From Figure 4 it appears that this number can range from smaller than 40 for very focussed states to more than 600 for very broad states. This clearly shows that a uniform number of mixtures per state is definitely sub-optimal.

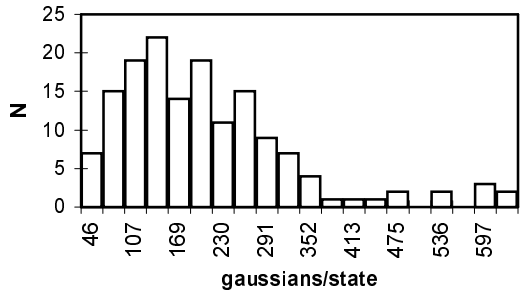

Figure 4 - Histogram of the number of gaussians per state for the 2K-size codebook

The training will be further continued with a system comprising 4K gaussians, and the hope is legitimate that this system will already yield an error rate of less than 6%, which is considered a very acceptable performance for a baseline system.

4. FUTURE WORK

Now that an acceptable baseline system is available, the time has come to start with the real work on multi-stream recognition. By adopting the parallel expert principle, we hope to increase the recognition accuracy.

The objective is to extract multiple information streams from the speech signal, and to investigate different ways of recombination of model scores at certain points in the recognition process. Thanks to the collaboration with ELIS I can dispose of the output stream generated by the ELIS auditory model [Van Immerseel & Martens, 1992] as one of the input streams. Two main approaches are planned for the immediate future.

One approach is to recombine the information contained in word-graphs produced by separate recognisers working on the different information streams. The necessary software for implementing a word-graph generator [Ney, 1994] has already been integrated in my system.

A second approach is to process the input streams independently up to a pre-defined sub-word level, and to recombine the model scores at that level. This approach requires to develop additional software for implementing HMM recombination [Bourlard & Dupont, 1996].

5. CONCLUSIONS

During my stay at ELIS (Gent) I was able to develop a SC-HMM system for continuous speech recognition with an acceptable accuracy (6.7% word error rate on RM). This system was further improved since then, and it is now good enough to serve as a baseline system for the benchmarking of future multi-stream recognition systems. Since my return at CEFAT, I already implemented a method for generating word-graphs, and I am currently working on software for the implementation of recombination of word-graphs. Finally, I started to program a new search engine supporting HMM recombination.

ACKNOWLEDGEMENTS

I would like to acknowledge prof. Jean-Pierre Martens and his group for hosting me at ELIS and for spending so much time with me discussing the options to take during the development of my system. I also thank the action COST249 *Continuous Speech Recognition over the Telephone* for having created the favourable circumstances for initiating this collaboration between ELIS and CEFAT.

REFERENCES

H. Bourlard, S. Dupont (1996). “A new ASR approach based on independent processing and recombination of partial frequency bands,” Procs. ICSLP-96, 426-429.

H. Ney, X. Aubert (1994), “A word graph algorithm for large vocabulary, continuous speech recognition,” Proc. ICSLP-94, 1355-1358.

L. Van Immerseel, J.P. Martens (1992). “Pitch and voiced/unvoiced determination with an auditory model,” J. Acoust. Soc. Am. 91, 3511-3526.