# Text independent speaker verification using string codebooks

*Filipe Moreira, Carlos Espain*
*CEFAT / DEEC / FEUP / Universidade do Porto*

**ABSTRACT**

This paper reports on experiments on text independent speaker verification using vector quantisation. Several situations were considered using different features sets, training times and testing utterances durations, both with speaker independent and speaker dependent thresholds. The amount and location of silence in the testing sentences is usually a problem in text independent speaker verification systems. We proposed and tested the use of string codebooks to attenuate the problem. A codebook is extracted from the utterance and it is this codebook that is quantised again into the claimer's codebook.

## INTRODUCTION

Many systems of text independent speaker verification are well known [Matsui 92]. VQ methods [Deller 93] have been widely used. However they do not deal efficiently with the problem of the silence/noise which might be present, to a variable extension and localisation, when the claimer is free to utter whatever he wants, as opposed to when he/she is asked to utter a password. A speech detector may be used to cut the silence off.

We made an experiment where the expected size of the testing utterances were known, they were strings with the same number of digits. We tried then three approaches. In the first two we did a quantisation of the string frames into the claimer's codebook, not normalising and normalising the total quantisation error to number of frames. We then tried a third method directly intended to address the problem of the silence by generating first a codebook out of each utterance and then quantising this codebook into the claimer's. This way silence might be reduced to a few points of the string codebook whatever its duration might be.

## TRAINING CONDITIONS

The experiments were done using a subset of the TIDIGITS database, consisting of 110 speakers randomly chosen, each one uttering 77 strings of digits, of which only 44 were used, 22 for training the codebooks and 22 for testing.

The speech signal was downsampled to 20 kHz, pre-emphasised with a 0.97 coefficient filter and Hamming windowed into 20 ms frames starting every 9 ms. From each frame several features were extracted: energy and $\Delta$-energy; 8 linear predictive coefficients (LPCs) via the Levinson-Durbin algorithm and their first-derivatives; 8 cesptral coefficients (CCs) also with their first-derivatives. Silence was not removed from strings used both in training and testing.

A common LBG algorithm defined our four types of codebooks. The first type (Cbk 1) was generated from 6 3-digitis strings from each speaker to be verified, the second (Cbk 2) from 6 5-digits strings, the third (Cbk 3) from 11 3-digits and the fourth (Cbk 4) from 11 5-digits strings. Average training time was 9.6 s for Cbk 1, 13.8 s for Cbk 2, 17.6 s for Cbk 3 and 25.2 s for Cbk 4.

Within each type, the 4 kinds of codebooks were generated according to the features chosen as components of the vector representing each frame: CCs plus $\Delta$ CCs, CCs, $\Delta$ CCs, LPCs plus $\Delta$ LPCs, LPCs, $\Delta$ LPCs.

## TESTING CONDITIONS

Two subsets of the database were used for testing conditions. The first one consisted of 11 2-digits strings from the claimed speaker and one 2-digits string from each of the 109 impostors. The second one consisted of 11 7-digits strings from the claimed speaker and one 7-digits string from each of the 109 impostors. Each 2-digits string has an average duration of 1.3 s and each 7-digits string has an average duration of 3.2 s. All speakers were in turn considered as claimers. This gives a total of 1210 sessions for testing the false rejection rate (FR) and 11990 sessions for testing the false acceptance rate (FA).

We have therefore used four training situations differing on the amount of training time, six sets of features and two testing subsets corresponding to two types of digits strings, a short one (1.3 s) and a long one (3.2 s).

For each test subset, three different methods were tried.
The first one consisted of a quantisation of the frames into the claimer's codebook, taking the Euclidean distance of frame as the quantisation error and summing up those errors for the entire utterance.
This sum was then compared to a threshold: the claimer speaker would be rejected if the sum was higher then the threshold and accepted if otherwise.
The second method was the same as the above but here the sum was divided by the total number of frames of the utterance.
The third method consisted in generating a string codebook from the spoken string itself. This codebook was then quantised into the claimer's codebook. The total quantisation error was again compared to a threshold.

In all three methods described above two kinds of thresholds were used: the first kind consisted of a single threshold for all the speakers, i. e., a speaker independent threshold. The second kind of threshold is a speaker dependent threshold, i. e., a threshold for each speaker.

Equal error rate (EER) was used as the assessment parameter in all situations.

## RESULTS

For the subset consisting of the 2-digits strings, the results were the following:

| | Cbk 1 | Cbk 2 | Cbk 3 | Cbk 4 |
|---|---|---|---|---|
| CCs + Δ CCs | 23,9 | 23,5 | 23,1 | 22,7 |
| CCs | 21,6 | 20,9 | 21,3 | 19,8 |
| Δ CCs | 40,8 | 40,2 | 40,2 | 39,9 |
| LPCs + Δ LPCs | 27,6 | 27,1 | 27,3 | 26,7 |
| LPCs | 22,9 | 22,8 | 22,5 | 22,7 |
| Δ LPCs | 38,4 | 37,2 | 37,9 | 36,9 |

**Table 1.** EER for the 2-digits testing strings using a speaker independent threshold for the first method

| | Cbk 1 | Cbk 2 | Cbk 3 | Cbk 4 |
|---|---|---|---|---|
| CCs + Δ CCs | 21,7 | 21,1 | 20,4 | 20,2 |
| CCs | 19,0 | 18,4 | 17,4 | 17,0 |
| Δ CCs | 40,7 | 40,6 | 39,8 | 39,9 |
| LPCs + Δ LPCs | 25,5 | 24,5 | 24,7 | 24,3 |
| LPCs | 21,0 | 20,6 | 20,6 | 20,4 |
| Δ LPCs | 37,0 | 36,8 | 36,1 | 36,0 |

**Table 2.** EER for the 2-digits testing strings using a speaker independent threshold for the second method

| | Cbk 1 | Cbk 2 | Cbk 3 | Cbk 4 |
|---|---|---|---|---|
| CCs + Δ CCs | 19,8 | 18,8 | 17,4 | 18,1 |
| CCs | 20,1 | 19,6 | 18,3 | 18,2 |
| Δ CCs | 37,5 | 36,2 | 35,7 | 36,4 |
| LPCs + Δ LPCs | 39,0 | 40,7 | 39,6 | 40,5 |
| LPCs | 40,1 | 41,3 | 41,1 | 41,0 |
| Δ LPCs | 44,2 | 44,9 | 44,1 | 45,1 |

**Table 3.** EER for the 2-digits testing strings using a speaker independent threshold for the third method

| | Cbk 1 | Cbk 2 | Cbk 3 | Cbk 4 |
|---|---|---|---|---|
| CCs + Δ CCs | 20,4 | 19,8 | 19,1 | 18,1 |
| CCs | 18,3 | 16,9 | 16,0 | 15,3 |
| Δ CCs | 39,7 | 39,1 | 39,2 | 38,7 |
| LPCs + Δ LPCs | 24,4 | 23,4 | 23,7 | 22,9 |
| LPCs | 20,1 | 19,1 | 19,4 | 19,0 |
| Δ LPCs | 35,2 | 35,3 | 34,2 | 34,7 |

**Table 4.** EER for the 2-digits testing strings using a speaker dependent threshold for the first method

| | Cbk 1 | Cbk 2 | Cbk 3 | Cbk 4 |
|---|---|---|---|---|
| CCs + Δ CCs | 23,9 | 22,9 | 22,5 | 22,2 |
| CCs | 20,8 | 19,8 | 19,1 | 18,6 |
| Δ CCs | 40,6 | 40,2 | 39,9 | 39,4 |
| LPCs + Δ LPCs | 25,5 | 25,7 | 25,5 | 25,2 |
| LPCs | 21,6 | 20,7 | 20,7 | 20,0 |
| Δ LPCs | 37,2 | 36,7 | 36,9 | 36,3 |

**Table 5.** EER for the 2-digits testing strings using a speaker dependent threshold for the second method

| | Cbk 1 | Cbk 2 | Cbk 3 | Cbk 4 |
|---|---|---|---|---|
| CCs + Δ CCs | 14,1 | 14,5 | 13,3 | 13,8 |
| CCs | 15,3 | 15,9 | 14,7 | 14,9 |
| Δ CCs | 31,1 | 31,5 | 30,5 | 31,1 |
| LPCs + Δ LPCs | 35,5 | 37,6 | 36,7 | 38,0 |
| LPCs | 37,6 | 39,2 | 38,7 | 39,9 |
| Δ LPCs | 41,7 | 42,6 | 42,7 | 43,7 |

**Table 6.** EER for the 2-digits testing strings using a speaker dependent threshold for the third method

Next are the results for the subset test consisting of the 7-digits strings.

| | Cbk 1 | Cbk 2 | Cbk 3 | Cbk 4 |
|---|---|---|---|---|
| CCs + Δ CCs | 21,6 | 20,5 | 20,6 | 19,6 |
| CCs | 18,5 | 17,2 | 17,4 | 16,0 |
| Δ CCs | 39,9 | 38,6 | 38,5 | 37,8 |
| LPCs + Δ LPCs | 26,6 | 25,4 | 25,6 | 24,6 |
| LPCs | 21,3 | 20,2 | 20,7 | 19,5 |
| Δ LPCs | 36,0 | 34,9 | 35,7 | 34,5 |

**Table 7.** EER for the 7-digits testing strings using a speaker independent threshold for the first method

| | Cbk 1 | Cbk 2 | Cbk 3 | Cbk 4 |
|---|---|---|---|---|
| CCs + Δ CCs | 20,5 | 18,5 | 19,1 | 17,5 |
| CCs | 16,9 | 15,2 | 15,6 | 13,7 |
| Δ CCs | 39,9 | 38,7 | 38,8 | 37,9 |
| LPCs + Δ LPCs | 25,1 | 24,2 | 24,5 | 23,5 |
| LPCs | 19,2 | 18,3 | 18,3 | 17,0 |
| Δ LPCs | 37,0 | 35,3 | 35,5 | 34,6 |

**Table 8.** EER for the 7-digits testing strings using a speaker independent threshold for the second method

| | Cbk 1 | Cbk 2 | Cbk 3 | Cbk 4 |
|---|---|---|---|---|
| CCs + Δ CCs | 8,0 | 6,4 | 5,9 | 5,5 |
| CCs | 8,7 | 6,8 | 6,2 | 5,8 |
| Δ CCs | 24,8 | 21,3 | 22,8 | 21,2 |
| LPCs + Δ LPCs | 24,2 | 23,4 | 23,7 | 23,3 |
| LPCs | 22,6 | 21,6 | 22,7 | 21,5 |
| Δ LPCs | 34,3 | 30,9 | 31,6 | 31,4 |

**Table 9.** EER for the 7-digits testing strings using a speaker independent threshold for the third method

| | Cbk 1 | Cbk 2 | Cbk 3 | Cbk 4 |
|---|---|---|---|---|
| CCs + Δ CCs | 19,3 | 18,3 | 18,4 | 17,2 |
| CCs | 16,3 | 15,1 | 15,2 | 13,9 |
| Δ CCs | 39,0 | 37,9 | 37,5 | 37,0 |
| LPCs + Δ LPCs | 23,6 | 23,1 | 23,2 | 22,4 |
| LPCs | 19,1 | 17,9 | 18,3 | 16,6 |
| Δ LPCs | 35,0 | 33,8 | 34,5 | 33,3 |

**Table 10.** EER for the 7-digits testing strings using a speaker dependent threshold for the first method

| | Cbk 1 | Cbk 2 | Cbk 3 | Cbk 4 |
|---|---|---|---|---|
| CCs + Δ CCs | 17,9 | 16,8 | 16,5 | 15,4 |
| CCs | 14,7 | 13,4 | 13,3 | 11,8 |
| Δ CCs | 38,9 | 37,3 | 37,9 | 36,2 |
| LPCs + Δ LPCs | 23,2 | 22,3 | 22,0 | 21,2 |
| LPCs | 18,0 | 16,8 | 17,1 | 15,5 |
| Δ LPCs | 34,8 | 33,9 | 34,3 | 33,0 |

**Table 11.** EER for the 7-digits testing strings using a speaker dependent threshold for the second method

| | Cbk 1 | Cbk 2 | Cbk 3 | Cbk 4 |
|---|---|---|---|---|
| CCs + Δ CCs | 4,5 | 3,3 | 3,2 | 2,6 |
| CCs | 5,1 | 4,6 | 3,8 | 3,6 |
| Δ CCs | 14,6 | 12,6 | 13,1 | 12,0 |
| LPCs + Δ LPCs | 15,2 | 14,8 | 14,8 | 19,5 |
| LPCs | 16,2 | 15,8 | 16,5 | 16,6 |
| Δ LPCs | 18,6 | 18,4 | 18,5 | 18,3 |

**Table 12.** EER for the 7-digits testing strings using a speaker dependent threshold for the third method

## CONCLUSIONS

As expected, in all the three methods, results are better when the threshold is speaker dependent, tuned individually for each speaker.
Testing with 7-digits strings proves better as also expected.

The second method performed better than the first one because the distances are normalised and this takes account for the duration variability of the pronounciation.

However it does not entirely solve the problem and it seems that most of all it does not deal properly with different duration of silence that can be before, in between, and after the strings.

The third method tried to address this problem by generating a codebook for each testing string and then a quantisation of this codebook into the claimer's codebook. Results show much lower equal error rates except for codebooks using linear prediction coefficients for 2-digits testing strings. Apparently when using LPCs these strings are too short for generating proper string codebooks.

In the first and second method cepstral coefficients codebooks worked better than all the others, including those generated from CCs plus $\Delta$CCs. However in the third method, using string codebooks, best performance was achieved with CCs plus $\Delta$CCs. First derivative information becomes relevant to speaker verification.

## REFERENCES

[Deller 93] - Deller, J. R., Proakis J. G., Hansen J. H. L., "Discrete-time Processing of Speech Signals", Macmillan. 1993.

[Matsui 92] - Matsui, T., Furui, S., *Comparison of Text-Independent Speaker Recognition Methods Using VQ-Distortion and Discrete/Continuous HMMs*, Proc. IEEE Int. Conf. ASSP, San Francisco, 1992.