Noise Reduction for Noise Robust Feature Extraction for Distributed Speech Recognition

Bernhard Noé[1], Jürgen Sienel[1], Denis Jouvet[2], Laurent Mauuary[2], Louis Boves[3], Johan de Veth[3], Febe de Wet[3]

[1]Alcatel SEL AG, Stuttgart, Germany
[2]France Télécom R&D, DIH/IPS, Lannion, France
[3]A[2]RT, Dept. of Language & Speech, University of Nijmegen, The Netherlands

{Bernhard.Noe,Juergen.Sienel}@alcatel.de, {mauuary,jouvet}@francetelecom.com,
{j.deveth,f.dewet,l.boves}@let.kun.nl

Abstract

This paper describes the noise robust feature extraction methods developed by France Telecom and Alcatel for the noise robust front-end standardisation of ETSI Aurora. It is shown that both noise reduction methods give a substantial improvement when compared to a standard MFCC feature extraction algorithm for speech recognition in noisy environments. In addition, blind equalisation and feature vector selection were used for further improvement of recognition performance. Results are discussed for the ETSI Aurora 2 task and the SDC-Italian task as well. It was found that the combination of noise reduction with the proposed methods is capable to achieve around 50% reduction of the error rate. In the context of the open ETSI Aurora standardisation, two proposals were submitted based on these methods, they achieved the best results among all the proposals.

1. Introduction

Information services in mobile networks suffer from a lack of interaction mechanisms that allow for user friendly input. The aim of making mobile terminals smaller makes the handling of such services even more difficult, because only a small number of keys is available to input the information required by the service. Thus speech recognition may offer a crucial benefit. However, speech recognition over mobile networks suffers from several drawbacks: the voice signal is coded with different coding schemes, the presence of noise in the mobile environment and the degradation of the speech signal, due to errors on the radio link between terminal and base station. At the same time mobile phones offer powerful signal processing. Against this background, ETSI has launched the AURORA project, to introduce distributed speech recognition (DSR), aiming at a standardised front-end algorithm that enhances recognition performance in noisy environments.

Besides the improvement of the recognition rate other criteria have been defined (such as the computational performance, the delay introduced by front-end computations and the feature vector size) to allow processing of the front-end within standard mobile terminals and of the acoustic decoding by the back-end recogniser with requirements similar to that of current technologies.

Moreover, the proposals should prove that they are independent from the acoustic input channel (handsfree or close talk microphone). Therefore test situation have been defined, where the recogniser is trained with close-talk speech data, while the testing is done with data recorded via the hands-free microphone. A blind equalisation module [1] eliminates the effects of differences between microphones and channels.

Finally transmission over the mobile network is costly. Thus, the integration of a voice activity detection (VAD) in the front-end, to avoid the transmission of silence frames -at least at the beginning of an utterance- should reduce network traffic. A VAD is also helpful for removing non-speech frames, which are often harmful for the recogniser in mismatched conditions.

The organisation of the paper is as follows. Section 2 and 3 respectively describe the two noise reduction approaches that were developed, and their integration in the feature extraction process. Section 4 recalls the blind equalisation module, and section 5 summarises some computations currently done in the backend. Finally section 6 presents the experiments, details the obtained results and discusses them.

2. Time domain noise reduction and feature extraction

2.1. Overview

Time domain noise reduction (TDNR) is applied prior to the Mel frequency cepstral computations, as indicated in Figure 1. The speech signal after noise reduction is also used for computing the logarithm of the energy parameter. Blind equalisation is also part of the front-end and is described in section 4.

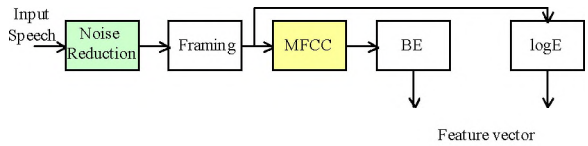

Figure 1: Block diagram of the time domain noise reduction based front-end

2.2. Noise reduction

The noise reduction module operates in time domain. Its architecture is shown in Figure 2. After offset compensation (Offcom block) several types of processing take place. First,

the spectrum is computed (Analysis block). A VAD module classifies frames as speech or non-speech (noise) by comparing the SNR to a threshold. The SNR corresponds to the difference between the short-term and long-term signal log-energy estimates. The long-term estimate is updated when the VAD decides that the current frame corresponds to non-speech and the energy of the current frame is used as the short-term estimate. In addition, a hangover of 50 ms is applied after any speech to non-speech transition. The hangover is only applied if the duration of the speech segment immediately before the transition is greater than 50 ms. This effectively avoids that the hangover is applied after very short noise segments, that may be misclassified as speech.

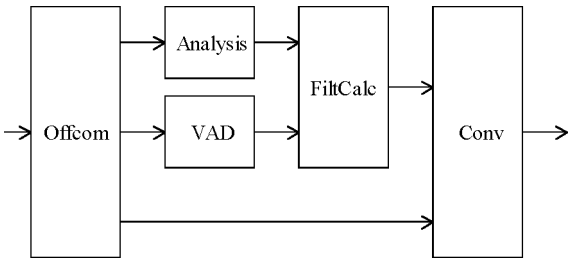

Figure 2: Block diagram of time domain noise reduction module.

The output of the VAD is used to decide if the noise spectrum estimate has to be updated. An improved SNR estimate is then obtained from a 2-step SNR estimation technique: the noise spectrum is used to compute a first estimate of the noiseless signal spectrum using a "decision-directed" approach. Then the noiseless signal spectrum is used to compute a priori signal to noise ratio (SNR) in the different frequency bands. A filter transfer function is computed from these SNR values, which is used to refine the estimate of the noiseless signal spectrum by applying in the frequency domain the filter transfer function on the noisy signal spectrum. Using this improved noiseless signal estimate, new SNRs in the frequency bands as well as an improved filter transfer function are computed.

The impulse response of the filter transfer function is then computed using an inverse Fourier transform. This impulse response is truncated to a length of 17 and a Hanning window is applied to the truncated impulse response. Truncation and windowing of the impulse response results in a smooth filter, that is highly beneficial for speech recognition performance. The noise-reduced signal is finally obtained by convolving the noisy input signal with the filter impulse response.

The noise-reduced signal is then used for computing the Mel-cepstrum coefficients (MFCC), using 20 ms signal windows, and a frame shift of 10 ms. 12 coefficients are computed per frame, plus the logarithm of the energy.

3. Frequency domain noise reduction and feature extraction

3.1. Overview

The proposed approach combines a standard MFCC feature extraction algorithm with a noise reduction scheme and an additional silence frame processing method. As in the previous front-end, the cepstral output data are normalised using a blind equalization algorithm. Figure 3 shows all processing blocks of the algorithm.

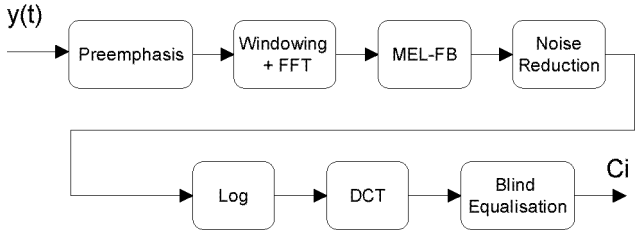

Figure 3: Block diagram of the frequency domain noise reduction based front-end

The algorithm uses a frame length of 20 ms with 10 ms overlap. A pre-emphasis is performed on speech data and a Hamming window is applied, before a 256-point FFT is performed. On this data the power spectral densities are transformed to the Mel-filterbank outputs using 30 coefficients These coefficients provide the input for the frequency domain noise reduction algorithm (FDNR).

3.2. Noise reduction

The FDNR algorithm uses a modified maximum likelihood estimation [2], where the gain is calculated dependent on the estimated signal to noise ratio $\hat{\rho}_k(t)$ for each band, whereas signal means here speech plus noise:

$$g_k(t) = \frac{1}{2}\left[1 + \sqrt{\frac{\hat{\rho}_k(t) - 1}{\hat{\rho}_k(t)}}\right] \cdot P(H_1) + G_{\min} \cdot [1 - P(H_1)]$$

G_{min} defines the minimum gain value to avoid spectral distortions, that may result from occasional incorrect SNR estimation. $P(H_1)$ denotes the a posteriori probability of the hypothesis that a given frame is a noisy speech frame; it is derived from a priori SNR values $\eta_k(t)$ and the estimated SNR for the actual frame [2].

$$P(H_1 \,|\, |X_k(t)|) = \frac{e^{-\eta_k(t)} \cdot I_0\left(2\sqrt{\eta_k(t) \cdot \hat{\rho}_k(t)}\right)}{1 + e^{-\eta_k(t)} \cdot I_0\left(2\sqrt{\eta_k(t) \cdot \hat{\rho}_k(t)}\right)}$$

I_o denotes the Bessel function of first order. The noise energy estimate is updated permanently without using VAD information. Fixed values for the a priori SNR are used. For further improvement of recognition rate several tests were carried out with filtering the gain values in frequency and time domain, before applying it to the speech data. It can be expected that smoothing will reduce recognition errors due to spectral distortions caused by badly estimated SNR values. Smoothing in the frequency domain turned out to be very helpful for reduction of error rate, which can be explained by reduction of the distortion in static cepstral coefficients. Smoothing in time domain helped less, but still gave an improvement, which can be explained by reduction of distortion in the dynamic cepstral coefficients, such as velocity and acceleration parameters. For the frequency domain filtering a 9th order FIR filter was used, whereas the time domain filter is a 1th order IIR Filter.

3.3. Additional attenuation for silence periods

As the SDC databases contain a high percentage of noisy non-speech signal portions, the number of insertion errorswas in-

creased dramatically. While for Aurora 2 with a small part of non-speech portions the percentage of insertion errors from the total error count is 10%, for the SDC databases this value is increased to about 50% . Thus, additional effort has to be spent in order to decrease the mismatch between trained word models and high noise non-speech signal portions. Especially for the log-energy parameter the mismatch is very large if training is done with clean, and testing with noisy speech. The solution we chose is to calculate an additional gain only during non speech periods and appliedit to C0. For doing this a voice activity flag, based on the estimated SNR in three subbands is generated. The subbands are created by splitting the Mel-domain into three parts with equal Mel-frequency range. Then the SNRs of each sub-band are compared to fixed thresholds and the VAD flag is set if one of the three subbands SNR is above the threshold.

$$VAD = \begin{cases} 1 \text{ if } SNR_i > thresh_i \text{ for at least one } i = 1,2,3 \\ 0 \text{ else} \end{cases}$$

A hangover of 5 frames is applied to the falling transition of the VAD-Flag. Then by using the VAD an additional gain is calculated for non-speech periods only. This gain is applied directly to C0.

4. Blind Equalisation

A description of the blind equalisation process is available in [1]. This module reduces the convolutional distortion caused by the microphone and transmission channel.

The blind equalisation used relies on a LMS algorithm, which adjusts the cepstral coefficients according to the difference between the current cepstral vector and a reference cepstrum. The reference cepstrum corresponds to the cepstrum of a flat spectrum.

5. Backend

In our DSR front-end proposals, the noise reduction and the Mel-frequency cepstrum analysis are processed on the terminal. Static coefficients are transmitted. Then a few computations are conducted on the server side (backend) prior accessing the decoder module. These computations include the calculation of the temporal derivatives and a selection of feature vectors.

First and second order temporal derivatives are computed on a 9 frame-window, centred over the current frame.

The feature vector selection process is described in details in [3]. Its role is to discard part of the noisy frames that often badly match with the "silence" models when there is a significant mismatch between training and test conditions. A voice activity detector is used to detect the non-speech frames. Its decision is based on a comparison of the frame energy with an adaptive threshold. Contrary to [3] only noisy frames at the beginning of the file were dropped.

6. Experiments

6.1. Experimental setup

In order to evaluate the impact of TDNR and FDNR, we compared recognition performance with and without noise reduction modules on the Aurora 2 and SDC-Italian databases.

The Aurora 2 database [4] is a composition of the Tidigits database and noise data. Eight noise types at 7 SNR conditions from clean to -5 dB were defined. Tidigits contains connected digits spoken by American English talkers. Noise signals are recorded in several typical environments. Six test sets are provided for testing the performance with the HTK recogniser.

The SDC-Italian database [5] is recorded in car environment with Italian talkers. Several environmental conditions (high speed, low speed, stopped, window open, ...) are used and data is collected from a close talk microphone and hands-free microphone simultaneously. For the evaluation we used only the tests containing connected digits. Three test sets are provided (well matched, medium mismatched, highly mismatched) to asses the performance with the HTK recogniser.

6.2. Results

The basis for evaluating the gain of the noise reduction were simulations with both front-ends and noise reduction switched off. Tables 1 and 3 show the obtained results on Aurora 2, together with a comparison with the standard ETSI Aurora Mel-cepstrum front-end WI-007. It can be seen that even without noise reduction a relative improvement 15.7% (resp. 24.6%) were achieved. If the noise reduction is switched on this gain increases to 44.5% (resp. 44,6%) for FDNR (resp. TDNR) algorithms.

Table 5 reports the results on the SDC Italian database. Results are given for the 3 test conditions: well-matched (WM), medium mismatched (MM) and high mismatched (HM).

Table 1 - Recognition performance on Aurora 2 with C0 and without noise reduction

Absolute performance without Noise reduction				
Training Mode	Set A	Set B	Set C	Overall
Multicondition	87,70	88,46	86,11	87,68
Clean Only	67,73	68,75	71,03	68,80
Average	77,71	78,61	78,57	**78,24**
Performance relative to Mel-cepstrum				
Training Mode	Set A	Set B	Set C	Overall
Multicondition	-0,98	15,94	14,35	9,50
Clean Only	16,51	29,39	14,45	21,87
Average	7,77	22,67	14,40	**15,69**

Table 2 - Recognition performance on Aurora 2 with C0 and frequency domain noise reduction algorithm

Absolute performance with frequency domain NR				
Training Mode	Set A	Set B	Set C	Overall
Multicondition	90,62	90,69	89,92	90,51
Clean Only	83,63	83,83	82,62	83,51
Average	87,12	87,26	86,27	**87,01**
Performance relative to Mel-cepstrum				
Training Mode	Set A	Set B	Set C	Overall
Multicondition	22,98	32,22	37,85	30,26
Clean Only	57,64	63,46	48,67	58,70
Average	40,31	47,84	43,26	**44,48**

Table 3 - Recognition performance on Aurora 2 with logE and without noise reduction

Absolute performance				
Training Mode	Set A	Set B	Set C	Overall
Multicondition	89,03%	88,98%	85,00%	88,20%
Clean Only	74,44%	74,40%	74,27%	74,39%
Average	81,73%	81,69%	79,64%	**81,29%**

Performance relative to Mel-cepstrum				
Training Mode	Set A	Set B	Set C	Overall
Multicondition	9,93%	19,70%	7,54%	13,30%
Clean Only	33,87%	42,15%	24,01%	35,87%
Average	21,90%	30,92%	15,77%	**24,58%**

Table 4 - Recognition performance on Aurora 2 with logE and frequency domain noise reduction algorithm

Absolute performance				
Training Mode	Set A	Set B	Set C	Overall
Multicondition	91,26%	90,68%	88,68%	90,51%
Clean Only	84,44%	83,82%	81,54%	83,61%
Average	87,85%	87,25%	85,11%	**87,06%**

Performance relative to Mel-cepstrum				
Training Mode	Set A	Set B	Set C	Overall
Multicondition	28,30%	32,12%	30,23%	30,30%
Clean Only	59,75%	63,45%	45,47%	58,97%
Average	44,03%	47,78%	37,85%	**44,63%**

Table 5 – Recognition performance on SDC-Italian database

Absolute performance	WM	MM	HM
Baseline (WI007)	93.64	82.02	39.84
No NR & C0	94.98	81.28	56.22
No NR & loge	95.00	83.20	73.60
FD NR & C0	96.30	90.77	83.60
TD NR & logE	96.64	91.29	86.25

For SDC-Italian the results observed on Aurora 2 are confirmed, as both front-ends achieve significant improvements when switching the noise reduction on.

6.3. Discussion

Figure 4 compares the performances with and without noise reduction. The two complete front-ends, including noise reduction modules (whether TDNR or FDNR) provide rather similar performance on the various data base subsets. As compared to the performances without noise reduction, the improvement is small for the well-match conditions, gets larger for the medium mismatch conditions and is the most important for the high-mismatch conditions.
The second and third front-ends (No NR & C0, and No NR & Log E) include the blind equalisation that reduces the (convolution) channel effect. Compared to the baseline system, a large improvement is observed on the high-mismatch conditions, where training and test conditions are very different.

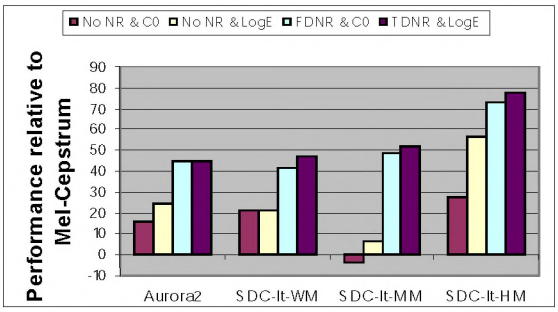

Figure 4 – Noise reduction performances

7. Conclusion

This paper has presented two ways of achieving noise robust front-ends. Both include an efficient noise reduction module, which reduces the effect of the additive noise, and a blind equalisation module, which reduces the convolutional noise. In one case the noise reduction is done in the time domain, before computing the Mel-frequency cepstral coefficients, whereas in the other case the noise reduction module is part of the cepstral computation chain. Both approaches prove to be useful and achieve a noticeable error rate reduction (around 50% reduction) as compared to the baseline system.

Acknowledgements

This work was partially supported by the SMADA European project. The SMADA project is partially funded by the European Commission, under the Action Line Human Language Technology in the 5th Framework IST Programme.

References

[1] Mauuary, L., "Blind equalization in the cepstral domain for robust telephone based speech recognition", *proceedings EUSIPCO'98, IX European Signal Processing Conference*, September 8-11, 1998, Rhodes, Greece, vol. 1, pp. 359-363.

[2] Yang, J., "Frequency domain noise suppression approaches in mobile telephone systems", *Proc. ICASSP*, April 1993, vol. 2, pp. 363-366.

[3] de Veth, J., de Wet, F., Mauuary, L., Noe, B., Sienel, J., Boves, L., Jouvet, D., "Feature vector selection to improve ASR robustness in noisy conditions", *submitted to this conference.*

[4] Hirsch, H.-G., Pearce, D., "The AURORA Experimental Framework for the Performance Evaluation of Speech Recognition under Noisy Conditions", *In Proceedings of the ISCA ITRW ASR 2000*, September 2000, Paris, France, pp. 181-188.

[5] Knoblich, U., "Description and baseline results for the subset of the SpeechDat Car Italian database used for ETSI STQ Aurora WI008 advanced DSR front-end evaluation", *STQ Aurora DSR Working Group*, document AU/237/00, 2000.