Title: **Applying the Advanced ETSI frontend to the Aurora-2 task**

Author: **Guenter Hirsch, Niederrhein University of Applied Sciences**
David Pearce, Motorola

Date: **1st September 2006**
Version: **1.1**

1 Abstract

A scheme for the extraction of robust acoustic features in the field of speech recognition was selected and defined as standard by ETSI (European Telecommunications Standard Institute) in 2002. As part of this selection process a whole set of recognition experiments including noisy data has been defined and set up for the comparative evaluation of different proposals. These have become publicly available for use by the speech community. One often used experiment is the so called "Aurora-2" task that contains distorted versions of the well known TIDigits data base. While the detailed results for applying the advanced front-end (AFE) on the Aurora-2 task have been presented inside the Aurora working group, they have not been completely and officially published at a conference or as part of a scientific paper. Furthermore small modifications have been introduced later on.
Because of some alternative operation modes and because of missing notes about the exact application, several researchers applied the front-end on the Aurora-2 task in different ways. So, they achieved different recognition results which they take as baseline results for their own experiments. This makes the comparison of different approaches and their performance difficult. The intention of this paper is to fill this gap of information and present the detailed results.
Both authors were active members of the Aurora working group where one author was also part of the consortium whose proposal was finally selected as the standard scheme. A link to a Web site is given where scripts can be downloaded for achieving the baseline results.
After presenting some information about the Aurora activities a few details will be given about the AFE and the Aurora-2 task. The detailed recognition results will be presented.

2 Background

The Aurora working group (formally named the ETSI "STQ-Aurora DSR working group") was mainly active from about 1997 till 2002 and after this further evaluation and the definition of fixed-point DSR standards were conducted within 3GPP. A brief background relating to the performance benchmarking with the Aurora-2 database is given below but for more detailed information about the standards development see reference [8].

The goal in ETSI Aurora was the definition and the standardization of two schemes for extracting the acoustic features from a speech signal for automatic speech recognition. The target scenario is a distributed realization of speech recognition (DSR) where the acoustic features are extracted in any type of terminal in a fixed or mobile network and are transmitted to a recognition system at a remote position somewhere in the network. The standard documents as well as an exemplary software realization as floating point C code are available for both schemes from ETSI [1]. The first one consists of an "usual" cepstral analysis scheme. The second one can be seen as an extension of the first one by adding two further processing blocks for extracting robust features in the presence of background noise and unknown frequency characteristics as they occur in real application scenarios. Besides the intention of applying this feature extraction scheme in a DSR scenario as assumed for the standardization within ETSI, more in general the AFE has been taken as reference of a robust front-end for investigations on robust recognition.

During the process of defining the second standard, several data bases have been created or set up to enable the comparison of different approaches with respect to their performance on recognizing distorted speech data. The first data base is referenced under the abbreviation “Aurora-2” [2]. It is based on the usage of the well known TIDigits data base, containing sequences of English digits [3]. Distortions have been artificially added to the data. The Aurora-2 CDs contain a set of scripts and configuration files for running the experiments with the first standardized front-end. Because this experiment was released during the process of defining the advanced front-end it does not contain scripts and results for running the recognition experiments with the second standard.

All the Aurora databases themselves have been made available publicly and are distributed by ELRA [9].

3 Advanced Front-End (AFE)

The standardized AFE is based on a cepstral analysis scheme as it also part of the first standard [4]. The analysis scheme of the first front-end has been extended by two further processing blocks to achieve higher recognition performance in situations with noise in the background and with unknown modifications of the frequency characteristics due to e.g. the microphone or the transmission channel. A block diagram is shown in figure 1.

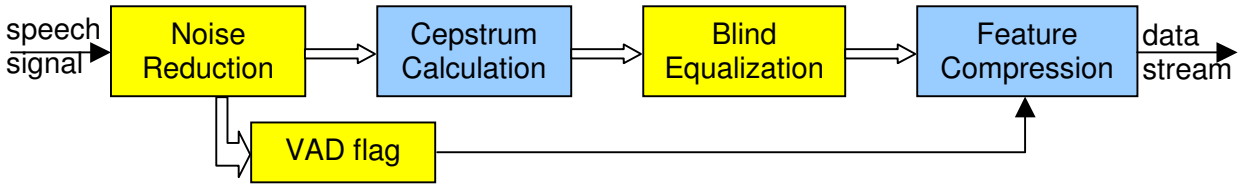

Figure 1: Block diagram of the AFE

A two stage Wiener filter is applied on the speech signal as processing step to reduce background noise. The filter characteristics is estimated in the frequency domain where the filtering itself is done in the time domain after transforming back the estimated filter characteristics to the time domain. The filter estimation is individually done for short segments of the signal where two consecutive frames have a difference in time of 10 ms. A further SNR dependent waveform processing is applied on the filtered signal.
Furthermore a VAD (voice activity detection) flag is created as part of the noise reduction process for each frame. This flag is included as part of the data stream at the output so that it could be used for excluding frames from the recognition process at the recognition stage.

The noise reduced signal is taken as input to a cepstral analysis scheme that is almost identical with the scheme that has been defined as first standard. The output of this processing block are 13 cepstral coefficients (including C0) and 1 logarithmic energy coefficient per frame.
The cepstral coefficients (without C0) are processed with a blind equalization scheme to compensate the influence of unknown frequency characteristics. This blind equalization is based on the comparison to a flat spectrum and the application of the LMS algorithm.

Finally the 13 cepstral coefficients and the energy coefficient are compressed with the means of a split vector codebook. The outcoming data stream, that contains also the VAD flag, can be taken for a circuit data or a packet data transmission.

More details about the front-end processing can be found in the standard document [5] or in [6].

ETSI maintains and distributes the standards documents defining the algorithms and the reference floating point C code. Updates to the version are made either when there are updates to the text documentation or to the software itself. The floating point C code of version 1.1.3 of ES 202 050 from November 2003 has been used to generate the results presented in this paper (1.1.3 is the latest version at the time of writing this paper).

Several possibilities exist for applying the AFE as part of a recognition experiment:

- The cepstral and the energy coefficients can be taken as they occur before the compression. Alternatively the corresponding coefficients can be taken after compression and decompression in case the influence of the transmission should be studied.
- Up to 14 acoustic parameters are available to build the feature vector for the recognition, where C0 and the logarithmic energy coefficient describe the frame energy. Thus, these two coefficients are highly correlated. A proposal is made in the standard document how to combine these two coefficients as a single parameter.
- Usually so called Delta and Delta-Delta parameters are added as further coefficients to a feature vector. These are derived from the static parameters and describe the modification of each acoustic parameter across time. A scheme for the calculation of Delta and Delta-Delta is proposed as part of the standard document. This is based on filtering the contour of each parameter by a given set of filter coefficients.
- The VAD flag can be used to exclude “non-speech” frames from the recognition process. The term “frame dropping” has been introduced for this exclusion of certain frames.

First we will present the detailed results for using the uncompressed parameters as coefficients of the feature vector. We apply the proposed methods for combining the logarithmic energy and C0 as a single coefficient. We add the Delta and Delta-Delta parameters according to the proposed method so that a feature vector finally consists of 39 coefficients. Frame dropping is not applied first. This set-up can be taken to compare alternative approaches for a robust feature extraction against.
Furthermore we present the average results for additionally applying the compression scheme without and with frame dropping.

4 Aurora-2

The Aurora-2 data base and the corresponding recognition experiments are based on the use of a downsampled version of the TIDigits. The software package HTK [7] is applied for modelling speech with the statistical approach of HMMs and recognizing speech by Viterbi decoding.
Two training modes have been defined. One takes only the clean utterances as input data. The second mode is based on the usage of clean and noisy data so that it is referred as multi-condition training.
HTK in its version 3.3 is used to create whole word HMMs for all digits. The gender independent HMMs are defined by the following parameters:

- 16 states per word
- simple left-to-right models without skips over states
- mixture of 3 Gaussians per feature and state

A single HMM that consists of 3 states is used to model the pauses.

Three sets of test data exist. The first set A contains data that have been artificially distorted by adding recorded noise signals at a desired SNR. Four different noise signals have been selected to be added at SNRs of –5, 0, 5, 10, 15 and 20 dB. Thus, in total 28 subsets (= 4

noise signals times 7 conditions {6 SNRs plus clean}) are available where each subset consists of 1000 utterances. The noise signals of set A have also been taken for the creation of the noisy training data. Set B has the same principal organization as set A with 28 subsets. 4 different noise signals have been applied that are not used for the creation of noisy training data. Set C consists of only 14 subsets containing one noise condition of set A and one noise condition of set B. The 7 subsets from set A and the 7 subsets from set B include the above mentioned range of SNRs and the clean condition. Additionally a frequency weighting is applied according to the MIRS filter characteristic as defined by ITU for simulating the influence of using telephone devices with their restriction to the frequency range of about 300 to 3400 Hz.

The recognition performance for a single subset is reflected by the word error rate including substitution, deletion and insertion errors. For each of sets A and B an average performance measure is calculated as average over 20 word error rates where the clean and the –5 dB conditions are excluded. These two measures for sets A and B should reflect the performance in situations where noise is present in the background. In the same way an average performance is calculated for the corresponding 10 subsets of set C. This should reflect the condition with noise in the background and with an additional frequency weighting. A measure for the total performance is calculated as average error rate over all 50 mentioned conditions of the three subsets.

5 Recognition results

The results of the recognition experiments will be presented for different modes of applying the advanced front-end.
Regarding the practical C code implementation as available at ETSI, two respectively three programs have to be called with different command line flags for the different modes. The first program creates the feature vectors containing the 14 static parameters. The second program does the compression and decompression of the features simulating the transmission at a fixed data rate. This program is called only in the cases including compression. The third program is applied to create the energy parameter as combination of the zeroth cepstral coefficient and the logarithmic frame energy and to calculate the Delta and Delta-Delta features as defined in the standard. Details about the command line flags can be found in the scripts files that are mentioned in the last section of this report.

5.1 AFE without compression and without frame dropping

As described before the results in this section are obtained from applying the advanced front-end without compressing the cepstral and energy parameters and without applying the optional frame dropping for excluding non-speech frames from the recognition process.

5.1.1 Training on clean data

The detailed results are listed as word error rates in tables 1 to 3 for the three test sets when training the set of gender independent HMMs on clean data only.

SNR	Subway	Babble	Car	Exhibition	Average
Clean	0.68 %	0.94 %	0.84 %	0.65 %	
20dB	1.87 %	1.93 %	1.22 %	1.79 %	1.70 %
15dB	3.72 %	3.14 %	2.15 %	3.30 %	3.08 %
10dB	7.31 %	7.98 %	4.03 %	6.48 %	6.45 %
5dB	14.06 %	18.23 %	9.84 %	14.50 %	14.16 %

0dB	33.50 %	46.86 %	29.20 %	34.13 %	35.92 %
-5dB	64.91 %	78.36 %	66.54 %	65.01 %	
Average (0-20 dB)	12.09 %	15.63 %	9.29 %	12.04 %	**12.26 %**

Table 1: Word error rates for test set A in clean training mode

SNR	Restaurant	Street	Airport	Train-Station	Average
Clean	0.68 %	0.94 %	0.84 %	0.65 %	
20dB	1.93 %	2.18 %	1.40 %	1.39 %	1.73 %
15dB	4.45 %	3.42 %	2.39 %	2.96 %	3.31 %
10dB	8.23 %	6.71 %	5.67 %	5.18 %	6.45 %
5dB	20.82 %	15.39 %	13.12 %	13.48 %	15.70 %
0dB	45.13 %	35.97 %	35.73 %	32.58 %	37.35 %
-5dB	77.62 %	67.74 %	68.60 %	66.00 %	
Average (0-20 dB)	16.11 %	12.73 %	11.66 %	11.12 %	**12.91 %**

Table 2: Word error rates for test set B in clean training mode

SNR	Subway	Street	Average
Clean	0.71 %	0.79 %	
20dB	2.58 %	2.18 %	2.38 %
15dB	4.39 %	3.48 %	3.94 %
10dB	8.35 %	7.74 %	8.05 %
5dB	17.16 %	17.71 %	17.44 %
0dB	40.93 %	41.05 %	40.99 %
-5dB	70.59 %	71.40 %	
Average (0-20 dB)	14.68 %	14.43 %	**14.56 %**

Table 3: Word error rates for test set C in clean training mode

The average performances for set A and B are almost the same. Each set contains one noise condition (“babble” respectively “restaurant”) that leads to a slightly worse error rate in comparison to the other conditions. The error rates for set C with the additional frequency weighting are higher in general when comparing them to the corresponding results without spectral modification.

5.1.2 Training on multi-condition data

The detailed results are listed as word error rates in tables 4 to 6 for the three test sets when training the set of gender independent HMMs on multi-condition data.

SNR	Subway	Babble	Car	Exhibition	Average

Clean	0.83 %	0.94 %	0.86 %	0.52 %	
20dB	1.29 %	1.42 %	1.13 %	1.54 %	1.35 %
15dB	2.30 %	2.30 %	1.49 %	2.01 %	2.03 %
10dB	4.73 %	3.93 %	2.56 %	4.07 %	3.82 %
5dB	8.50 %	9.67 %	6.23 %	8.76 %	8.29 %
0dB	22.38 %	29.84 %	17.24 %	22.80 %	23.07 %
-5dB	54.93 %	67.96 %	51.48 %	51.56 %	
Average (0-20 dB)	7.84 %	9.43 %	5.73 %	7.84 %	**7.71 %**

Table 4: Word error rates for test set A in multi-condition training mode

SNR	**Restaurant**	**Street**	**Airport**	**Train-Station**	**Average**
Clean	0.83 %	0.94 %	0.86 %	0.52 %	
20dB	1.57 %	1.75 %	1.13 %	1.02 %	1.37 %
15dB	2.24 %	2.30 %	2.03 %	1.94 %	2.13 %
10dB	4.51 %	3.84 %	3.64 %	3.46 %	3.86 %
5dB	11.39 %	9.58 %	7.58 %	8.95 %	9.38 %
0dB	29.51 %	23.43 %	21.50 %	23.17 %	24.40 %
-5dB	67.24 %	56.92 %	54.22 %	52.42 %	
Average (0-20 dB)	9.84 %	8.18 %	7.18 %	7.71 %	**8.23 %**

Table 5: Word error rates for test set B in multi-condition training mode

SNR	**Subway**	**Street**	**Average**
Clean	0.89 %	0.91 %	
20dB	1.50 %	1.72 %	1.61 %
15dB	2.18 %	2.39 %	2.29 %
10dB	3.87 %	4.35 %	4.11 %
5dB	9.46 %	10.40 %	9.93 %
0dB	27.63 %	28.78 %	28.21 %
-5dB	63.49 %	63.15 %	
Average (0-20 dB)	8.93 %	9.53 %	**9.23 %**

Table 6: Word error rates for test set C in multi-condition training mode
As expected in general the error rates are considerably lower when comparing them with the results in clean training mode.

5.1.3 Average performance

The average performances as listed in tables 1 to 6 are summarized in table 7.

Training mode	Test set		
	A	**B**	**C**
clean	12.26 %	12.91 %	14.56 %
multi-condition	7.71 %	8.23 %	9.23 %

Table 7: Average word error rates for AFE without compression and without frame dropping

5.2 AFE with compression but without frame dropping

The average performances for the three test sets are listed in table 8 for clean and multi-condition training mode when additionally compressing the acoustic features as defined in the standard. The results are almost the same as for the case without compression. Because of this we do not present the results for all conditions in detail again.

Training mode	Test set		
	A	**B**	**C**
clean	12.19 %	12.91 %	14.23 %
multi-condition	7.86 %	8.46 %	9.39 %

Table 8: Average word error rates for AFE with compression and without frame dropping

5.3 AFE with compression and with frame dropping

Finally the average results are presented in table 9 for the compressed features and applying the optional frame dropping. The results are not much different in comparison to the previous cases. The VAD based exclusion of pause segments from the recognition has no major influence on the recognition performance in case of the TIDigits. There are only pause segments of a few hundred milliseconds at the beginning and at the end of these data. For data with longer pause segments (for example the Aurora-3 databases) the enabling of the frame dropping can considerably reduce the number of insertion errors. It is noted that this is the particular configuration that was used during the Aurora selection.

Training mode	Test set		
	A	**B**	**C**
clean	12.46 %	13.03 %	14.38 %
multi-condition	7.96 %	8.07 %	9.73 %

Table 9: Average word error rates for AFE with compression and with frame dropping

6 Scripts

A set of shell scripts can be downloaded at http://aurora.hsnr.de

These shell scripts can be used to run the experiments mentioned above. The scripts are modified versions of these scripts that are part of the Aurora-2 CDs and that are used for running the experiments with the first standardized front-end. There are mainly 3 scripts called "create_pattern_etsi2", "train_recog_clean_etsi2" and "train_recog_multi_etsi2". The original scripts on the CDs under the subdirectory <recognizer> have the same names just without the extension "_etsi2". Like with the original scripts, the headers of all scripts have to be edited to set the paths to the speech data and the recognition scripts.
There exist two further scripts "create_pattern_quant_etsi2" and "create_pattern_quant_fd_etsi2" for applying the front-end with compression but without frame dropping or with compression and with frame dropping. ASCii files are created as output of the recognition scripts containing the recognition results (output of the HTK Viterbi recognizer) for all conditions. A script "eval_results_html" is available to create a HTML file containing the error rates for all test sets and training modes in separate tables.
Furthermore a script "do_all_etsi2" is available that can be used to run the whole experiment by calling the scripts mentioned before. This script has to be edited to enable the desired front-end mode and to define some paths in your working environment.

References

[1] http://pda.etsi.org/pda/queryform.asp (search for ES 201108 and ES 202050)

[2] H.G. Hirsch, D. Pearce, "The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions", *ISCA workshop on automatic speech recognition*, Paris, France, 2000

[3] R.G. Leonard, "A database for speaker independent digit recognition, *ICASSP84*, Vol.3, p.42.11, 1984

[4] ETSI standard document, "Speech Processing, Transmission and Quality aspects (STQ); Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithm", *ETSI ES 201 108 v1.1.3 (2003-09)*, Sep. 2003

[5] ETSI standard document, "Speech Processing, Transmission and Quality aspects (STQ); Distributed speech recognition; Advanced Front-end feature extraction algorithm; Compression algorithm", *ETSI ES 202 050 v1.1.3 (2003-11)*, Nov. 2003

[6] D. Macho, L. Mauuary et. al.: "Evaluation of a noise robust DSR front-end on Aurora databases", *7th International Conf. on Spoken Language Processing*, Denver, 2002

[7] S. Young et al.: "The HTK book (for version 3.3)", *http://htk.eng.cam.ac.uk* , April 2005

[8] D Pearce, "Enabling Speech & Multimodal Services on Mobile Devices: The ETSI Aurora DSR standards & 3GPP Speech Enabled Services," VoiceXML Review, Nov/Dec 2004.
www.voicexmlreview.org/Nov2004/features/dsr.html

[9] European Language Resources Association
http://www.elda.org/