INVESTIGATIONS ON THE COMBINATION OF FOUR ALGORITHMS TO INCREASE THE NOISE ROBUSTNESS OF A DSR FRONT-END FOR REAL WORLD CAR DATA

Bernt Andrassy [a], Florian Hilger [b] and Christophe Beaugeant [c]

[a] SIEMENS AG, CT IC 5, Otto-Hahn-Ring 6, D-81730 München
bernt.andrassy@mchp.siemens.de
[b] Lehrstuhl für Informatik VI, RWTH Aachen – University of Technology, D-52056 Aachen
hilger@informatik.rwth-aachen.de
[c] SIEMENS AG, ICM MP RD MCH 82, Haidenauplatz 1, D-81667 München
christophe.beaugeant@mch.siemens.de

ABSTRACT

This paper shows how the noise robustness of a MFCC feature extraction front-end can be improved by integrating four noise robustness algorithms being a Spectral Attenuation -, a Noise Level Normalistion -, a Cepstral Mean Normalization - and a Frame Dropping algorithm.

The algorithms were tested separately and in varying combinations on three real world car data sets with different amounts of mismatch between the training and the testing conditions. It was shown that although the algorithms partly have similar effects none of them is completely redundant. Every algorithm can contribute to a further improvement of the recognition results so the best results can be achieved by a combination of all four of them. A relative reduction of the word error rate of up to 57% is achieved.

1. INTRODUCTION

Distributed Speech Recognition (DSR) i.e. the separation of the feature extraction from the actual speech recognition allows to split up the speech recognition into a computationally simple front-end and a computationally complex back-end. Using centralised servers as backends which share the computational burden between users and enable the easy upgrade of technologies allows to implement the speech recognition technology in mobile devices used as front-ends with their constraints regarding computational complexity and memory requirements.

The problem of codec distortion as well as of packet losses that arise due to the transmission over mobile voice networks is addressed by sending a parametrised representation which is sufficient for recognition and dropping the speech channel. This approach also leads to a decrease in the amount of data to be transmitted.

In order to promote this concept the ETSI-STQ-Aurora working group has already introduced a first standard [1] for a DSR front-end and is currently working at a new standard for an advanced noise robust DSR front-end.

This paper shows one way of improving the noise robustness of the first standard by integrating four different algorithms into the front-end. In order to get information about how the algorithms work together, they were integrated separately and in varying combinations with each other.

2. FOUR ALGORITHMS IMPROVING NOISE ROBUSTNESS

In this section the four algorithms that were added to the baseline front-end are introduced. The first algorithm is based on a well-known family of speech enhancement algorithms, the so called 'short-time spectral attenuation algorithms' [2] and is simply called Spectral Attenuation (SA) in the following. It was originally developed to increase the intelegibility of a speech signal for humans, whereas the other algorithms are specific to the use with automatic speech recognition systems.

2.1. Spectral Attenuation

With the SA-algorithm the analysis is performed in the frequency domain.

The estimation of the short-time spectrum of the speech $\hat{S}_f[t]$ is obtained by applying an attenuation function $G_f[t]$ to the spectrum of the noisy speech $Y_f[t]$:

$$\hat{S}_f[t] = G_f[t] \cdot Y_f[t] \tag{1}$$

For $G_f[t]$ a Wiener filter is used which can be expressed according to equation (2):

$$G_f[t] = \frac{\gamma_{ss,f}[t]}{\gamma_{ss,f}[t] + \gamma_{nn,f}[t]} \tag{2}$$

where $\gamma_{ss,f}[t]$ and $\gamma_{nn,f}[t]$ stand for the power spectral densities (psd) of the speech and of the noise. They are computed through the first order IIR filtering expressed by :

$$\gamma_{uu,f}[t] = (1-\lambda)\left(\lambda \cdot \gamma_{uu,f}[t-1] + \left|\tilde{U}_f\right|^2[t] \right) \tag{3}$$

where $\lambda < 1$, $u \in \{s,n\}$, $\tilde{U} \in \{\tilde{S}, \tilde{N}\}$ and $\tilde{S}_f[t]$ and $\tilde{N}_f[t]$ are rough etimates of the short-time spectra of the speech and of the noise.

After the Spectral Attenuation, the signal is passed through a Mel scaled filter bank. The following Noise Level Normalization is applied to these filter bank outputs.

2.2. Noise Level Normalization

Noise Level Normalization (NLN) [3] is an approach to normalize the noise level at the outputs of the Mel scaled filterbank $Y_k[t]$ during recognition to a level observed in training.

The NLN distinguishes between speech and non-speech parts of the signal. Only the non-speech parts of the signal are normalized (i.e. scaled down) by applying a normalizing factor $\nu_k[t]$ while the speech parts are left unchanged.

The normalizing factor $\nu_k[t]$ is calculated using the current estimates for the speech level $\overline{Y}^s$ averaged over all filters, the estimated noise levels for each filter k $\overline{Y}_k^n$ and the average relation between the noise and the speech level determined in training γ [3].

$$\nu_k[t] = \min\left(\gamma \cdot \frac{\overline{Y}^s[t]}{\overline{Y}_k^n[t]} \; , \; 1 \right) \tag{4}$$

The normalizing factor is clipped at an upper limit of 1 to avoid unintended raising of a noise that might be lower than the average in training.

As the Aurora evaluations require one setup for all databases, γ was set to a predefined value that leads to good results on all databases and was not set to a value which was specified in training.

The speech non-speech distinction is done by a sigmoid function $s[t]$ which works as a soft threshold.

Combining $\nu_k[t]$ and $s[t]$ leads to the normalizing function with the properties described above:

$$Y_k^{norm}[t] = \underbrace{(\nu_k[t] + (1-\nu_k[t]) \cdot s[t])}_{H_k[t]} \cdot Y_k[t] \tag{5}$$

A clean testing signal is not changed. Speech parts of noisy signals also remain unchanged $s[t] \approx 1 \Rightarrow H_k[t] \approx 1$, while the level of the noise parts is scaled down $s[t] \approx 0 \Rightarrow H_k[t] \approx \nu_k$.

2.3. Cepstral Mean Normalization

Cepstral Mean Normalization (CMN) is a very simple but efficient method to compensate for the so called convolutive noise that arises from channel distortions. It has found widespread use in many systems. Here an online implementation without delay was chosen. The means are estimated using a weighted sum of the current feature vector components $X_c[t]$ and the previous estimate:

$$\overline{X}_c[t] = (1-\tau)\, \overline{X}_c[t-1] + \tau\, X_c[t] \tag{6}$$

An optimal value for the time constant was found to be $\tau = 0.01$. Having updated the means, the normalized cepstral coefficients are obtained by subtraction of the means:

$$X_c^{cmn}[t] = X_c[t] - \overline{X}_c[t] \tag{7}$$

2.4. Frame Dropping

The Aurora evaluation scheme [4] does not allow any modifications of recognizer parameters, which can lead to an unfavorably large number of insertion errors on some of the test sets. This effect can be encountered by eliminating silence parts of the signal during feature extraction (Frame Dropping, FD). Here a criterion based on the energy ahead $\overline{Y}[t+5]$ and on the current noise estimate $\overline{Y}^n[t]$ calculated from the Mel filter outputs was chosen:

- If $\overline{Y}[t+5] < \vartheta \cdot \overline{Y}^n[t]$ for T_{off} succeeding time frames, the skipping mode is turned on and the following time frames are not passed on to the recognizer. Optimal values for ϑ and T_{off} were found to be $\vartheta = 1.0$ and T_{off} = 7 .
- If $\overline{Y}[t+5] \geq \vartheta \cdot \overline{Y}^n[t]$ for T_{on} time frames, the skipping mode is turned off again, and the time frames are passed on to the recognizer. Here T_{on} is one time frame.

3. RESULTS AND DISCUSSION

3.1. Database definition

In order to evaluate the performance of the different algorithms they were tested on a subset of the Spanish SpeechDat Car database as it is also used in the Aurora group. The subset is derived from the SpeechDat Car data [5]. The database contains isolated digits as well as continuous digits. The sampling rate is 8kHz. There are more than 3000 files from over 100 speakers from both genders. All files were recorded in cars with varying noise conditions. Three tasks are defined with different mismatches beween the training and corresponding test sets concerning the noise condition the files were recorded in. WM stands for a well match, MM for a medium mismatch and HM for a high mismatch between training and test set.

3.2. Baseline recognizer setup

All recognition tests where conducted using the HTK speech recognition toolkit with the settings defined for the ETSI Aurora evaluations [4]. The baseline results reported where obtained with the standardized Aurora WI007 MFCC front end [1].

Baseline setup:

- HTK speech recognition toolkit (Aurora evaluation settings [4])
- MFCC feature extraction (Aurora WI007 [1])
- 23 filters; 13 cep. coeff.+13 first deriv.+13 second deriv. = 39 – dimensional feature vector
- Word models of fixed length (16 states) for the digits
- Gender independent models
- Gaussian mixtures, 552 densities, pooled diagonal covariance matrix

The four new modules described in section 2 where simply added to the existing WI007 modules. Except for using the 0th cepstral coefficient instead of the log-energy none of the given parameters or modules were changed.´

The SA and CMN algorithms were used in training as well as in testing whereas the NLN and FD algorithms were only used in testing.

3.3. Recognition results

Tables 1-4 show the results of the integration of the different algorithms into the front-end.

According to the Aurora evaluation scheme the results are reported as word accuracies while the improvement is measured as the relative reduction *R(a)* of word error rates of an algorithm *a* over the Baseline result:

$$R(a) = \frac{WER_{baseline} - WER_a}{WER_{baseline}} \tag{8}$$

where WER_a is the word error rate of algortihm *a*.
The Total is a weighted sum of the different mismatch conditions:

$$Total = 0.4 \cdot R_{WM} + 0.35 \cdot R_{MM} + 0.25 \cdot R_{HM} \tag{9}$$

Table 1 shows the results of the individual integration of the single algorithms. Apart from the CMN in the WM-condition (-4%) all the algorithms lead to a considerable improvement of the recognition result in all conditions. The SA has the strongest impact on the overall error reduction (R_{Total}=31%). FD, NLN and CMN all lead to a similar Total of *R* (15%, 15% and 16%).

SDC_Spanish					
	Word accuracy [%]				
Training condition	Baseline	SA	NLN	CMN	FD
WM	86.9	90.2	88.6	86.3	88.5
MM	73.7	84.7	79.0	81.7	77.9
HM	42.2	55.5	49.2	59.3	52.5
Total	**71.1**	**79.6**	**75.4**	**78.0**	**75.8**
	Relative reduction of word error rates				
WM	0.0%	26%	13%	-4%	12%
MM	0.0%	42%	20%	30%	16%
HM	0.0%	23%	12%	30%	18%
Total	**0.0%**	**31%**	**15%**	**16%**	**15%**

Table 1: Results of the individual integration of Spectral Attenuation (SA), Noise Level Normalisation (NLN), Cepstral Mean Normalisation (CMN) and Frame Dropping (FD) into the front-end

SDC_Spanish				
	Word accuracy [%]			
Training condition	Baseline	SA + NLN	SA + CMN	NLN+CMN
WM	86.9	91.5	90.4	86.9
MM	73.7	86.0	89.4	83.8
HM	42.2	64.8	75.3	67.6
Total	**71.1**	**82.9**	**86.3**	**81.0**
	Relative reduction of word error rates			
WM	0%	36%	27%	1%
MM	0%	47%	60%	38%
HM	0%	39%	57%	44%
Total	**0%**	**40%**	**46%**	**25%**
	Redundancy			
WM	0%	3%	-6%	8%
MM	0%	15%	12%	12%
HM	0%	-4%	-5%	-2%
Total	**0%**	**6%**	**1%**	**7%**

Table 2: Results of the pairwise integration of SA, NLN and CMN into the front-end

Table 2 shows the results of the pairwise integration of SA, NLN and CMN into the front-end. The pairwise integration should provide information about the redundancy of the different algorithms towards each other. It is shown to which amount the effects of the single algorithms add up if they are combined. For that a measure for the redundancy of the algorithms a and b *Rcy(a+b)* was introduced as follows (10): The relative reductions of word error rates *R(a)* and *R(b)* which are achieved by inserting a single algorithm *a* and a single algorithm *b* separately into the front-end are added up. This sum represents the case that the effects of the algorithms *a* and *b* are independent from each other.

From this sum the reduction of the word error rate *R(a+b)* achieved through a combined integration of algorithms *a* and *b* into the frontend is subtracted.

$$Rcy(a+b) = [R(a) + R(b)] - R(a+b) \tag{10}$$

The result is the amount of the word error rate reduction which is lost due to a similarity in the effects of the algorithms *a* and *b*.

The combination of SA and CMN has hardly any overall redundancy (1%) whereas the combination of SA and NLN (6%) as well as the combination of NLN and CMN (7%) show a significant amount of redundancy. The redundancy between SA and NLN is not surprising because they are similar approaches both applying a gain to the noisy spectrum which is based on estimations of the speech and the noise power spectra. Yet a considerable increase in the recognition performance is still obtained by combining the two.

The combined integration of the three algorithms SA, NLN and CMN into the front-end leads to a further improvement of the recognition results. *R(a+b+c)* the relative reduction of word error rates of the combination of algorithms a, b and c amounts to 52% (Table 3). The three algorithms work especially well together in the HM-condition which can be seen from the negative redundancy of –2%. This means the combination of the three algorithms works better than could have been expected by summing up the individual relative reductions in word error rates.

SDC_Spanish				
Training condition	Baseline	SA + NLN + CMN		
	Word Accuracy [%]		Rel. red. of WER	Redundancy
WM	86.9	91.6	36%	-1%
MM	73.7	89.4	60%	32%
HM	42.2	80.6	66%	-2%
Total	**71.1**	**88.1**	**52%**	**10%**

Table 3: Results of the combined integration of SA, NLN and CMN into the front-end

Another matter of interest was to determine if the increase in the number of noise robust algorithms always leads to an increase in the recognition performance over the front-end with less noise robust algorithms in every mismatch condition. To evaluate this the relative word error rate reductions of the front-end with three algorithms (Table 3) were compared to the maximum relative word error rate reductions which were achieved with any two algorithms (Table 2).

The combination of the three algorithms leads to a significant increase of the recognition results over the improvement achieved so far in the HM-condition (66% relative WER for the three algorithms compared to 57% relative WER for the combination of SA and CMN) and leads to no further improvement in the WM- and MM-conditions (compared to the combination of SA and NLN for the WM task and the combination of SA and CMN for the MM task).

Thus the front-end with the three noise robust algorithms equals or surpasses the front-end with any combination of two algorithms in every single mismatch condition.

SDC_Spanish				
Training condition	Baseline	SA + NLN + CMN + FD		
	Word Accuracy [%]		Rel. red. of WER	Redundancy
WM	86.9	93.0	47%	0%
MM	73.7	90.1	62%	46%
HM	42.2	80.4	66%	16%
Total	**71.1**	**88.8**	**57%**	**20%**

Table 4: Results of the combined integration of SA, NLN, CMN and FD into the front-end

The additional integration of the frame dropping algorithm as a fourth noise robust algorithm into the front-end leads to a further increase of the recognition result (R_{Total}=57%). Yet the recognition result is mainly increased in the WM-condition whereas there is hardly any change in the MM- and HM-condition.

In the WM-condition the number of insertions is usually much higher than the number of deletions. Here the frame dropping can balance the numbers and increase the recognition performance. In the MM condition the contribution is much smaller. In the HM condition the number of deletions is already high, so there is no effect on the performance.

3.4. Results with other languages

The combination of all four algorithms was also tested on other languages to see if the conclusions drawn from the Spanish data could be generalized. They were tested with the two real world noise car databases, SpeechDat Car Danish [6] and SpeechDat Car German [7]. Those databases have similar properties as described for the Spanish.

It can be seen from the word accuracies in Table 5 that similar recognition rates to the Spanish database can be achieved for the German database whereas the word accuracies for the Danish database are not as good. The Danish Database seems to be a particularily difficult task which can be seen from the very poor recognition results with the Baseline front-end. Yet the average relative reduction of the word error rates by 40% is still high.

	SDC_Danish			SDC_German		
Training condition	Baseline	SA+NLN+CMN+FD		Baseline	SA+NLN+CMN+FD	
	Word Accuracy [%]		Rel. red. of WER	Word Accuracy [%]		Rel. red. of WER
WM	80.2	86.6	32%	90.6	92.2	17%
MM	51.2	72.0	43%	79.1	85.5	31%
HM	33.1	64.2	47%	74.3	84.4	39%
Total	**58.3**	**75.9**	**40%**	**82.5**	**87.9**	**27%**

Table 5: Results of the Integration of SA, NLN, CMN and FD into the front-end achieved with the Databases SDC_Danish and SDC_German

The very high relative error rate reductions obtained on the Spanish database that was used for the parameter optimisations are not reached on the Danish and German databases. But a general tendency is confirmed: the relative error rate reductions are still high and they rise considerably with a growing amount of mismatch between training and testing conditions.

4. CONCLUSION

The integration of the four algorithms, Spectral Attenuation, Noise Level Normalization, Cepstral Mean Normalization and Frame Dropping into an MFCC front-end with the aim of improving the noise robustness of speech recognition has been described. The integration led to significant relative reductions of the word error rates for three real world noise car databases. It was particularly successful under high mismatch conditions between the training and the test sets.

Furthermore it was shown that although threre is a considerable amount of similarity in the effects of the different algorithms none of them is redundant.

5. REFERENCES

[1] "ETSI ES 201 108 v1.1.2 Distributed Speech Recognition; Front-end Feature Extraction Algorithm; Compression Algorithm", April 2000.

[2] Lim, J.S. *Speech enhancement* Prentice-Hall Signal Processing Series Alan V. Oppenheim, Englewood Cliffs New Jersey, 1983.

[3] Hilger, F. and Ney, H., "Noise Level Normalization and Reference Adaptation for Robust Speech Recognition", in *ASR2000 - International Workshop on Automatic Speech Recognition*, pp. 64-68. Paris, France, September 2000.

[4] Hirsch, H.-G. and Pearce, D. "The Aurora Experimental Framework for the Performance Evaluation of Speech Recognition Systems under Noisy Conditions", in *ASR2000 - International Workshop on Automatic Speech Recognition*, pp. 181-188. Paris, France, September 2000.

[5] Moreno, A. "SpeechDat Car: Spanish – Selected Subset, Ver.1", Universitat Politécnica de Catalunya, Spain, April 2000.

[6] Lindberg, B. "Danish SpeechDat-Car Database for ETSI STQ Aurora Advanced DSR", *documentation on the Danish Aurora Project Database CD-ROMs*, January 2001.

[7] Netsch, L. "Description and Baseline Results for the Subset of the SpeechDat-Car German Database used for ETSI STQ Aurora WI008 Advanced Front-End Evaluation", *documentation on the German Aurora Project Database CD-ROMs,* January 2001.