Recognition Performance of the Siemens Front-end with and without Frame Dropping on the Aurora 2 Database

Bernt Andrassy\textsuperscript{a}, Damjan Vlaj\textsuperscript{b} and Christophe Beaugeant\textsuperscript{c}

\textsuperscript{a} SIEMENS AG, CT IC 5, Otto-Hahn-Ring 6, D-81730 München  
bernt.andrassy@mchp.siemens.de

\textsuperscript{b} University of Maribor, Faculty of Electrical Engineering and Computer Science, Smetanova 17, 2000 Maribor, Slovenia  
damjan.vlaj@uni-mb.si

\textsuperscript{c} SIEMENS AG, ICM MP RD MCH 82, Haidenauplatz 1, D-81667 München  
christophe.beaugeant@mch.siemens.de

Abstract

Following the objective of the Eurospeech special event, 'Noise Robust Recognition', the recognition results of a noise robust front-end, developed by Siemens, on the Aurora 2 database [1] are presented in this paper. The front-end was tested with and without a frame dropping algorithm. It is shown that the front-end improves the recognition results in high mismatch between training and testing by 43.90\% over the reference front-end and works particularly well in conditions with high noise. Furthermore it is shown that the frame dropping mainly increases the performance of the front-end.

1. Introduction

The front-end employed is a cepstral analysis scheme with spectral attenuation and spectral subtraction, a channel compensation, calculation of the time derivatives, an LDA and a frame dropping algorithm (Figure 1).

Details of the front-end can be seen from Table 1. The specific algorithms are shortly described in the following.

<table>
<thead>
<tr>
<th>Parameter settings of the front-end</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame length: 32 ms</td>
</tr>
<tr>
<td>Frame shift: 15 ms</td>
</tr>
<tr>
<td>Number of samples per frame, NF: 256</td>
</tr>
<tr>
<td>Preemphasis factor: 0.95</td>
</tr>
<tr>
<td>Length of FFT, FFTL: 256</td>
</tr>
<tr>
<td>Number of Mel filter banks: 15</td>
</tr>
<tr>
<td>Number of cepstral coefficients: 12</td>
</tr>
<tr>
<td>Number of first derivatives: 13</td>
</tr>
<tr>
<td>Number of second derivatives: 13</td>
</tr>
<tr>
<td>Size of the feature vector: 24</td>
</tr>
</tbody>
</table>

Table 1: Parameter settings of the front-end

2. Description of the front-end

2.1. FFT and bandpass filtering

An FFT of the order 8 is applied which requires a block of 256 values to be processed. Depending on the value of NF the vector $X^n$ containing all sample values in frame $n$ is filled up with $FFT L - NF$ zeros (zero padding). With the FFT of the zero padded vector $X^n$ the magnitude spectrum is computed:

$$b_{nk} = \frac{FFT L - 1}{\sum_{n=0}^{FFT L - 1} X^n \cdot e^{-j\frac{2\pi}{FFT L} nk}} \quad k = 0, ..., FFT L - 1$$

Due to the symmetry of $bin$ only $bin_{0, FFTL/2}$ are considered for the further processing.

After the FFT an ideal bandpass (180Hz-4000Hz) is applied.

![Figure 1: Block diagram of the front-end](image-url)
Two smoothed power spectra are calculated:

\[ |\hat{s}(f)|^2 = a_{\eta_2} |s(f)|^2 + (1 - a_{\eta_2}) |\eta(f)|^2 \]  

(9)

where \( |s(f)|^2, |\eta(f)|^2 \) are the short term power spectra of \( s(t) \) and \( \eta(t) \).

### 2.2. Spectral attenuation and spectral subtraction

The spectral attenuation as well as the spectral subtraction regard noise as an additive noise superposed on the undisturbed speech signal \( s(t) \):

\[ x(t) = s(t) + \eta(t) \]  

(2)

where \( x(t) \) is the signal of the noisy speech and the noise \( \eta(t) \) is regarded as statistically independent from the undisturbed speech \( s(t) \). In the spectral domain this leads to:

\[ |\tilde{s}(f)|^2 = |s(f)|^2 + |\eta(f)|^2 \]  

(3)

where \( |\tilde{s}(f)|^2, |s(f)|^2 \) and \( |\eta(f)|^2 \) are the short term power spectra of \( x(t), s(t) \) and \( \eta(t) \).

#### 2.2.1. Spectral attenuation

The spectral attenuation is based on a well-known family of speech enhancement algorithms, the so called ‘short-time spectral attenuation algorithms’ [2].

The estimation of the short-time spectrum of the speech \( \hat{s}(f) \) for a current frame \( n \) is obtained by applying an attenuation function \( G^n(f) \) to the spectrum of the noisy speech \( x^n(f) \):

\[ \hat{s}(f) = G^n(f) \cdot \tilde{x}(f) \]  

(4)

For \( G^n(f) \) a Wiener filter was used which can be expressed according to equation (5):

\[ G^n(f) = \frac{|\tilde{x}(f)|^2}{|\tilde{x}(f)|^2 + |\hat{\eta}(f)|^2} \]  

(5)

\( |\tilde{x}(f)|^2 \) and \( |\hat{\eta}(f)|^2 \) are computed through the first order IIR filtering expressed by:

\[ |\tilde{x}(f)|^2 = (1 - \lambda) \left( \lambda |\tilde{x}(f)|^2 + |\hat{\eta}(f)|^2 \right) \]  

(6)

\[ |\hat{\eta}(f)|^2 = (1 - \lambda) \left( \lambda |\tilde{x}(f)|^2 + |\tilde{\eta}(f)|^2 \right) \]  

(7)

where \( \lambda < 1 \) and \( \tilde{x}(f) \) and \( \tilde{\eta}(f) \) are continuous estimates of the short-time spectrum of the speech and of the noise.

#### 2.2.2. Spectral subtraction

According to (8) an estimate for the additive noise power spectrum has to be found which is subtracted from the power spectrum leading to a better estimate for the power spectrum of the original speech.

\[ |\tilde{\eta}(f)|^2 = |x(f)|^2 - |\hat{s}(f)|^2 \]  

(8)

Two smoothed power spectra are calculated:

\[ |\tilde{s}^2(f)|^2 = a_{\eta_1} |\tilde{x}^2(f)|^2 + (1 - a_{\eta_1}) |\hat{s}^2(f)|^2 \]  

(9)

#### 2.3. Cepstral filter analysis

The power spectrum is processed by a mel scaled filter bank where the spectral energies are smoothed by triangular functions. The frequency band is divided into \( J_{cep} \) channels which are spaced equidistantly with the same width \( \Delta_{mel} \) on the mel scale. The smoothing is implemented by a convolution of the power spectrum with a triangular filter in the mel domain.

\[ x^n_mel = \sum_{j=1}^{J_{cep}} x^n_mel \cos \frac{\pi_j}{2} \]  

(10)

with \( a_{\eta_1} \) and \( a_{\eta_2} < 1 \).

The power spectrum \( |\hat{\eta}(f)|^2 \) is estimated by the minima of the smoothed power spectrum \( |\tilde{s}^2(f)|^2 \) within a moving interval \( I^n \) with the fixed width \( D \) [3].

\[ |\hat{\eta}(f)|^2 = \text{Min}_{\text{mel}^n} |\tilde{s}^2(f)|^2 : I^n = [a - D, a] \]  

(11)

Introducing an over-subtraction factor \( \beta_{osub} \) [4] and a flooring factor \( \beta_\eta \) the power spectrum of \( |\tilde{s}^2(f)|^2 \) is estimated by

\[ |\tilde{s}^2(f)|^2 = \begin{cases} |x^n_mel| - \beta_{osub} \left( \frac{|x^n_mel|}{|\hat{\eta}(f)|^2} \right)^2 |\hat{\eta}(f)|^2 & \text{if } |x^n_mel|^2 < \beta_\eta |\hat{\eta}(f)|^2 \\ \beta_\eta |\hat{\eta}(f)|^2 & \text{if } |x^n_mel|^2 \geq \beta_\eta |\hat{\eta}(f)|^2 \end{cases} \]  

(12)

#### Table 2: Parameter settings for the triangular filters

| \( f_s \) | \( \Delta_{mel} \) | \( J_{cep} \) | \( K_{cep} \) |
| 8 kHz | 133.66 | 15 | 12 |

The smoothed power spectra \( x^n_mel \) are transformed into the mel-cepstral domain leading to \( K_{cep} \) cepstral coefficients:

\[ u_k^n = \sum_{j=1}^{J_{cep}} x^n_mel \cos \frac{\pi_j}{2} \]  

(13)

Finally the cepstral coefficients \( u_k^n \) are liftered:

\[ x_k^n = u_k^n (1 + 0.5 c_L \sin \frac{k \pi}{c_L}) ; c_L \in 2 \cdot K_{cep} ; k = 1, \ldots, K_{cep} \]  

(14)

#### 2.4. Channel compensation

In real world applications the spectrum of the speech signal is changed due to the different characteristics of the transmission channels. Commonly these transmission characteristics can be modeled by a time dependent transfer function \( H(f, t) \) which transforms the spectrum of the original speech signal \( s(t) \) into a modified spectrum \( \tilde{s}(f) \). Assuming that this model is also valid for the short time power spectrum of a frame and
assuming that \( H(f,t) \) does not change within a small number of frames \( \tilde{s}(f) \) can be written in the log space as:
\[
\log |H(f,t)|^2 = \log |H(f,t)|^2 + \log |H(f,t)|^2 = \log |H(f,t)|^2 \quad ; \quad i = 0, \ldots, \text{FFTL} / 2 \quad (15)
\]
This relation leads to the model of the features \( \tilde{x}^n \) which are a processed form of \( \tilde{f}(f)^2 \) as defined in \((14)\) having an offset on the ‘clean’ features \( x^n \) stemming from the term \( \log |H(f,t)|^2 \) in \((15)\). The offset \( x^n_{ko} \) is estimated by \( \tilde{x}^n_{ko} \) using a Maximum Likelihood Estimator leading to:
\[
\tilde{x}^n_{ko} = \tilde{x}^n_{ko} + x^n_k; \quad k = 1, \ldots, K_{cep}
\]
and thus to the offset free features:
\[
x^n_k = \tilde{x}^n_{ko} - \tilde{x}^n_{ko}; \quad k = 1, \ldots, K_{cep}
\]
The estimates \( \tilde{x}^n_{ko} \) for frame \( n \) are determined by the algorithm:\[5]\]
\[
\begin{align*}
\tilde{x}^n_0 &= 0 \\
\tilde{x}^n_k &= \frac{n-1}{2} \tilde{x}^{n-1}_k + \frac{1}{n} x^n_k; \\
\kappa &= \frac{1}{\sigma^2} \frac{1}{2} \left( \tilde{x}^n_{ko} + \left( n \tilde{x}^n_{ko} \right) \right); \\
\end{align*}
\]
where:
\( \sigma^2 \) denotes the variance of the samples \( x^n_k \)
\( \sigma^2_{ko} \) denotes the variance of the channel offsets \( x^n_{ko} \)
\( \tilde{x}_{ko} \) denotes the mean value of the channel offsets \( x^n_{ko} \)
\( k = 1, \ldots, K_{cep} \)

2.5. Context Modeling

In order to cover the context of the segments within a word the feature vector \( \tilde{X}^n \) consisting of the cepstral coefficients \((14)\) is augmented by its first and second derivatives with respect to time as well as with the energy value \( E^n \) and its first and second derivatives with respect to time. The first and second derivatives are built by:
\[
\Delta x^n_k = x^n_k - x^n_{k-\Delta t}
\]
\[
\Delta \Delta x^n_k = \Delta x^n_{k+\Delta t} - \Delta x^n_k
\]
where \( \Delta t = 3 \) and \( k = 1, \ldots, K_{cep} \).

The derivatives of the energy value are calculated accordingly.

2.6. Linear discriminant analysis (LDA)

For the input of the LDA two neighbouring augmented feature vectors are put together building a so-called super vector. The LDA transforms the input super vector of dimension 78 to an output feature vector of dimension 24 without loosing the discriminative potential of the features \([5]\).

Although the LDA is data dependent, the LDA employed here was not generated on the Aurora 2 database. This was done in order to follow the Aurora scheme using one front-end for different databases. To account for the unavoidable mismatch between test set and LDA, the LDA was generated during a training of a speaker independent whole-word HMM on the SpeechDat 2 German database.

2.7. Frame Dropping

The Aurora evaluation scheme \([1]\) does not allow any modifications of the recognizer parameters, which can lead to an unfavorably large number of insertion errors on the test set. This effect can be reduced by eliminating silence parts of the signal during feature extraction (Frame Dropping, FD). Here a criterion based on the power spectrum \( |f(f)|^2 \) before and after the spectral subtraction and on the noise power spectrum \( \rho(f) \) (see section 2.2.2) was chosen:

- If \( \rho^n f < w |f^n| \) for 7 succeeding time frames the skipping mode is turned on and the following time frames are not passed on to the recognizer.
- If \( \rho^n f > w |f^n| \) for one time frame the skipping mode is turned off again and the time frames are passed on to the recognizer. At the same time the two time frames preceding the time frame that fulfills the condition written above are as well passed on to the recognizer. They were before stored in a ring buffer.

Here \( \rho^n f \) is the mean value of the noise power spectrum of frame \( n \) and \( w \) is the weight which is calculated by:
\[
w = \frac{|f^n|^2}{|f|^2}
\]
where \( |f^n|^2 \) and \( |f|^2 \) denote the mean values of the short term power spectra of frame \( n \) before and after the spectral subtraction.

3. Results and discussion

3.1. Baseline recognizer setup

All recognition tests where conducted using the HTK speech recognition toolkit with the settings defined for the ETSI Aurora evaluations \([1]\). As an exception the calculation of the time derivatives by the HTK recognizer was turned off because it was done by the front-end.

3.2. Recognition Results

The detailed results of the performance of the front-end on the Aurora 2 database are shown in Table 3. Furthermore the results of the front-end without the frame dropping algorithm are presented in Table 4.

It can be seen that the recognition results in conditions with no or low noise are deteriorated whereas the recognition results in conditions with high noise are increased. In multi-condition training this leads to a small overall increase of the recognition results over the reference front-end of 1.94%.

Eurospeech 2001 - Scandinavia
clean training the overall recognition result is significantly increased by 43.90%.

Furthermore it can be seen from Tables 3 and 4 that the addition of the frame dropping algorithm could increase the recognition results in clean training from an overall relative performance of 32.51% without frame dropping to 43.90% with frame dropping. At the same time the overall relative performance in multicondition training is slightly deteriorated from 3.17% with frame dropping to 1.94% without it.

The effect of the deterioration of the results in low noisy conditions can probably be explained with the LDA which has not been optimised for the use together with the Aurora 2 database.

<table>
<thead>
<tr>
<th>Siemens front-end</th>
<th>Aurora 2 Multicondition Training - Results</th>
<th>Overall Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>Clean</td>
<td>Subway 95.64 96.16 95.59 95.16 96.16 95.59 95.16 95.64 96.16 95.59 95.16 95.64</td>
<td>Subway M 95.21 95.95 95.58 95.03</td>
</tr>
<tr>
<td>20 dB</td>
<td>94.81 95.38 95.16 95.38 95.00 95.29 95.18 95.32 95.31 94.15 93.58 94.54</td>
<td>-8,08%</td>
</tr>
<tr>
<td>15 dB</td>
<td>94.38 95.28 94.81 95.37 94.55 94.92 94.32 95.31 94.92 94.32 95.31 94.92</td>
<td>-11,98%</td>
</tr>
<tr>
<td>10 dB</td>
<td>91.46 93.02 93.11 91.24 92.21 91.56 92.23 92.76 92.16 92.18 91.31 91.69 91.53 90.26</td>
<td></td>
</tr>
<tr>
<td>9 dB</td>
<td>85.75 86.46 86.70 82.38 85.07 82.50 85.82 86.25 85.71 85.07 84.10 83.43 82.77 84.41</td>
<td></td>
</tr>
<tr>
<td>3 dB</td>
<td>70.25 62.79 70.38 64.73 67.04 60.82 68.96 69.84 68.56 66.97 65.56 65.54 65.55 66.01</td>
<td></td>
</tr>
<tr>
<td>5 dB</td>
<td>38.73 30.53 30.19 35.28 33.68 31.56 36.80 36.26 34.94 35.39 34.12 32.81 33.47 34.32</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>87.34 86.51 88.08 85.41 86.84 87.57 87.54 86.95 85.85 86.14 86.00 86.65</td>
<td>1.94%</td>
</tr>
</tbody>
</table>

Table 3: Results of the Siemens front-end on the Aurora 2 database

<table>
<thead>
<tr>
<th>Siemens front-end without frame dropping</th>
<th>Absolute performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training Mode</td>
<td>Set A</td>
</tr>
<tr>
<td>Multicondition</td>
<td>87.10 86.61 86.68 86.82</td>
</tr>
<tr>
<td>Clean Only</td>
<td>72.35 72.70 86.68 75.13</td>
</tr>
<tr>
<td>Average</td>
<td>79.73 79.86 86.68 80.57</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Siemens front-end without frame dropping</th>
<th>Performance relative to Mel-cepstrum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training Mode</td>
<td>Set A</td>
</tr>
<tr>
<td>Multicondition</td>
<td>-5.88% 2.49% 17.91% 3.17%</td>
</tr>
<tr>
<td>Clean Only</td>
<td>28.48% 38.31% 26.54% 32.51%</td>
</tr>
<tr>
<td>Average</td>
<td>11.30% 20.40% 22.23% 17.84%</td>
</tr>
</tbody>
</table>

Table 4: Results of the Siemens front-end without frame dropping on the Aurora 2 database

4. Conclusion

The performance of a front-end, developed by Siemens, on the Aurora 2 database was presented in this paper. The recognition results in conditions with low mismatch between training and testing data (Multicondition Training) are deteriorated relative to the reference front-end especially when no or low noise is present and are increased when high noise is present.

In conditions with a high mismatch between training and testing data (Clean Training) the recognition results are significantly increased especially when noise is present. The weak results in conditions with no noise are probably due to the LDA which has not been optimised for its use here. Also the way the different algorithms work together has not fully been investigated yet. Results of an optimised front-end are expected for the Web-only version of this paper [6].

5. References