

LARGE VOCABULARY NOISE ROBUSTNESS ON AURORA4

Luca Rigazio, Patrick Nguyen, David Kryze, Jean-Claude Junqua

Panasonic Speech Technology Laboratory
3888 State Street, Santa Barbara, CA 93105
{rigazio, nguyen, kryze, jcj}@research.panasonic.com

ABSTRACT

This paper presents experiments of noise robust ASR on the Aurora4 database. The database is designed to test large vocabulary systems in presence of noise and channel distortions. A number of different model-based and signal-based noise robustness techniques have been tested. Results show that it is difficult to design a technique that is superior in every condition. Because of this we combined different techniques to improve results. Best results have been obtained when short time compensation / normalization methods are combined with long term environmental adaptation and robust acoustic models. The best average error rate obtained over the 52 conditions is 30.8%. This represents a 40% relative improvement compared to the baseline results [1].

1. INTRODUCTION

The Aurora4 database has been deployed in the context of the ETSI/OTC standardization process to evaluate performance of Large Vocabulary Speech Recognition (LVCSR) in presence of noise. The database is based on the SI84 Wall Street Journal database for training, and on the Nov'92 5000 words evaluation set for testing. The training set is available in two different versions: clean condition training set, and multi-condition training set, and in two different sampling rates, 16kHz and 8kHz. The clean condition training set is identical to the SI84 training set (downsampled for the 8kHz case) with 84 speakers for a total of 12 hours of speech. The multi-condition training set has a variety of noises added to it, and mixes data from the Sennheiser close talking microphone and from several far talking microphones. The test set is split in two main conditions, one based on the Sennheiser microphone and the other based on the far talking microphones. Six noises are added at test time, to create a total of 14 testing conditions (including clean conditions). The second microphone condition is very challenging due to the low bandwidth of some of the recordings.

2. NOISE ROBUSTNESS

We will briefly review some of the robustness methods tested. In particular we will focus on the methods that provided a clear and consistent improvement. In section 6 we will also discuss about other methods that did not perform as expected on Aurora4. Due to the complexity of LVCSR in such degraded conditions, we will try several techniques to improve results. These include robust front-end analysis, model compensation, long term adaptation and accurate acoustic modeling. The best results are obtained with a combination of those methods. However not all combinations are

possible. For instance, our noise robust front-end is not compatible model compensation. Also model compensation is difficult to use with multi-condition / long-term adapted acoustic models.

2.1. Robust front-end analysis

The robust front-end used in the experiments is based on the subband analysis presented in [2]. The analysis performs a bi-orthogonal wavelet transform to achieve a hierarchical time-frequency decomposition of the speech signal. The wavelet analysis is computed recursively by decomposing the signal into its low-pass and high-pass components. The recursion results in a cascade of high-pass and low-pass filters followed by down-sampling. The structure of the tree and the filter coefficients define the wavelet basis onto which the signal is projected as well as the time-frequency resolution of the analysis (Figure 1).

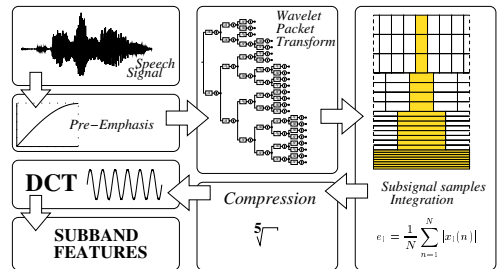


Figure 1: Subband features extraction scheme

For each subband l , we compute the average energy e_l of the leaf filters outputs $x_l(n)$:

$$e_l = \frac{1}{N} \sum_{n=1}^N |x_l(n)|^2, \quad l = 1, 2, \dots, L$$

This average is computed over the same number N of coefficients for each subband to provide a hierarchical time resolution. The size of the larger averaging window (corresponding to the lowest subband) is called the *window size*. The averaging window is then shifted by the size of a *frame* to deliver subband amplitudes at frame-rate. Here we used a frame rate of 100 Hz, and a window size of 32 milliseconds.

The local energy $|x_l(n)|^2$ is estimated with the Teager Energy Operator [3]:

$$\Psi[x_l(n)] = x_l^2(n) - x_l(n+1)x_l(n-1)$$

The modulation energy tracking capability of the Teager Energy Operator has been shown provide robustness to stationary additive noise [4], [2].

Thirteen cepstral coefficients are computed by applying a non-linear compression operator to the amplitudes e_l , followed by a discrete cosine transform. A fifth root-compression operator is used for increased robustness to additive noise. The first cepstral coefficient c_0 is not included in the feature set and is replaced by the log-energy. First, second and third derivatives are computed, to form a vector of 52 dimensions. Cepstral mean normalization is then performed over a sliding window of 200 frames centered around the current frame.

2.2. Model compensation

The model compensation algorithm used is based on the simple version of PMC [5] that adapts only the means of the acoustic models. Model compensation is applied to a standard MFCC front-end, with static, first and second derivatives, and no cepstral mean normalization. Indeed it is very difficult, if not impossible, to derive compensation formulas for heavily processed noise robust front-ends, like the one described in section 2.1. This is because model compensation computes

$$C(S + N) = C(C^{-1}(C(S)) + C^{-1}(C(N))), \quad (1)$$

where X is a spectral vector and $C(X)$ is the cepstral operator. For equation 1 to be computed, $C(X)$ needs to be invertible. Simple operations commonly used to improve front-end robustness, such as CMN, will violate this condition, thus preventing us from compensate the resulting acoustic models.

However, if $C(X) = F \log(X)$ is the cepstral operator, where F is the DCT matrix and $\log(X)$ a component-wise logarithm, noise compensation formulas may be derived for both static and dynamic features [5].

In our implementation we neglected all second order statistics in the compensation of the second derivatives. This is because second order statistics are difficult to estimate and unreliable. Also, we perform up-sampling during the log-energy reconstruction step $\log(X) = F^{-1}C(X)$. This is done by using an inverse DCT matrix larger than the original matrix used in the front-end. Specifically, the MFCC front-end has 20 filter-bank outputs and 13 cepstral coefficients and we use 64 log-spectral energies for noise compensation. We discovered that the up-sampling provides consistent performance improvements. We argue that this may be due to reduced quantization errors during the compensation.

To achieve good performance for multi-condition training, we have to take into account that the acoustic models are also modeling some noise. This idea is similar to that used in the Jacobian adaptation [6] [7]. We will assume that the acoustic models are estimated at a reference noise level N_R and have to be compensated to the target noise level N_T . This will modify the PMC mismatch function in:

$$C(S + N_T) = C(C^{-1}(C(S + N_R)) + \Delta N), \quad (2)$$

where $\Delta N = \alpha(N_R - N_T)$, and α is a noise underestimation factor. Compensation formulas for static and dynamic coefficients were modified according to this mismatch-function.

In our experiments the target noise is estimated during the first 250 milliseconds of each sentence. The reference noise is estimated from the Gaussian mixture associated to the central state of

the silence model with:

$$N_R = \sum_k w_k C^{-1}(\mu_k), \quad (3)$$

where w_k, μ_k are the Gaussian mixture weights and the Gaussian means. We observed that the optimal noise underestimation factor, was close to 1 for clean conditions models, and below 0.5 for multi-condition models.

3. ACCURATE ACOUSTIC MODELING

Context dependent, tree clustered, word internal triphone models were trained for each of the four training conditions in the Aurora4 training database. A constant likelihood threshold and occupation probability was used for the tree growing. This provided a different number of leaves for the four training conditions, ranging between 800 to 1500. Acoustic models complexity was increased to 64 distribution per state. Then the total number of gaussians was reduced to 32k with a minimum likelihood loss agglomerative clustering. For the acoustic models based on the subband front-end, maximum-likelihood feature transformation (section 3.1) was applied at training time to achieve a higher degree of feature-decorrelation. This provided an average improvement of 1.5%. For the acoustic models based on the MFCC front-end MMIE was applied, and provided an average improvement of 0.9% (Tables 4 and 5). Performance in clean conditions / 16kHz is 5.3%, close to state-of-the art reported in the literature.

3.1. MLLU based feature transformation

Maximum-Likelihood feature transformations can be used to adapt features in a noisy environment, or to decorrelate features. In the first setting, they compensate for mismatch by adapting both means and variances. In the second setting, they provide better features that are more robust to noise on individual cepstrum coefficients. We transformed the feature space using a linear maximum-likelihood linear transformation. Throughout this paper, we have used a closed-form solution for the iterative LU factorization of the feature transform [8].

We transform the feature vectors o_t with $o'_t = A o_t$. It can be shown that maximizing the following auxiliary Q function increases the likelihood:

$$Q = -\frac{1}{2} \sum_{t,m} \gamma_m(t) \left\{ -\log |A|^2 + (\mu_m - A o_t)^T C_m^{-1} (\mu_m - A o_t) \right\},$$

where $\gamma_m(t)$ are the state posteriors for all Gaussians $\mathcal{N}(\mu_m, C_m)$, and o_t are the observation vectors. It can also be shown that transforming the features is equivalent to transforming both means and variances with:

$$\begin{aligned} \mu_m &\leftarrow A \mu, \\ C_m &\leftarrow A C_m A^T. \end{aligned}$$

Adapting variances is an essential component in noisy conditions. However, in our experiments, MLLU long-term adaptation has not outperformed MLLR long-term adaptation. Instead, MLLU has provided a significant improvement when applied at training time.

3.2. Robust discriminative models

While porting models to noisy environments, the distortion of parameters augments the confusability of models, and therefore creates decoding errors. Discriminative models are able to separate better between classes. They are more suitable both in clean and noisy conditions. In this paper, we have used the maximum-mutual information (MMI) [9] criterion to devise models. The MMI criterion is:

$$\hat{\lambda} = \arg \max_{\lambda} \frac{p(O, w|\lambda)}{\sum_v p(O, v|\lambda)}$$

where $\lambda = \mu$ is a model parameter. The word sequence w is the true transcription. For efficient computations we integrate the expectations on a state lattice.

4. LONG TERM ADAPTATION

Long term unsupervised adaptation was used to adapt to the test condition. Best results were obtained with MLLR adaptation [10]:

$$\mu_{MLLR} = \left[\arg \max_W p(O|W\mu_0) \right] \mu_0; \quad (4)$$

In the previous equations μ_0 is the speaker independent mean, W is the regression matrix, $p(O|\mu)$ is the likelihood.

For each testing condition, a single regression matrix was estimated on the recognition lattices produced with the unadapted models. Only one iteration of EM was performed on the lattices to derive the matrix. Adapted models were then used for a second recognition pass.

5. EXPERIMENTS

Experiments were conducted on the large evaluation set of the Aurora4 database. The set has 330 sentences for each testing condition. A small evaluation set of 166 sentences is also defined for the database; however, we decided to use the large evaluation set to improve the significance of our results. We have fourteen testing conditions times four training conditions for a total of 56 recognition results per experiment. For reasons of space and for ease of reading, we decided to report the results only by microphone type and training condition.

In table 1 we show the baseline results, obtained by Mississippi State University (MSU) and published with the database [1]. Results are reported in word error rates (WER). In table 2 we report results obtained with the noise robust subband front-end described in section 2.1. Notice that acoustic models for this front-end were trained with MLLU feature decorrelation. Clean condition training error rates double for the 16kHz models between the first and the second microphone. Also notice that results for the second microphone are almost identical for 8kHz and 16kHz models. This shows that the second microphone poses a clear challenge to our system. We checked some data of the second microphone and discovered that some signals were originally recorded with very narrow band (possibly 4kHz), and then noise was added to the full-band (8kHz). This type of mismatch is so severe that the 16kHz models give lower performance than the 8kHz models on some testing conditions of the second microphone. This unlikely effect happens in spite of the fact that the front-end is compensating for the channel with CMN. We believe that this is a strong artifact of this database. This problem is even more evident after

Train	Freq	Mic1	Mic2	Avg
Clean	16k	61.0%	78.5%	69.8%
Clean	8k	52.6%	63.3%	57.9%
Multi	16k	30.2%	49.0%	39.6%
Multi	8k	33.9%	43.7%	38.8%
Avg	Avg	44.4%	58.6%	51.5%

Table 1: MSU Baseline (WER)

long-time adaptation is carried out (Table 3). Results for 8kHz / clean training / second microphone test are more than 5% better than results of 16kHz models. The problem eventually disappears for multi-condition training, due to the fact that the acoustic models have seen data from the second microphone at training time.

Finally we can observe that results for multi-condition training are about 10% better than results from clean training for unadapted models. Long-term adaptation clearly improved 8kHz / clean, but did not perform as well for 16kHz or for multi-condition training. The total improvement due to long-term adaptation is of 1.7% (5% absolute).

Train	Freq	Mic1	Mic2	Avg
Clean	16k	22.7%	46.8%	34.7%
Clean	8k	34.5%	46.7%	40.6%
Multi	16k	17.2%	32.5%	24.9%
Multi	8k	25.1%	34.3%	29.7%
Avg	Avg	24.9%	40.1%	32.5%

Table 2: Subband front-end (WER)

Train	Freq	Mic1	Mic2	Avg
Clean	16k	20.4%	46.2%	33.3%
Clean	8k	28.4%	41.1%	34.7%
Multi	16k	17.3%	33.9%	25.6%
Multi	8k	25.0%	34.3%	29.7%
Avg	Avg	22.8%	38.9%	30.8%

Table 3: Subband front-end with long term adaptation (WER)

In table 4 we report results obtained with the MFCC front-end and the PMC model compensation described in section 2.2. The noise underestimation factor α was empirically set to 0.8 for clean condition models, and to 0.2 for multi-condition models. Notice that acoustic models for this front-end were not trained with MLLU feature decorrelation. Comparing results from tables 4 and 2 we notice that the MFCC/PMC system outperformed the Subband system by 0.7%. The MFCC/PMC system clearly outperformed the Subband system for 16kHz. Results for 16kHz / clean were expected to be improved by model compensation, however also results for 16kHz / multi-condition improved by about 2.5%. Unfortunately, the MFCC/PMC system did not perform as well for the 8kHz conditions. Specifically 8kHz / multi-condition results are 5% lower than results obtained with Subband. The cause of this problem has not been found, but its influence on narrow-band results may warrant further search. Table 5 reports results with the

MFCC/PMC system when the acoustic models are trained with 6 iterations of MMIE. Lattices were generated for the clean / 16kHz condition, and were used for training in all conditions. This may be the reason why results improved more for the clean / 16kHz condition. Finally we notice that the MFCC/PMC system is superior to the unadapted Subband system but is comparable to the long-term adapted Subband system.

Train	Freq	Mic1	Mic2	Avg
Clean	16k	19.4%	41.8%	30.6%
Clean	8k	35.2%	45.5%	40.3%
Multi	16k	14.4%	29.2%	21.8%
Multi	8k	30.6%	38.8%	34.7%
Avg	Avg	24.9%	38.8%	31.8%

Table 4: MFCC front-end with model compensation (WER)

Train	Freq	Mic1	Mic2	Avg
Clean	16k	17.5%	39.5%	28.5%
Clean	8k	34.5%	45.7%	40.1%
Multi	16k	13.5%	28.6%	21.1%
Multi	8k	29.3%	38.9%	34.1%
Avg	Avg	23.7%	38.2%	30.9%

Table 5: MFCC front-end with model compensation and MMIE trained acoustic models (WER)

6. DISCUSSION

Some well-known noise-robustness techniques were also tested but did not perform as expected. For the Subband front-end only the mean of the static coefficients is normalized. Variance normalization has also been tried but resulted in a consistent degradation. Histogram equalization has also been tested but resulted in a degradation. Spectral subtraction did provide some improvement for the Subband front-end on stationary noise conditions, but resulted in an overall degradation. The same problem was observed for Jacobian adaptation and MMLU long-term adaptation. Moreover our noise robust front-end developed for Aurora2 did not perform as well on Aurora4.

Regarding the two main approaches of model compensation and noise robust front-end, we notice that in practice the model compensation requires an estimate of the background noise to be transmitted, whereas a noise robust front-end does not need it. Moreover, if model compensation has to be used in the back-end, the front-end needs to be close to a complete reconstruction analysis. This poses limitations on the amount of processing that can be applied to cepstral features before being transmitted (CMN for instance does not allow for PMC model compensation). Finally, results from the two approaches are comparable, especially when long-term adaptation is performed.

7. CONCLUSIONS

Noise robustness for LVCSR is a challenging problem. This paper presents our experience with the Aurora4 database. A multitude of

techniques have been applied to improve noise robustness. In particular the two main approaches of model compensation and robust front-ends are compared. Results show that it is difficult to design a method that is superior in every condition; however, significant improvements can be achieved by combining different techniques. In particular compensation / normalization methods can be combined with long term environmental adaptation and robust acoustic models for improved results. The best average error rate obtained is 30.8%, and represents a 40% relative improvement compared to the baseline results [1].

8. REFERENCES

- [1] Gunter Hirsch, "Experimental framework for the performance evaluation of speech recognition front-ends on a large vocabulary task," in *STQ Aurora DSR Working Group*, 2002.
- [2] D. Kryze, L. Rigazio, T. Applebaum, and J.-C. Junqua, "A New Noise-robust Subband Front-end And Its Comparison To PLP," in *Proc. of ASRU*, Keystone, CO, Dec. 1999.
- [3] James F. Kaiser, "On a simple algorithm to calculate the energy of a signal," in *Proc. of ICASSP*, 1990, pp. 381–384.
- [4] F. Jabloun and A. E. Cetin, "The Teager Energy Based Feature Parameters for Robust Speech Recognition in Car Noise," in *Proc. of ICASSP*, Phoenix, AZ, USA, April 1999, vol. 1.
- [5] M. Gales, "Predictive model-based compensation schemes for robust speech recognition," *Speech Communication*, vol. 25, pp. 49–74, 1998.
- [6] S. Sagayama, Y. Yamaguchi, and S. Takahashi, "Jacobian Adaptation of Noisy Speech Models," in *Proc. of ASRU*, Santa Barbara, CA, Dec. 1997, pp. 396–403.
- [7] C. Cerisara, L. Rigazio, R. Boman, and J.-C. Junqua, "Environmental adaptation based on first order approximation," in *Proc. of ICASSP*, 2001.
- [8] P. Nguyen, L. Rigazio, C. Wellekens, and J.-C. Junqua, "LU Factorization for Feature Transformation," in *Proc. of IC-SLP*, Boulder, USA, 2002, pp. 73–76.
- [9] P. C. Woodland and D. Povey, "Large scale discriminative training for speech recognition," in *ISCA ITRW Automatic Speech Recognition: Challenges for the Millenium*, Paris, 2000, pp. 7–16.
- [10] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaption of continuous density hidden Markov models," *Computer Speech and Language*, vol. 9, pp. 171–185, 1995.