EVALUATION OF A NOISE ADAPTIVE SPEECH RECOGNITION SYSTEM ON THE AURORA 3 DATABASE

Kaisheng Yao⋆, Dong-Lai Zhu†⋆, and Satoshi Nakamura⋆

⋆ATR Spoken Language Translation Research Laboratories, Kyoto, Japan
kaisheng.yao@atr.co.jp xdzhu@slt.atr.co.jp satoshi.nakamura@atr.co.jp

†Internship student from the Department of Electronic Engineering and Information Science, University of Science and Technology of China.

ABSTRACT
In this paper, we present evaluation results of a noise adaptive speech recognition system with combination of several techniques for robust speech recognition. The evaluation was on AURORA 3 database which contains noisy digit utterances collected in real car environments through close-talking and hands-free microphones. The techniques in the system include segmentation, maximum likelihood linear regression (MLLR) and non-stationary environment compensation by noise adaptive speech recognition. Through experiments, it is observed that the system has competitive performance improvement in all evaluations over the baseline results provided for the evaluation. As a whole, the system achieved 28% of relative performance improvement.

1. INTRODUCTION
Speech recognition has to be carried out often in situations where there exists environment distortions, such as channel distortion, background noise, competing speech and room reverberation, which cause mismatches between pre-trained models and real testing data. A huge variety of algorithms has been proposed to enhance system environment robustness. The approaches range from signal processing front-end, robust feature extraction, to back-end model adaptation. The AURORA task [1] is available as a framework to compare different algorithms and systems on a common base. In particular, AURORA 3 is a subset of the SpeechDat-Car (SDC) corpus collected in cars with different driving conditions, e.g., High-speed, Low-speed, and various placing configurations, such as climate control on/off, etc. Four European languages, Spanish, Finnish, German, Danish, of continuous digits utterances were collected in cars through close-talking microphone and hands-free microphones. In all of the tested languages, three sets of evaluations are provided. The Well-matched (WM) evaluation has training and testing set from utterances through both microphone types and all driving conditions. The Medium-mismatched (MM) evaluation utilizes training data from hands-free microphones using all driving conditions except for the High Speed driving condition. The testing set has data from hands-free microphones and the High Speed driving condition only. High-mismatched (HM) evaluation utilizes data from close-talking microphones and all driving conditions. The testing set in the evaluation has data through hands-free microphones and all driving conditions except the Stopped Motor driving condition.

We follow the strategy of the AURORA by keeping the HMM structure the same as that provided by AURORA consortium. A system with combination of several techniques was evaluated on the AURORA 3 in its whole processing chain from feature extraction to back-end recognizer.

2. DIAGRAM OF THE SYSTEM

![Diagram of the robust speech recognition system.](image)

Fig. 1. Diagram of the robust speech recognition system.

The diagram of the robust speech recognition system is shown in Figure 1. Feature generation module extracts features for segmentation and recognition. Since the noise frames after feature extraction may cause recognition error in the recognition stage, a segmentation module is applied after feature extraction to remove some noise segments. The remained segments are for training acoustic models \( \lambda_S = \{ \lambda_{v_m} \} \) and testing. Each unit model \( \lambda_{v_m} \) is CDHMM with state transition probability \( \alpha_{v_m} \) and each state \( i \) has probability density as mixture of Gaussian \( b_k(\cdot) \) with weight \( \pi_{ik} \). Acoustic models are adapted by MLLR in a supervised way, and speech recognition is carried out by noise adaptive speech recognition, carrying out non-stationary environment parameter estimation, sequential acoustic model adaptation and speech recognition jointly.

3. NOISE ROBUST TECHNIQUES

3.1. Feature generation, segmentation and model training

3.1.1. Feature generation - median MFCCs
The feature used in this work was a modified MFCC + C0 and its first- and second-order coefficients. Feature dimension is denoted
as $N$. Feature generation procedures are normal except for a median filtering introduced after FFT. The median filter was added in each frequency bin after FFT to filter the sequence of linear spectral amplitude coefficients along frame index. The median of the input sequence was extracted in order to smooth the sequence. The filter length was set according to the spectral frequency. The higher the frequency, the longer the median filter length. In this work, the longest filter length was 10 frames and the shortest filter length was 3 frames. We denote the feature as median MFCCs.

3.1.2. Segmentation for training and recognition

Data collected in AURORA 3 databases has long background segments in the beginning and end of utterances. Noises in the segments may cause insertion error if the background noise is strong and non-stationary. Noises can also cause errors in segments between speech events. The segmentation module introduced in this work was to remove some noise segments in order to improve system robustness to noise.

Energy-based segmentation may fail the objective in non-stationary noises which are seen in the database. Information from speech models can be useful so that a two-class classifier for segmentation was devised. The two classes were speech and non-speech respectively. The speech class refers to the frames of the signal corresponding to speech, and the non-speech class refers to the frames of background noises. General speech and non-speech models were GMMs, trained for each language from the training utterances in Well-matched evaluations, which include all training and testing conditions. Training transcripts were modified to label all digit events as “speech” and others as “non-speech”.

For this two-class classification, a log-likelihood ratio was calculated at each frame $t$ for observation vector $y(t) \in R^{N \times 1}$ by

$$ D(y(t)) = \log \frac{P(y(t)|{\text{Speech}})}{P(y(t)|{\text{Non-speech}})} $$

The log-likelihood ratio was further filtered by a median filter with filter length of five.

Noisy segments in each utterance could possibly identified and removed by the following way. 1) Set $\alpha$ as the percentage of the frames that might belong to non-speech class, and set the minimum length $M$ that a segment to be classified as non-speech. 2) Calculate the histogram of $D(y(t))$ for the utterance. 3) From the smallest $D(y(t))$, count to $\alpha$ percentage of the histogram, and set the $D(y(t))$ at the count as the threshold. 4) Remove those frames which continuously have log-likelihood ratio $D(y(t))$ below the threshold for at least $M$ frames.

The segmentation will remove some noise segments. Since in this situation, the cost of identifying a noise segment as speech was smaller than identifying speech frames as belonging to noise segment, the $\alpha$ was normally set smaller than 50%.

3.2. Recognition network

Since the beginning and end of each utterance for recognition are background signal, we made the recognition network to begin from silence node and to end at silence node too.

3.3. MLLR for linear adaptation of acoustic models

Before recognition on the testing set, acoustic models $\Lambda$ were adapted to the testing environments in order to decrease the mismatch between trained models and testing environments. We used MLLR [2] for linear transformation of the mean vectors of HMMs. A global MLLR transformation was used, which adapted the mean vector $\mu_{ik} \in R^{N \times 1}$ in mixture $k$ of state $i$ by

$$ \tilde{\mu}_{ik} = A_i \mu_{ik} + b $$

Given observation sequence $Y(T) = (y(1), y(2), \ldots, y(T))$, EM algorithm for maximum likelihood estimation of the transformation matrix $A \in R^{N \times N}$ and bias vector $b \in R^{N \times 1}$ arrives at solving the following equation,

$$ \sum_{t=1}^{T} \sum_{i=1}^{N} \sum_{k=1}^{N} \gamma_{ik}(t) \Sigma_{ik}^{-1} y(t) \xi_{ik}^T = 0 $$

$$ \sum_{t=1}^{T} \sum_{i=1}^{N} \sum_{k=1}^{N} \gamma_{ik}(t) \Sigma_{ik}^{-1} W \xi_{ik} \xi_{ik}^T $$

where $\Sigma_{ik}$ is the diagonal covariance matrix. The extended variable $\xi_{ik} \in R^{(N+1) \times 1}$ and $W \in R^{N \times (N+1)}$ are constructed as

$$ \xi_{ik} = [1 \mu_{ik}]^T $$

$$ W = [b A] $$

where $\gamma_{ik}(t) = P(s(t) = i, k(t) = k) | Y(T), \Lambda_X)$ denotes the posterior probability at state $i$ and mixture $k$ given the observation sequence and acoustic models. Superscript $T$ denotes transpose operation.

In this work, block diagonal matrix $A$ was used where each sub-matrix for static, first- and second-order coefficients was full.

3.4. Noise adaptive speech recognition

Noise adaptive speech recognition was proposed to do speech recognition in non-stationary environments [3]. It includes iterative processes between un-supervised sequential environment parameter estimation and acoustic model adaptation. The method makes use of a model for channel distortion and noise effects on speech features [4],

$$ y'_{ik}(t) = x'_{ik}(t) + h_{ik}(t) + \log(1 + \exp(n'(t) - x'_{ik}(t) - h_{ik}(t))) $$

where $y'_{ik}(t)$, $x'_{ik}(t)$, $h_{ik}(t)$ and $n'(t)$ each denotes the noisy observation, speech, channel distortion and additive noise at frame $t$. Their dimensions are decided by number of filter banks $J$. Superscript $t$ denotes that they are in log-spectral domain.

Denote $\Lambda_X(t) = (\lambda_X(1), \lambda_X(2), \ldots, \lambda_X(t))$ as environment parameter sequence for channel $h_{ik}(t)$ and noise $n'(t)$. Given the current observation sequence $Y(T) = (y(1), y(2), \ldots, y(t))$ till frame $t$ and the previous environment parameter sequence $\Lambda_X(t-1)$, the noise parameter estimation procedure will find $\lambda_X(t)$ as the current environment parameter estimate, according to the following objective function.

$$ F_t(\lambda_X(t)) = Q_t(\lambda_X(t); \lambda_X(t-1)) - (\beta_t - 1) \sum_{S(t)} \log \frac{P(S(t)|Y(t), \Lambda_X, (\lambda_X(t-1), \lambda_X(t-1)))}{P(S(t)|Y(t), \Lambda_X, (\lambda_X(t-1), \lambda_X(t)))} $$

$$ P(S(t)|Y(t), \Lambda_X, (\lambda_X(t-1), \lambda_X(t-1))) $$

where $\eta_{ik}(t) = P(s(t) = i, k(t) = k) | Y(T), \Lambda_X)$
where $Q_t(\lambda^*_N(t); \hat{\lambda}_N(t))$ is the auxiliary function in sequential EM algorithm, given as,

$$Q_t(\lambda^*_N(t); \hat{\lambda}_N(t)) = \sum_{\tau=1}^{t} \rho^{t-\tau} \sum_{S(\tau)} P(S(\tau)|Y(\tau), \Lambda_X, (\lambda_N(\tau-1), \lambda_N^*(\tau))) \log P(Y(\tau), S(\tau)|\Lambda_X, (\lambda_N(\tau-1), \lambda_N^*(\tau))) / P(S(\tau)|Y(\tau), \Lambda_X, (\lambda_N(\tau-1), \lambda_N^*(\tau))) \quad (8)$$

The second term in the right of (7) works as a regularization term. $S(t) = ((s(1), k(1)), (s(2), k(2)), \ldots, (s(t), k(t)))$ is the state and mixture sequence till frame $t$. Forgetting factor $\rho$ is normally smaller than 1.0, $\beta_t \in R^+$ works as a relaxation factor. $\lambda_N^*(t)$ is initialized to be $\lambda_N(t-1)$.

Estimation is carried out by iterations between the procedure to calculate the posterior probabilities $P(S(t)|Y(t), \Lambda_X, (\lambda_N(t-1), \lambda_N^*(t)))$, and maximization of the objective function to obtain $\hat{\lambda}_N(t)$. For each iteration, estimated $\hat{\lambda}_N(t)$ is for initialization of $\lambda_N^*(t)$ in the next iteration. The time recursive procedure is the sequential Kullback proximal algorithm [5], which is a generalization of the sequential EM algorithm. The sequential EM algorithm is a special case of this algorithm and corresponds to setting $\beta_t$ equal to 1.0 in the algorithm. The algorithm can achieve faster parameter estimation than that by sequential EM algorithm.

The joint likelihood of observation sequence $Y(t)$ and state/mixture sequence $S(t)$ can be approximately obtained from the Viterbi process, i.e.,

$$P(Y(t), S(t)|\Lambda_X, \lambda_N(t)) \approx a_{s^{*}(t-1), s^{*}(t)} \pi_{s^{*}(t), k(t)} \quad (9)$$

$$b_{s^{*}(t), y(t)} P(Y(t-1), s^{*}(t-1)|\Lambda_X, \lambda_N(t-1))$$

where the previous state is decided by

$$s^{*}(t-1) = \arg \max_{s(t-1)} a_{s(t-1), s(t)} P(Y(t-1), s(t-1)|\Lambda_X, \lambda_N(t-1)) \quad (10)$$

By normalizing the joint likelihood with respect to the sum of those from all active partial state sequences, an approximation of the posterior probability of state/mixture sequence can be obtained.

In particular, the environment parameter $\lambda_N(t)$ denotes the mean vector of time-varying channel distortion $\mu^k_h(t)$ and the mean vector of the time-varying noise $\mu^l_h(t)$.

Once the environment parameters have been estimated, the model adaptation at each frame is carried out by the following function on the static mean vector $\mu^k_h(t)$ in each mixture $k$ of state $i$ in acoustic models. That is,

$$\mu^k_h(t) = \mu^k_h(t) + \rho^i \mu^l_h(t) + \log(1 + \exp(\mu^l_h(t) - \mu^k_h(t))) \quad (11)$$

Note that (11) is an approximation to (6) with the assumption that the “channel” $h(t)$ and “noise” $l(t)$ have very small variance. The diagram of the noise adaptive speech recognition is shown in Figure 2. Readers please refer to [3] for details of the approach.

### 4. EXPERIMENTS

#### 4.1. Experimental setup

In the feature generation stage, pre-emphasis factor was 0.97. Window size for FFT was 25 ms, and time-shift was 10 ms. The number

![Fig. 2. Diagram of the noise adaptive speech recognition. $\Lambda_X$, $\Lambda_N(t)$ and $\Lambda_Y(t)$ are the original acoustic model, environment model at frame $t$, and adapted acoustic model at frame $t$, respectively. $Y(t)$ is the input noisy speech observation sequence till frame $t$. Recognition module provides approximated posterior probabilities of state/mixture sequences given noisy observation sequences till frame $t$ to the noise parameter estimation module, which outputs $\Lambda_N(t)$ to adapt acoustic model $\Lambda_X$ to $\Lambda_Y(t)$.](image)

<table>
<thead>
<tr>
<th>Table 1. Segmentation parameter for the evaluation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finnish</td>
</tr>
<tr>
<td>$\alpha$</td>
</tr>
<tr>
<td>$M$</td>
</tr>
</tbody>
</table>

GMMs for segmentation had 32 Gaussian mixtures. Segmentation was performed on each language with different parameters. Table 1 shows the parameter set empirically for segmentation of utterances in each evaluation. Segmentation could be optional. In particular, segmentation did not perform on Spanish subset.

The HMM back-end was defined by AURORA 3 task as 18 states with 3 Gaussian mixtures in each state for speech models, and five state HMM with 6 mixtures in each state for silence model. A three state tee model with six Gaussian mixtures in each state was used to model short-pause between speech events.

In each evaluation set, MLLR was supervised with 22 adaptation utterances randomly selected in the testing set. Noise adaptive speech recognition took one iteration at each frame. It had forgetting factor $\rho = 0.995$ and relaxation factor $\beta_t = 0.95$. At the beginning for each evaluation, channel parameter $\mu^k_h(t)$ and noise parameter $\mu^l_h(t)$ were initialized to be zero vector. The noise adaptive speech recognition was with beam-width of 2,000 for German, and 300 for other languages.
4.2. Development evaluation

We report in this section experiments on the evaluation of each noise robust module on experiments in Finnish set of the database. The baseline denotes system provided by AURORA 3 trained on normal MFCC + C0 and its first- and second-order coefficients. Performances by incremental adding each module are shown in Table 2. It is seen that adding Median filtering stage could decrease word error rate (WER) from 64.35% of the baseline in High-mismatched evaluation to 53.35%, though the error rate in Well-matched evaluation was slightly increased. Further adding segmentation module could decrease WER down to 42.26% in the High-mismatched evaluation. Using the recognition network shown in Section 3.2 could further decrease WER to 31.21%. Though adding noise adaptive speech recognition after MLLR slightly increased WER in the High-mismatched condition over that without noise adaptive speech recognition, it decreased WER in Well-matched and Medium-mismatched evaluation. For example, in Medium-mismatched evaluation, the WER decreased from 21.95% after MLLR to 16.65% by applying noise adaptive speech recognition. This trend of performance improvements is similar for other evaluated languages.

<table>
<thead>
<tr>
<th>Table 2. Word error rate (WER) of the Finnish set.</th>
</tr>
</thead>
<tbody>
<tr>
<td>WM</td>
</tr>
<tr>
<td>HTK baseline</td>
</tr>
<tr>
<td>+ Median filter</td>
</tr>
<tr>
<td>+ Segmentation</td>
</tr>
<tr>
<td>+ Recognition network</td>
</tr>
<tr>
<td>+ MLLR</td>
</tr>
<tr>
<td>+ Noise adaptive recognition</td>
</tr>
</tbody>
</table>

4.3. Experimental results

<table>
<thead>
<tr>
<th>Table 3. Aurora 3 Summaries.</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Aurora 3 Reference Word Error Rate</em></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><em>Finnish</em></th>
<th><em>Spanish</em></th>
<th><em>German</em></th>
<th><em>Danish</em></th>
<th><em>Average</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>Well</td>
<td>7.26%</td>
<td>7.06%</td>
<td>8.80%</td>
<td>12.72%</td>
</tr>
<tr>
<td>Mid</td>
<td>19.49%</td>
<td>16.69%</td>
<td>18.96%</td>
<td>32.68%</td>
</tr>
<tr>
<td>High</td>
<td>59.47%</td>
<td>48.45%</td>
<td>26.83%</td>
<td>60.63%</td>
</tr>
<tr>
<td>Overall</td>
<td>24.59%</td>
<td>20.78%</td>
<td>16.86%</td>
<td>31.68%</td>
</tr>
</tbody>
</table>

| *Aurora 3 Relative Improvement* |

<table>
<thead>
<tr>
<th><em>Finnish</em></th>
<th><em>Spanish</em></th>
<th><em>German</em></th>
<th><em>Danish</em></th>
<th><em>Average</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>Well</td>
<td>6.23%</td>
<td>6.07%</td>
<td>7.16%</td>
<td>10.41%</td>
</tr>
<tr>
<td>Mid</td>
<td>16.65%</td>
<td>10.48%</td>
<td>16.94%</td>
<td>23.00%</td>
</tr>
<tr>
<td>High</td>
<td>14.73%</td>
<td>27.95%</td>
<td>16.88%</td>
<td>22.55%</td>
</tr>
<tr>
<td>Overall</td>
<td>12.08%</td>
<td>13.08%</td>
<td>13.01%</td>
<td>17.85%</td>
</tr>
</tbody>
</table>

Recognition performances of the system and reference system provided by AURORA 3 are shown in Table 3. The reference system had applied voice activity detection. Compared to the reference performances, the system with combination of several techniques further decreased WER. For example, in the Danish subset, WER decreased from 12.72% in Well-matched evaluation to 10.41%, which had relative performance improvement of 18.16%. Significant performance improvement was observed in High-mismatched evaluation for all languages. For example, the relative performance improvement were 75.23% and 62.81%, respectively, for Finnish and Danish subset. In average of the four languages, the system made 16.25%, 23.01%, and 54.36% relative performance improvement, respectively, in Well-matched, Medium-mismatched, and High-mismatched evaluations. As a whole, the system made 28.15% relative performance improvement over the reference system.

5. CONCLUSIONS

The results presented in this paper show that system robustness can be achieved by combining a set of techniques in different stages for speech recognition. In particular, it is shown that segmentation is necessary for system robustness by removing some noise segments which can possibly cause recognition error in later stages. Noise adaptive speech recognition is shown to be a useful method for further improvement over that achieved by MLLR. It is seen that they can possibly boost each other. MLLR can do adaptation on not only the static coefficients but also first- and second-order coefficients, and the noise adaptive speech recognition can do non-stationary channel and noise compensation of the static coefficients. Further improvement of this work can be done by adaptation of the first- and second-order coefficients by the noise adaptive speech recognition and refinement of each modules.

6. ACKNOWLEDGEMENT

This research was supported in part by the Telecommunications Advanced Organization of Japan.

7. REFERENCES


