Speech corpus recycling for acoustic cross-domain environments for automatic speech recognition

Osamu Ichikawa\textsuperscript{1,}\textsuperscript{*}, Steven J. Rennie\textsuperscript{2}, Takashi Fukuda\textsuperscript{1} and Daniel Willett\textsuperscript{3}
\textsuperscript{1}IBM Research, 19–21 Nihonbashi, Hakoza-cho, Chuo-ku, Tokyo, 103–8510 Japan
\textsuperscript{2}IBM Research, T.J. Watson Research Center, 1101, Kitchawan Rd, Yorktown Heights, NY, 10598 U.S.A.
\textsuperscript{3}Nuance Communications Inc., Kackertstrasse 10, Aachen, 52072 Germany
(Received 5 January 2015, Accepted for publication 29 October 2015)

Abstract: In recent years, server-based automatic speech recognition (ASR) systems have become ubiquitous, and unprecedented amounts of speech data are now available for system training. The availability of such training data has greatly improved ASR accuracy, but how to maximize the ASR performance in new domains or domains where ASR systems currently fail (thus limiting data availability) is still an important open question. In this paper, we propose a framework for mapping large speech corpora to different acoustic environments, so that such data can be transformed to build high-quality acoustic models for other acoustic domains. In our experiments using a large corpus, our proposed method reduced errors by 18.6%.

Keywords: Speech recognition, Adaptation, Channel compensation, Noise reduction, Vector Taylor series

PACS number: 43.60.Uv [doi:10.1250/ast.37.55]

1. INTRODUCTION

Ideally, an acoustic model of automatic speech recognition (ASR) should be trained with data from the target acoustic domain because the performance of ASR is strongly dependent on the acoustic environment. Therefore, major investments have been made to acquire speech data for desktop dictation systems, telephony systems, and embedded systems for automobiles, among other domains. This expense often makes it difficult to justify developing new ASR products, especially for less broadly spoken languages.

Nowadays, server-based or cloud-based ASR services are becoming prevalent and the environment is changing. People talk to their smartphones and tablet devices for messaging, searches, and virtual assistant applications that transmit speech data to ASR servers that return recognized text. With this approach, an ASR server can easily and inexpensively collect huge amounts of natural and spontaneous speech data. Such data can be the most effective training data to make better acoustic models for these systems because the acoustic characteristics, speaking styles, and linguistic characteristics match the true target. To use this collected data for training, accurate transcriptions (manual or automatic) are needed. For an automatic transcription approach, a retrained acoustic model can be used to transcribe the collected data. The retrained acoustic model makes better transcriptions, and the better transcriptions train the acoustic model for better recognition. Also, the language model for the transcriptions can be retrained with the transcribed text. By iterating this loop with huge amounts of real data, the acoustic model can be improved and thus better able to decode realistic speech with higher accuracy.

Currently, this benefit is limited to use within the same acoustic domain. Even if we have a huge real speech corpus with reliable transcriptions, we cannot build an acoustic model for a different domain.

Our objective in this paper is to improve the performance of ASR for a specific target domain using a large speech corpus from a different domain. For example, hands-free ASR in a meeting room, in-car ASR, or embedded ASR for an appliance might be the target domain, but these target domains have different recording conditions (such as microphone distances and noise characteristics). In addition, even within one source

\textsuperscript{*}e-mail: ichikaw@jp.ibm.com
domain, each utterance may have different recording conditions and there may be no tags for the conditions. If we can overcome these technical hurdles, there is enormous potential to apply the power of a large speech corpus for speech recognition in other domains.

2. CONVENTIONAL METHODS

2.1. Model Adaptation

In this approach, a base acoustic model must be trained with abundant source domain data. Then we can adapt the base acoustic model to a target domain using a small amount of target data. Parallel model combination (PMC) [1] or Vector Taylor series (VTS) adaptation [2] can transform models to target environments. Maximum likelihood linear regression (MLLR) [3] adjusts the model parameters to maximize the likelihood of the adaptation data. Also, Maximum a posteriori (MAP) adaptation [4] is often used.

However, recent product-level acoustic models have been deliberately built with nonlinear complex steps including feature-space discriminative training [5] and model-based discriminative training [6]. Therefore, adaptation as a post-process may weaken the optimized acoustic model.

In the field of deep neural networks, transfer learning [7] is attracting attention as a means of utilizing large resources in a different domain so to generate a model for a low-resource domain. So far, a cross-lingual task [8] has been reported as a successful example of this.

2.2. Data Selection

If a speech corpus in a source domain is sufficiently large and recorded in various environments, it may be possible to select only the data most compatible with a target domain. If there are several speech corpora, COSMOS [9] can find the most suitable data set, considering the acoustic environments, speaking styles, and speaker characteristics.

However, such opportunities are sometimes limited. If the speech corpora are considerably different from the target domain, the amount of filtered data will tend to be too small. For example, there may be only a small amount of distant-talk data in a large speech corpus collected from smartphones.

2.3. Data Transformation

Speech data can be transformed by channel compensation and noise addition. The most basic approach to synthesizing data for a target domain is to first convolve the impulse responses over clean speech data and then add the environmental noise [10] at a designated target signal-to-noise ratio (SNR). However, to use this approach appropriately, the source data must be recorded with a microphone whose frequency-wise gain characteristic is flat. Also, the recording environment must be noise-free, which does not match our source domain specification.

Stereo mapping method such as SPLICE [11] can address this restriction. The source data does not have to be clean and the microphone can be of any type. Since stereo mapping method estimates the mapping of the features between the source domain and the target domain on the basis of an a priori model trained with stereo data, it can compensate for noise characteristics as well as for channel characteristics. However, SPLICE requires stereo data from both the source domain and the target domain, which in practice is difficult to prepare. Also, our source domain is so diverse that it cannot be modeled by one mapping model.

2.4. Adaptive Training

Speaker adaptive training (SAT) [12] normalizes interspeaker variations in the training data, making it possible to build acoustic models that more accurately represent the phonetically relevant information. Similar to SAT, noise adaptive training (NAT) [13] was proposed to normalize environmental variations. SAT and NAT are highly advantageous for obtaining better ASR accuracy. However, the decoding scenarios include speaker adaptation and noise compensation. As speaker adaptation requires several utterances, this scenario is sometimes not accepted depending on the system to be used.

As we assume a large corpus spoken by many people without labels, we would prefer to rely on the variation in the data itself than to normalize the data.

2.5. Decoding-Time Front End Processing

Another approach is to use an original acoustic model trained for the source domain, rather than building a model specialized for the target domain. The input features should be compensated in the front end of the decoder. A model-based approach is often used for noise reduction [14]. VTS can be used to compensate for both the channel and noise [15]. An advantage of this approach is that we can use the same acoustic model for any target domain. However, the improvement in accuracy tends to be smaller than that with specialized acoustic models. For example, multi-condition training for a specific noisy environment often shows better accuracy than clean-condition training combined with noise reduction at the time of decoding [16].

3. OVERVIEW

Our approach is based on data transformation, as discussed in Sect. 2.3. Figure 1 shows our mapping pipeline for the two major factors of channel and noise [17]. Here, we assume that a small amount of clean speech data and noise data in the target domain is available.
For the channel mapping, the basics are as follows. The source domain has multiple utterances but they are so dissimilar that they cannot be modeled with one model. There is a target domain with a small amount of clean data from which a Gaussian mixture model (GMM) can be estimated. For each utterance in the source domain, a channel bias and amplitude must be found to map the utterance to the target domain. To do this, VTS [18,19] maps the target domain Gaussians to the source domain. These mappings include the bias and amplitude as unknown parameters, and the maximum likelihood (ML) is used to obtain the best estimates. Then source utterances are transformed to the target domain by using the channel bias and amplitude. After the noise addition step, the transformed source data is used as training data to estimate a complete acoustic model for decoding sentences in the target domain. In this paper, we call this framework "audio mapping."

It is important to study how much the channel mapping may affect speaker characteristics. Generally speaking, it is difficult to work independently of channel characteristics and speaker characteristics. Cascaded MLLR [20] was devised to serially combine the environmental adaptation and speaker adaptation steps. In the field of speaker verification, feature mapping [21] and joint factor analysis [22] have been used to address this issue. It remains a challenging task to separate the environmental characteristics and the speaker characteristics without identifying the speakers and labeling the environments. Because we do not have such labels, the mapping must be performed for each utterance. This paper acknowledges this issue and attempts to mitigate it by

- using a GMM trained with multiple speakers,
- projecting deviations in the source domain to the target domain,
- using a gender-dependent GMM.

Another issue we should discuss is the need for channel mapping in the pipeline while the ASR decoder performs the run-time channel normalization. To study this problem, the same automobile noise was added to two different kinds of speech data. Figure 2 shows the distributions of the noises after cepstrum mean normalization (CMN) for each utterance. Even though the same noise was added, the resulting noises are markedly different. This motivated us to map the channel characteristics before noise addition to ensure consistent noise characteristics on the decoder side.

## 4. CHANNEL MAPPING

### 4.1. Bias-Only Formulation

First we consider the channel bias only. For the source domain, observation \( y \) is described using clean speech \( x \), channel \( h \), noise \( n \), and mismatch function \( G \) in the cepstrum domain as

\[
y = h + x + G(x + h, n),
\]

where

\[
G(x, n) = C \log(1 + \exp(C^{-1}(n - x))).
\]

The matrix \( C \) is a discrete cosine transform (DCT) matrix.

The target domain clean speech \( \hat{y} \) can be characterized as

\[
\hat{y} = \hat{h} + x.
\]

Combining Eqs. (1) and (3), we have

\[
y = \hat{y} + (h - \hat{h}) + G(x + h, n) = \hat{y} + c + G(\hat{y} + c, n).
\]

The channel bias \( c \) is now defined as

\[
c = h - \hat{h}.
\]

We set \( c \) so as to minimize the auxiliary function \( Q \).
\[ Q = E \left[ \sum_{k} \rho_k(y) \cdot \left\{ \sum_{d} \left( \frac{y_d - \mu_{y,k,d}}{\Sigma_{y,k,d}} \right)^2 / \Sigma_{y,k,d} + \log |\Sigma_{y,k}| \right\} \right]. \] (6)

The expectation is calculated in the speech segments of each utterance. They are detected by model-based segmentation using the speech GMM and the noise statistics [23]. The noise statistics are calculated in the non-speech segments detected by power-based segmentation. The posterior probability \( \rho_k \) is for the \( k \)th Gaussian, and is calculated as

\[ \rho_k(y) = \gamma_k \cdot N(y; \mu_{y,k}, \Sigma_{y,k}) \Bigg/ \sum_k \gamma_k \cdot N(y; \mu_{y,k}, \Sigma_{y,k}), \] (7)

where \( \gamma_k \) is the prior probability, \( \mu_{y,k,d} \) is the mean statistic for the \( d \)th component of the \( k \)th Gaussian in the source domain, and \( \Sigma_{y,k,d} \) is the variance. We used a diagonal covariance approximation. Because the GMM is given for the target domain, the source domain statistics need to be derived from the target domain statistics \( \mu_{\tilde{y},k,d} \) and \( \Sigma_{\tilde{y},k,d} \) using Eq. (4).

\[ \mu_{y,k,d} \approx \mu_{\tilde{y},k,d} + c_d + G(\mu_{\tilde{y},k} + c, \mu_n). \] (8)

\[ \Sigma_{y,k,d} \approx \sum_l (\delta_{l,k} - F(\mu_{\tilde{y},k} + c, \mu_n)) \cdot (\gamma_l - \mu_{\tilde{y},k,l} - c_l - G(\mu_{\tilde{y},k} + c, \mu_n))(\Sigma_{\tilde{y},k,l}) = 0. \] (11)

The noise statistics \( \mu_n \) and \( \Sigma_{n,d} \) are calculated for the non-speech segments, for which we used power-based segmentation. Alternatively, these parameters could be found iteratively via the expectation maximization (EM) algorithm under a noisy speech model [2] and/or based on dynamic, frame-specific posterior estimates of the noise mean and variance [24].

Using the EM algorithm, we iteratively estimated the channel bias \( c \) for the \( d \)th component. We derive Eqs. (11) and (12) by differentiating Eq. (6) w.r.t. \( c_d \) and setting the derivative to zero. The indirect derivatives of \( F \) can be ignored: thus, the second term in Eq. (6) can also be ignored because \( \Sigma_{y,k} \) is treated as a constant. The symbol \( \delta_{l,i} \) is the Kronecker delta.

Using MMSE, the mapped output \( \hat{y} \) is given by

\[ \hat{y} = y - c - \sum_k \rho_k(y) \cdot G(\mu_{\tilde{y},k} + c, \mu_n). \] (13)

On the right side, the second term \( c \) corresponds to the channel mapping and the third term corresponds to the noise reduction. If the SNR of the source domain data is sufficiently high, the third term can be ignored. In our experiments, the noise reduction was performed in the log-Mel spectrum domain, which will be explained in Sect. 4.5.

4.2. Bias and Amplitude Formulation

Some adaptation techniques such as mean and variance normalization (MVN) [25] and diagonal MLLR [26] amplify feature vectors. Since the amplification fills the SNR gap, it could also be a practical choice for our method, especially when the noise compensation step is omitted. In this section, we incorporate the amplitude into the channel mapping formulation. Thus, Eq. (4) can be extended as

\[ y = a \cdot \hat{y} + c + G(a \cdot \hat{y} + c, n), \] (14)

where * denotes a componentwise product and \( a \) is the amplitude. The source domain statistics are given by

\[ \mu_{y,k,d} \approx a_d \cdot \mu_{\tilde{y},k,d} + c_d + G(a \cdot \mu_{\tilde{y},k} + c, \mu_n). \] (15)

\[ \Sigma_{y,k,d} \approx \sum_l a_l^2 \cdot (\delta_{l,k} - F(a \cdot \mu_{\tilde{y},k} + c, \mu_n)) \cdot (\Sigma_{\tilde{y},k,l}) + \sum_l F(a \cdot \mu_{\tilde{y},k} + c, \mu_n) \cdot \Sigma_{n,l}. \] (16)

Similar to the bias-only case, we take the derivatives of Eq. (6) w.r.t. \( a \) and \( c \), and the derivatives are set to zero to estimate the parameters. For the variance term differentiated w.r.t. \( a \), we only take values contributed by the diagonal part of \( F \) in Eq. (16) for the approximation.
Because the iterative estimation of $a$ involves many indirect derivatives, the stability depends on the way of iteration and approximation. Thus, we introduce a parameter $\beta$ given by Eq. (17) to control the stability.

$$Q = E \left[ \sum_{k}^{K} \rho_k(y) \cdot \left\{ \sum_{d}^{D} (y_{d} - \mu_{y,k,d})^2 / \Sigma_{y,k,d} + \beta \log |\Sigma_{y,k}| \right\} \right].$$ (17)

Starting with the initial values of $c = 0$ and $a = 1$, the mapping parameters $c$ and $a$ are updated iteratively. In our experiment, we updated $c$ first, then $a$. For $c$, all components were updated. For $a$, only the zeroth to second components were updated because the lower cepstra are the major and stable representations of the channel. To handle possibly problematic utterances, we limited the range of $a$ from 0.5 to 2.0.

The mapped output $\hat{y}$ is given by

$$\hat{y} = a^{-1} \left( y - c - \sum_{k}^{K} \rho_k(y) \cdot G(a \cdot \mu_{y,k} + c, \mu_{a}) \right).$$ (18)

### 4.3. Projection of Deviation in the Source Domain

Figure 3 shows an example distribution of the channel characteristics. In general, the channel characteristics can be displayed only as offsets from a reference model. In the case of the target data (a), the mean of the offsets is close to zero because the data and the reference model are matched. In contrast, the source data in (b) has a shifted mean and the distribution is diverse. After processing with the channel mapping in (c), the channel characteristics are appropriately projected into the target. However, the distribution seems overly focused on the target compared with the reference distribution in (a), so we were concerned that the speaker characteristics may have been affected. It may be better to use a looser focus to allow for some variation among the speakers and channels.

Therefore, we introduced a perturbation option in the channel mapping. Equation (13) can be extended to

$$\hat{y} = y - c + \varepsilon - \sum_{k}^{K} \rho_k(y) \cdot G(\mu_{y,k} + c, \mu_{a}).$$ (19)

The perturbation $\varepsilon$ is given for each utterance. This is a scaled deviation measured in the source domain, defined as

$$\varepsilon = s \cdot (c - \bar{c}).$$ (20)

The mean channel bias $\bar{c}$ is for all of the utterances in the source domain, and $s$ is the scaling factor. In our experiments, $s$ was set to 0.1. Figure 3(d) shows an example of the distribution processed with this projection. It has a somewhat larger distribution than that obtained with the standard channel mapping (c).

### 4.4. Gender-Dependent Formulation

Another attempt to handle the variation in speaker characteristics is a gender-dependent approach. Similar to the gender-dependent labeling (GDL) [27] for an ASR decoder, we can prepare separate GMMs for male and female voices, which are combined with the gender weights $\lambda_g$. The auxiliary function $Q$ in Eq. (6) is extended to Eq. (21).

$$Q = E \left[ \sum_{g=m,f}^{m} \lambda_g \sum_{k}^{K} \rho_g(y) \cdot \left\{ \sum_{d}^{D} (y_{d} - \mu_{y,g,k,d})^2 / \Sigma_{y,g,k,d} + \log |\Sigma_{y,g,k}| \right\} \right].$$ (21)

The gender weights $\lambda_g$ are updated on the basis of the posterior probabilities of the GMMs during the iterations to determine the mapping parameters as

$$\lambda_g' = E \left[ \sum_{k}^{K} \gamma_{g,k} \cdot N(y; \mu_{y,g,k}, \Sigma_{y,g,k}) \right].$$ (22)
\[ \lambda'_g = \lambda' \frac{1}{\sum_{g' = f, m} \lambda'_{g'}.} \]  

We optionally used Softmax for faster gender determination as

\[ \lambda_g = \exp(\alpha \cdot \lambda'_g) \frac{1}{\sum_g \exp(\alpha \cdot \lambda'_g)}. \]  

If Softmax is not used, then \( \lambda'_g \) should be used for \( \lambda_g, \alpha \) is a constant. For the mapped output, Eq. (13) is enhanced as

\[ \hat{y} = y - \sum K_g \sum_k \lambda_g \cdot \rho_{g, k}(\hat{y}) \cdot G(\mu_{\hat{y}, g, k} + c, \mu_n). \]  

Figure 3(e) shows an example of the distribution processed with this method. It has the closest distribution to the target shown in Fig. 3(a).

4.5. Noise Reduction in the Log-Mel Spectrum Domain

The source domain data may include noisy data. Therefore, the noise reduction capability in the channel mapping is important. Various studies on noise reduction have been carried out. Many of them were implemented in the log-Mel spectrum domain instead of the cepstrum domain because various techniques can be combined in the log-Mel spectrum domain to enhance the performance. As we also planned to combine the frequency-wise confidence metric [28] for future improvement, we moved the noise reduction part to the log-Mel spectrum domain. The third term in the righthand side of Eq. (13) is omitted and the output \( \hat{y} \) is converted to \( \hat{\hat{y}} \), where

\[ \hat{\hat{y}} = C^{-1}\hat{\hat{y}}. \]  

We use the grave accent ‘ to indicate a variable or a function in the log-Mel spectrum domain.

Since our target domain GMM is given in a cepstrum domain, it needs to be converted into a log-Mel spectrum domain using

\[ \hat{\hat{\mu}}_{\hat{g}, k} = C^{-1}\hat{\hat{\mu}}_{\hat{g}, k}, \]  

and

\[ \hat{\hat{\Sigma}}_{\hat{g}, k} = C^{-1} \hat{\hat{\Sigma}}_{\hat{g}, k}(C^{-1})^T. \]

Note that the diagonal covariance matrix in the cepstrum domain becomes a full covariance matrix after this conversion.

Using MMSE, we can output the noise-reduced data \( \hat{\hat{y}} \) as

\[ \hat{\hat{y}} = \hat{\hat{y}} - \sum K_k \rho_k(\hat{\hat{y}}) \cdot G(\hat{\hat{\mu}}_{\hat{g}, k}, \mu_n). \]  

Using Eq. (29), we can perform noise reduction in the log-Mel spectrum domain. Then we can convert the noise-reduced signal back to the cepstrum domain as

\[ \hat{y} = C\hat{\hat{y}}. \]  

The posterior probability for the \( k \)th Gaussian is defined as

\[ \rho_k(\hat{y}) = \gamma_k \cdot N(\gamma_y; \mu_{\hat{g}, k}, \Sigma_{\gamma, k}) \]  

and

\[ \Sigma_{\gamma, k} \approx (1 - F(\hat{\hat{\mu}}_{\hat{g}, k}, \mu_n)) \cdot (1 - F(\hat{\hat{\mu}}_{\hat{g}, k}, \mu_n)) \cdot \Sigma_{\hat{g}, k} + F(\hat{\hat{\mu}}_{\hat{g}, k}, \mu_n) \cdot F(\hat{\hat{\mu}}_{\hat{g}, k}, \mu_n) \cdot \Sigma_{\hat{g}, k} + F(\hat{\hat{\mu}}_{\hat{g}, k}, \mu_n) \cdot F(\hat{\hat{\mu}}_{\hat{g}, k}, \mu_n) \cdot \Sigma_{\hat{g}, k}, \]

where the functions \( F \) and \( G \) are given as

\[ G(x, n) = \log(1 + \exp(n - x)), \]  

\[ F(x, n) = (1 + \exp(x - n))^{-1}. \]

When we use the gender-dependent approach, we can reuse the gender weights \( \lambda_g \) estimated in the channel mapping part in Eq. (24). Equation (29) can be extended to

\[ \hat{\hat{y}} = \hat{\hat{y}} - \sum K_k \lambda_g \cdot \rho_k(\hat{\hat{y}}) \cdot G(\hat{\hat{\mu}}_{\hat{g}, k}, \mu_n). \]  

5. NOISE ADDITION

After the channel mapping, we perform noise addition to simulate the noise characteristics in the target domain. Ideally, the microphone used to record the noise data should be the same as the microphone used to record the speech data for GMM training. Noise data should be recorded in various situations likely to occur in the target domain. The typical SNR in each situation should be measured so that the noise addition can be performed at realistic SNRs.

For example, if the target domain is in-car ASR, situations might include a parked car, city driving, highway driving, and the fan set to high. Before the channel mapping, the SNR of the source utterance may be measured. If the SNR is high, then the parked-car noise might be prioritized in the noise addition. If the SNR is low, then highway noise could be used. In this way, we can minimize side effects from the noise reduction and we can expect some benefit from the Lombard effect.

6. PRELIMINARY EXPERIMENTS

Before performing experiments with a large speech
was severely degraded, even though the far-field data mapping. Owing to the channel mismatch, the accuracy using close-talk data for the training without using channel gives us the upper limit for our trials. R2 is our baseline of the matched case using far-field data for the training. This cepstrum mean.

power, with 39-dimensional features (12 mel-cepstrum + log of CENSREC-3. The sampling frequency was 16 kHz and 50 Japanese words.

did not include noisy cases such as those air-conditioner recorded in parked cars with a far-field microphone. This spoken by 18 speakers (eight males and 10 females) were recorded. The recognition grammar was a list of GMMs and was trained with 13-dimensional MFCC vectors. Fifteen subject speakers (seven females and eight males) were recorded. The GMMs for the mapping had 256 Gaussians and was trained with 13-dimensional static MFCC vectors of 500 randomly selected utterances from clean far-field microphone data. For testing, a total of 898 utterances, recorded with a close-talk microphone, were transformed by channel mapping methods to simulate far-field or close-talk data. Results are given as the word error rate (WER) %.

<table>
<thead>
<tr>
<th>Training data</th>
<th>Channel mapping</th>
<th>WER %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>no CMN</td>
</tr>
<tr>
<td>R1 Far-field</td>
<td>No</td>
<td>0.3</td>
</tr>
<tr>
<td>R2 Close-Talk</td>
<td>No</td>
<td>7.7</td>
</tr>
<tr>
<td>R3 Close-Talk</td>
<td>Bias only</td>
<td>6.9</td>
</tr>
<tr>
<td>R3 Close-Talk</td>
<td>Gender Independent</td>
<td>5.5</td>
</tr>
<tr>
<td>R3 Close-Talk</td>
<td>Bias and amplitude, Gender Dependent</td>
<td>3.6</td>
</tr>
</tbody>
</table>

corpus, we verified our channel mapping capability with a relatively small but well-conditioned database.

6.1. CENSREC-3

We used the public database CENSREC-3 [29] to evaluate the channel mapping part in isolation. This is a standard evaluation framework for isolated Japanese word recognition in automobiles. It has both training and testing data for automatic speech recognition using multistyle-trained acoustic models.

In this experiment, we studied the channel mapping from the close-talk domain to the far-field domain. Noise reduction and noise addition were not enabled. For acoustic model training, a total of 3,608 utterances spoken by 293 drivers (202 males and 91 females) were recorded. The utterances, recorded with a close-talk microphone, were transformed by channel mapping methods to simulate far-field clean data. The GMM for the mapping had 256 Gaussians and was trained with 13-dimensional MFCC vectors of 500 randomly selected utterances from clean far-field microphone data. For testing, a total of 898 utterances spoken by 18 speakers (eight males and 10 females) were recorded in parked cars with a far-field microphone. This did not include noisy cases such as those air-conditioner or car-audio noise. The recognition grammar was a list of 50 Japanese words.

The front end was configured with the default settings of CENSREC-3. The sampling frequency was 16 kHz and we used 39-dimensional features (12 mel-cepstrum + log power, with $\Delta$ and $\Delta\Delta$) with and without subtracting the cepstrum mean.

Table 1 shows the experimental results. The R1 row is the matched case using far-field data for the training. This gives us the upper limit for our trials, R2 is our baseline using close-talk data for the training without using channel mapping. Owing to the channel mismatch, the accuracy was severely degraded, even though the far-field data contained little background noise. It should be noted that the simple CMN was still insufficient. In this experiment, CMN was performed for all the frames of each utterance regardless of speech or non-speech segments. The R3 to R5 rows use our proposed channel mapping and showed good recovery. For the R3 values, the bias-only implementation described in Sect. 4.1 was used for the mapping. Row R4 shows that the result was improved with gender-dependent GMMs. This is supporting evidence that the gender-dependent GMMs in our approach accounted for both the speaker and channel characteristics. For R5, both the bias and amplitude are considered in the mapping. The parameter $\beta$ in Eq. (17) was set to 0.5. The result shows that the proposed channel mapping method works well and reduces errors by 53% in the no-CMN case and by 58% in the CMN-combined case.

7. EXPERIMENTS

7.1. Experimental Setup

In this experiment, we assume in-car ASR as the target domain. To avoid overtraining for specific conditions such as car types or speeds, speaking styles, fan on/off, or speakers, we used three subdomains for the testing.

- A: In-car messaging, recorded in parked cars. Noise added to simulate moving cars.
- B: In-car commands, recorded in moving cars.
- C: In-car narrative commands, recorded in parked cars. Noise added to simulate moving cars.

For the test data, each subdomain includes 11 h of data. The performance was measured as the average for the three subdomains.

For subdomains A and B, we additionally prepared small amounts of clean speech data and noise data to train the GMMs. Since we did not use subdomain C for GMM training or for noise addition, subdomain C is regarded as an open condition for the transformation.

Specifically, subdomain A includes two types of car. Fifteen subject speakers (seven females and eight males) and 15 subject speakers (eight females and seven males) were used for each type to train two GMMs. The GMMs were trained with 13-dimensional static MFCC vectors. The size of the training data in subdomain A was 6.6 h in total. Also, subdomain B includes five data sets, each consisting of multiple cars, recorded under various driving conditions with multiple speakers. For each set, we selected clean speech data by checking the SNR, then trained five GMMs with the data. The size of the selected data was 3.4 h in total for the five data sets.

Therefore, we have seven GMMs (or 14 gender-dependent GMMs) in total. It should be noted that multiple noise data was associated with one GMM. Specifically, in subdomain A, six driving conditions were defined and noise data was collected for each condition in the car. In
subdomain B, we classified data into six conditions based on the SNR to extract noise data for each data set. When we perform the data transformation by audio mapping, the GMM and noise data were randomly selected for each utterance. First, subdomain A or B was randomly selected, and then a GMM was randomly chosen from the selected subdomain. Finally, noise data associated with the GMM was randomly selected. The noise data was added with an appropriate scale so that the SNRs of the transformed utterances have a realistic distribution.

The source domain data is a large amount of Japanese LVCSR data that was spoken by various speakers in various environments. Most of the data is spontaneous speech and the total amount of data is several thousand hours. A few hundred hours of data was manually transcribed and the rest was transcribed by ASR. Since this source data includes various SNR levels, the noise reduction step explained in Sect. 4.5 was also used, which means that the amplitude combination from Sect. 4.2 was not used.

Using the processed data and the transcriptions, a conventional GMM/HMM model was built for the target domain. The data at a sampling frequency of 16 kHz was converted to a 13-dimensional MFCC. Each frame was concatenated with neighboring frames and reduced to 32-dimensional features by using LDA combined with the semi-tied covariance (STC). Phones were modeled with three-state left-to-right HMMs that did not permit state skipping. Acoustically distinct variants of the HMM states were identified using decision trees that asked questions about the phonetic context in which a state occurs, and the leaves of the decision tree were the basic acoustic units that we modeled. The model we used contained roughly 120,000 Gaussians with 3,000 quin-phone context-dependent states. After the ML training, we performed the feature-space discriminative training, and then the acoustic models were further processed by feature-space discriminative training and model-space discriminative training.

Because the test task is Japanese dictation, the word error rate (WER) as an evaluation metric has problems with ambiguous word segmentation, so the character error rate (CER) is often used instead. However, the CER was also unsuitable for these data sets because of inconsistent notations and homonyms. Therefore, we used the Kana error rate (KER), where the recognized text and reference text are converted into the most appropriate Kana representing the pronunciation and then compared. Since we focused on the acoustic aspects in this experiment, we believe that using the KER was the most suitable approach.

Note that some of the preliminary experiments in this section were conducted with a small subset of the source data. The subset data size was only 50 h and the data was manually transcribed.

### Table 2 Results of decoding the three tasks. The acoustic models were built with subset data. Channel mapping was performed with various sizes of GMMs. GI means gender independent. Results are given as the Kana error rate (KER) %.

<table>
<thead>
<tr>
<th>GMM</th>
<th>Test A</th>
<th>Test B</th>
<th>Test C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>11.19</td>
<td>19.06</td>
<td>10.41</td>
</tr>
<tr>
<td>GI 64 mixture</td>
<td>10.18</td>
<td>18.88</td>
<td>10.09</td>
</tr>
<tr>
<td>GI 128 mixture</td>
<td><strong>10.09</strong></td>
<td>18.40</td>
<td>10.16</td>
</tr>
<tr>
<td>GI 256 mixture</td>
<td>10.26</td>
<td><strong>18.03</strong></td>
<td>10.15</td>
</tr>
<tr>
<td>GI 512 mixture</td>
<td>10.32</td>
<td>18.27</td>
<td>10.35</td>
</tr>
</tbody>
</table>

### Table 3 Results of decoding the Test A task. The acoustic models were built with subset data by audio mapping for Test A. Results are given as KER %.

<table>
<thead>
<tr>
<th>GMM</th>
<th>Test A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>11.19</td>
</tr>
<tr>
<td>2 subject speakers, GI 256 mix</td>
<td>10.48</td>
</tr>
<tr>
<td>30 subject speakers, GI 256 mix</td>
<td>10.28</td>
</tr>
</tbody>
</table>

#### 7.2. GMM Mixture Size

First, we attempted to find the appropriate number of mixtures for the GMM using the small subset of the source data. This was processed by channel mapping for GMMs of different sizes and with added noise. Acoustic models were built with the processed data. Table 2 shows the results of these experiments. “Baseline” is the case in which the acoustic model was built without audio mapping. All of the audio mapping cases showed improvements over the baseline. In total, for the three tasks, the result for 256 mixtures was the best, so we used 256-mixture GMMs in the following experiments.

#### 7.3. Speaker Variation for GMM Training

We verified that speaker variation must be considered in GMM training. Two acoustic models were built with the subset data processed by audio mapping with different numbers of subject speakers. The channel mapping target was subdomain A only. Since we had two types of cars in subdomain A, we selected one male speaker for each type and trained two GMMs. The data size was 0.22 h and about 160 sentences per speaker. This is the case “2 speakers” in Table 3. As we discussed in Sect. 7.1, our default setting uses 15 speakers for each type to train two GMMs for subdomain A. This is the case “30 speakers” in Table 3. The result indicates we still obtain marginal improvement with the minimum number of speakers, but we should include more speakers for further improvement.

#### 7.4. Need for Channel Mapping

We wanted to verify that channel mapping is an essential part of the data transformation. In Table 4, the
“Noise only” case includes noise reduction and noise addition but does not use channel mapping. There was some improvement in Test A, but the performance was degraded in Tests B and C. In summary, “Noise only” did not outperform the full audio mapping case including channel mapping.

7.5. Deviation Projection and Gender-Dependent GMM

To preserve the variations in speaker characteristics, two methods were discussed in Sects. 4.3 and 4.4. We evaluated the proposed methods using the subset data. Table 5 shows the testing results for these methods. Deviation projection combined with a gender-independent GMM improved the accuracy in Tests A and C but there was major degradation in Test B. In contrast, a gender-dependent approach significantly improved the accuracy in all three tasks. The combination of deviation projection and a gender-dependent approach did not perform well, so we did not use that combination in the main experiment.

7.6. Experimental Results with the Large Speech Corpus

We built several acoustic models trained with the full set of the large corpus, with and without audio mapping. Table 6 shows the experimental results using all of the source domain data. “Baseline” did not apply audio mapping for the source data in the training acoustic model. Compared with Table 2, the error rates were greatly reduced by the additional training data. Case 1 is for reference. The baseline model was adapted with CMLLR and MLLR with 40 h of subdomain A data. This adaptation data was generated from the clean-speech data and the noise data used for the training of the GMMs in subdomain A. With this adaptation, Test A was improved but side effects were observed in Tests B and C. The adaptation somewhat degraded the performance in unknown acoustic environments. This was the motivation for us to recommend audio mapping.

Case 2 is also for reference and uses more adaptation data. It was adapted with 40 h of subdomain A data and 138 h of subdomain B data. Since the subdomain B data was actually recorded in moving cars, it is relatively rich and favored condition considering the cost of recording in moving cars for long hours. However, this case still performed worse than the audio mapping cases. Similar to Case 1, there was degradation in the open condition (Test C). Set C is similar to Sets A and B in the sense that the data was collected in cars. However, the car types, microphone type/position, and speaking styles were different across the subdomains. This implies there were considerable variations in the channel and noise characteristics. In general, an adaptation method often uses only a small amount of data to transform the acoustic model. Therefore, the adapted model tends to be specialized for the specific target. In our interpretation, the degradation for Set C was caused by some overfitting to Sets A and B.

Case 3 used audio mapping with gender-independent 256-mixture GMMs. The improvement from the baseline was significant, even under the open condition of Test C. Overall, for the three tasks, the Case 3 model reduced the errors by 17.1%.

Case 4 used audio mapping with gender-dependent 256-mixture GMMs. This further reduced the errors by 18.6% for the three tasks.

The amount of training data we used in the experiment was huge and the data had many speaker and environmental variations. In other words, the data included various channel and speaker characteristics. We assume the variety in the source data still remained after our proposed transformation. That is why the proposed model showed better accuracy for Sets A and B than the model adapted for Sets A and B. In other words, the proposed model retained good generalization capability when it was adapted for the target domain.

Table 4 Results of decoding the three tasks. The acoustic models were built with subset data. Results are given as KER %.

<table>
<thead>
<tr>
<th>GMM</th>
<th>Test A</th>
<th>Test B</th>
<th>Test C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>11.19</td>
<td>19.06</td>
<td>10.41</td>
</tr>
<tr>
<td>Noise only</td>
<td>10.39</td>
<td>19.41</td>
<td>10.66</td>
</tr>
<tr>
<td>GI 256 mixture</td>
<td>10.26</td>
<td>18.03</td>
<td>10.15</td>
</tr>
</tbody>
</table>

Table 5 Results of decoding the three tasks. The acoustic models were built with subset data. DP means deviation projection. GD means gender dependent. Results are given as KER %.

<table>
<thead>
<tr>
<th>GMM</th>
<th>Test A</th>
<th>Test B</th>
<th>Test C</th>
</tr>
</thead>
<tbody>
<tr>
<td>GI 256 mixture</td>
<td>10.26</td>
<td>18.03</td>
<td>10.15</td>
</tr>
<tr>
<td>GI 256 mixture + DP</td>
<td>10.02</td>
<td>18.98</td>
<td>10.07</td>
</tr>
<tr>
<td>GD 256&amp;256</td>
<td>9.87</td>
<td>17.71</td>
<td>9.85</td>
</tr>
<tr>
<td>GD 256&amp;256 + DP</td>
<td>10.19</td>
<td>18.10</td>
<td>10.10</td>
</tr>
</tbody>
</table>

Table 6 Results of decoding the three tasks. The acoustic models were built with subset data. Results are given as KER %.

<table>
<thead>
<tr>
<th>Model</th>
<th>Test A</th>
<th>Test B</th>
<th>Test C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>8.89</td>
<td>13.71</td>
<td>7.52</td>
</tr>
<tr>
<td>Case 1: Adapt. small</td>
<td>8.74</td>
<td>14.11</td>
<td>7.99</td>
</tr>
<tr>
<td>Case 2: Adapt. large</td>
<td>7.67</td>
<td>10.54</td>
<td>7.63</td>
</tr>
<tr>
<td>Case 3: GI + DP</td>
<td>7.58</td>
<td>10.52</td>
<td>6.87</td>
</tr>
<tr>
<td>Case 4: GD</td>
<td>7.49</td>
<td>10.17</td>
<td>6.86</td>
</tr>
</tbody>
</table>
Table 7 shows the error reduction rate of our proposed method (Case 4) from the baseline. The proposed method had large gains in the lower-SNR region. On the other hand, it had some degradation in the higher-SNR region. This is because the baseline model was built mostly with high-SNR data. As the size of the baseline model was almost the same as that of the proposed model, it used more modeling power to describe high-SNR data. However, the degradation by several percent in the high-SNR region should not be a major problem in practice because the absolute error rate was only a few percent.

8. CONCLUSION

With the widespread use of server-side ASR, it has become feasible to collect huge amounts of real-world spoken data at little cost. Therefore, the interest in ways to reuse such large speech corpora in other domains for acoustic model development is increasing. In this paper, we discussed an audio mapping framework based on a VTS formulation that transforms speech data for different acoustic environments.

ACKNOWLEDGEMENTS

We would like to express our special thanks to Prof. Masafumi Nishimura, Dr. Ryuki Tachibana and Dr. Vaibhava Goel, who provided valuable comments to help this research succeed. The present study was conducted using the CENSREC-3 database developed by the IPSJ-SIG SLP Noisy Speech Recognition Evaluation Working Group.

REFERENCES

[23] J. Wu and X. L. Zhang, “An efficient voice activity detection algorithm by combining statistical model and energy detec-


Osamu Ichikawa is a research staff member at the IBM Research - Tokyo and belongs to the speech technology group. He received B.E. and M.E. degrees in Aeronautical Engineering from Tokyo University, Japan, in 1986 and 1988, respectively, and a Ph.D. degree in Information Science from Nara Institute of Science and Technology, Japan, in 2008.

Steven J. Rennie is a research staff member at the IBM T.J. Watson Research Center in New York. He received his Ph.D. degree in computer engineering from the University of Toronto, Canada, in 2007. His current research interests include machine learning using deep neural networks and graphical models, and their application to problems in machine perception.

Takashi Fukuda received his Ph.D. degree from Toyohashi University of Technology, Japan, in 2005. He has been with the speech technology group, IBM Research - Tokyo since 2005. Dr. Fukuda received the 28th Awaaya Kiyoshi Young Researcher Award in 2010, IEICE ISS Young Researcher’s Award in Speech Field in 2012, and IPSJ Yamashita SIG Research Award in 2013.

Daniel Willett received Diploma in Computer Science from TU Darmstadt in 1995, Ph.D. in Electrical Engineering from Duisburg University in 2000. After two years as PostDoc with NTT in Japan, he joined Harman Becker in Germany. In 2006, he joined Nuance in Aachen, Germany, where he manages an acoustic modeling research team in charge of Nuance’s cloud ASR offerings.