Delcroix *et al. EURASIP Journal on Advances in Signal Processing* (2015) 2015:60
DOI 10.1186/s13634-015-0245-7

EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH Open Access

Strategies for distant speech recognition in reverberant environments

Marc Delcroix*, Takuya Yoshioka, Atsunori Ogawa, Yotaro Kubo, Masakiyo Fujimoto, Nobutaka Ito, Keisuke Kinoshita, Miquel Espi, Shoko Araki, Takaaki Hori and Tomohiro Nakatani

Abstract

Reverberation and noise are known to severely affect the automatic speech recognition (ASR) performance of speech recorded by distant microphones. Therefore, we must deal with reverberation if we are to realize high-performance hands-free speech recognition. In this paper, we review a recognition system that we developed at our laboratory to deal with reverberant speech. The system consists of a speech enhancement (SE) front-end that employs long-term linear prediction-based dereverberation followed by noise reduction. We combine our SE front-end with an ASR back-end that uses neural networks for acoustic and language modeling. The proposed system achieved top scores on the ASR task of the REVERB challenge. This paper describes the different technologies used in our system and presents detailed experimental results that justify our implementation choices and may provide hints for designing distant ASR systems.

Keywords: Reverberant speech recognition; Robust speech recognition; REVERB challenge; Dereverberation; Noise reduction; Deep neural network

1 Introduction

Recently, automatic speech recognition (ASR) is being increasingly used in everyday life. Voice search on smart phones is a good example of a successful application that has triggered great interest in ASR technologies. However, progress is still needed if we are to handle more challenging situations such as distant ASR in the presence of noise and reverberation [1–3]. When using distant microphones to capture speech, the microphone signals are affected by noise and reverberation as follows [4],

$$y(t) = h(t) * s(t) + n(t), \tag{1}$$

where $y(t)$ is an observed microphone signal, $h(t)$ is the room impulse response between the speaker and the microphone, $s(t)$ is a clean speech signal, $n(t)$ represents background noise, and $*$ is a convolution operator. As seen in Eq. 1, reverberation is a convolutive distortion, which produces non-stationary interference that cannot be well suppressed with conventional noise reduction approaches. The problem of reverberation has long been recognized to be critical for distant speech recognition [2, 5–8]. The convolutive nature of reverberation induces a long-term correlation between a current observation and past observations of reverberant speech. This long-term correlation has been exploited to mitigate the effect of reverberation directly on the speech signal (i.e., speech [9–12] or feature [13, 14] dereverberation) or on the acoustic model used for recognition [15, 16].

The REVERB challenge [17] was organized to evaluate recent progress in the field of reverberant speech enhancement (SE) and recognition. The task of the REVERB challenge involves reverberant speech with a moderate amount of background noise for both single and multi-microphone scenarios. The test data covers various acoustic conditions obtained by simulations and real recordings. The training data consist of simulated multi-condition training data. Therefore, there are two main technical challenges posed by the REVERB task, i.e., (1) the recognition of noisy and reverberant speech and (2) the mismatch between the training simulated reverberant speech and testing conditions that include real recordings. In this paper, we describe the system we proposed for the REVERB challenge. Our system addresses the technical

*Correspondence: marc.delcroix@lab.ntt.co.jp
NTT Communication Science Laboratories, NTT Corporation, Hikaridai 2-4, Seika-cho, Kyoto, Japan

challenges of the REVERB task using the combination of an SE front-end and an ASR back-end.

First, we employ an SE front-end to reduce the acoustic interferences. We chose to employ an SE front-end rather than acoustic model-based approaches to handle reverberation and noise because it has the advantage of being independent of the ASR back-end that we used, making it directly usable with, e.g., recent neural network-based acoustic models. Our SE front-end deals first with reverberation using a dereverberation approach based on long-term linear prediction [12, 18, 19]. This approach can greatly reduce reverberation both for single and multi-microphone cases. In the latter case, it can be effectively combined with multi-channel SE for noise reduction. Therefore, we employ the combination of a conventional minimum variance distortionless response (MVDR) beamformer [20] and model-based SE [21, 22] to reduce background noise after dereverberation. Both dereverberation and the beamformer provide linear filtering, which does not cause time-varying distortions to the processed speech and is therefore empirically particularly effective for ASR. Moreover, model-based SE achieves noise reduction while preserving speech characteristics close to that of clean speech by using a pre-trained model of clean speech. Consequently, our SE front-end is well suited for ASR.

Second, to achieve high recognition performance, we use a recognizer that employs a deep neural network (DNN)-based acoustic model and a recurrent neural network-based language model (RNNLM). The acoustic model is carefully designed to prevent overfitting to the simulated reverberant speech that is used for training, by using multi-condition training data that cover various acoustic conditions. This is similar to existing data augmentation techniques [23, 24] but is performed here by varying the SNR of the training data. Moreover, we undertake the unsupervised environmental adaptation of the DNN acoustic model to compensate further for the mismatch between training and testing conditions.

The system we discussed was designed for the REVERB challenge [17, 25], where it achieved top performance in the ASR evaluation. In particular, we achieved a WER of 9 % for the RealData subset of the challenge using our 8 channel SE front-end, which corresponds to a relative improvement of more than 50 % over a strong DNN-based ASR back-end without enhancement. These results demonstrate that a well-designed SE front-end can greatly improve the performance of distant speech ASR even when using DNNs.

This work extends our previous papers about our system [26, 27], by incorporating more details about intermediate experimental results. In particular, we investigate the influence on recognition performance of such aspects of the SE front-end configuration as the prediction filter length, the processing scheme used to reduce reverberation, and the processing order of dereverberation and denoising. Moreover, we also analyze different factors influencing the performance of the ASR back-end, such as the acoustic model topology, the type of multi-condition training data, and the scheme used for environmental adaptation. These novel results make it easier to understand the implementation choices we made during the design of our system and may provide insights that will assist the development of reverberant robust ASR systems.

This paper focuses on the ASR task, and results are provided only in terms of ASR performance. Note that the proposed SE front-end also performed well in the SE evaluation of the REVERB challenge. Interested readers can find more details about the system submitted to the SE task and the results obtained for the SE evaluation in [25, 26].

The organization of the paper is as follows. In Section 2, we first briefly review the characteristics of the REVERB challenge for readers unfamiliar with this task. Section 3 presents a general overview of our proposed system for the recognition of reverberant speech and describes the main components of the SE front-end and ASR back-end. Section 4 provides details of the experimental settings, discusses implementation issues for the SE front-end and the ASR back-end, and presents the final results obtained for the evaluation set. We conclude the paper in Section 5.

2 REVERB challenge task

Before introducing our proposed recognition system, let us briefly review the main characteristics of the REVERB challenge task. More details about the task can be found in [17, 25]. All the utterances employed for the task of the REVERB challenge are text prompts extracted from the Wall Street Journal (WSJ) corpus [28]. The speech data are based on two corpora; i.e., WSJCAM0 [29], which is a British English version of WSJ, and MC-WSJ [30], which consists of read speech recorded in a meeting room with a distant microphone array.

The challenge data include a training set (Train), a development set (Dev), and an evaluation set (Eval).

- *The training set (Train)* has two different versions: clean and multi-condition. *The clean training set* consists of the clean training data set of WSJCAM0. *The multi-condition training set* consists of simulated reverberant speech with additional background noise at an SNR of 20 dB. A script with which to generate the multi-condition training data from the clean training data was available to the challenge participants as well as several room impulse responses and noise signals measured in real rooms [25]. We refer to that multi-condition training data set as the *baseline multi-condition training data set*. Note that the challenge regulations also allowed the use of additional training data.

- *The development set (Dev)* consists of two subsets, simulated reverberant speech (*SimData*), and real reverberant speech recordings (*RealData*). SimData was generated by convolving clean speech signals from WSJCAM0 with room impulse responses measured in three different rooms followed by the addition of background noise measured in the same rooms. The rooms used for the test sets differ from those used for the training data. The reverberation time (RT60) of the rooms ranged from 0.25 to 0.7 s, and the SNR level was about 20 dB. RealData consist of reverberant speech from the MC-WSJ corpus, which consists of speech recorded in a meeting room with a reverberation time of about 0.7 s.
- *The evaluation set (Eval)* also consists of SimData and RealData. The evaluation set covers the same acoustic environments as the Dev set, but with different speakers and different speaker positions in the rooms.

For all data sets, 1 microphone (1ch), 2 microphone (2ch), and 8 microphone (8ch) versions are available. All data sets are available through LDC and the REVERB challenge website [25].

There are two main technical challenges with the REVERB task, i.e., the acoustic conditions that include a large amount of reverberation in addition to a non-negligible amount of background noise and the mismatch between the training data obtained from simulation and the RealData set. In the next section, we present the system we proposed to address these technical challenges.

3 Overview of our proposed system

The system we submitted to the REVERB challenge combines an SE front-end and an ASR back-end as shown in Fig. 1. The SE front-end aims at removing acoustic interferences (i.e., reverberation and background noise) while the back-end undertakes the recognition task. The main characteristics of the front-end and back-end are summarized below.

3.1 SE front-end

Our SE front-end performs dereverberation, beamforming, and model-based SE. The following three subsections describe the ways in which these three functions are realized in our front-end.

3.1.1 Dereverberation

Our SE front-end starts by dereverberating microphone signals with a weighted prediction error (WPE) algorithm [12, 18, 19]. WPE is based on long-term linear prediction, which has been widely used for blind equalization when the target signal is stationary Gaussian. However, when used for speech dereverberation, linear prediction cannot distinguish between the room impulse response and the speech generative process, therefore causing excessive whitening of the processed signal [31]. WPE is a modified version of linear prediction, which is more suitable for speech signals because it models the target signal as a Gaussian with a time-varying variance and also introduces a time delay in the linear prediction to preserve the speech generative process.

The WPE algorithm performs long-term linear prediction in a short time Fourier transform (STFT) space as follows,

$$\boldsymbol{y}_n = \sum_{\tau=T_\perp}^{T_\top} \boldsymbol{G}_\tau^H \boldsymbol{y}_{n-\tau} + \boldsymbol{x}_n, \qquad (2)$$

where $\boldsymbol{y}_n$ is a vector comprising the STFT coefficients of the microphone signals, $\boldsymbol{x}_n$ a prediction error vector, $\boldsymbol{G}_\tau$ is a complex-valued square matrix, called a prediction matrix, n is the time frame index, $T_\top$ and $T_\perp$ are integers with $T_\top > T_\perp > 0$, and superscript H is a conjugate transposition. Note that the frequency bin index is omitted since this algorithm processes signals on a per frequency bin basis. $T_\perp$ represents the time delay in the linear prediction. Our submitted system used $T_\perp = 3$ for all the array set-ups (i.e., irrespective of the number of microphones used) while $T_\top$ was set at 40, 30, and 7 for the 1ch, 2ch, and 8ch set-ups, respectively. These settings provide good performance as shown by the results in Section 4.2.1, which examines the impact of these parameters on WER.

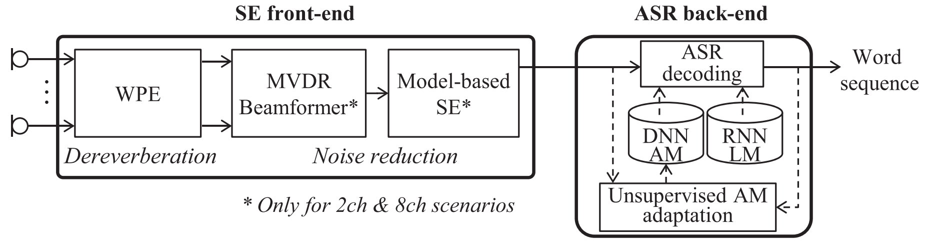


Fig. 1 Schematic diagram of the proposed system for the recognition of reverberant speech

The prediction matrices, $\boldsymbol{G}_{T_\perp}, \cdots, \boldsymbol{G}_{T_\top}$, are optimized for each utterance by minimizing an iteratively re-weighted prediction error criterion [19].

With the optimized prediction matrices, dereverberated signals can be obtained as prediction errors, that is, a dereverberated STFT coefficient vector can be computed as,

$$\boldsymbol{x}_n = \boldsymbol{y}_n - \sum_{\tau=T_\perp}^{T_\top} \boldsymbol{G}_\tau^H \boldsymbol{y}_{n-\tau}. \tag{3}$$

In Eq. 3, the predictions (the second term on the right hand side) are subtracted from the microphone signals. Thus, the predictions may be regarded as estimates of the reverberant interferences contained in the microphone signals. This view leads us to an alternative dereverberation scheme using spectral subtraction. Specifically, instead of subtracting the reverberation estimates in the STFT coefficient space, we subtract the power spectra of the reverberation estimates from the observed power spectra while leaving the phase spectra unchanged in a similar way to [11]. The difference between the two schemes is investigated in Section 4.2.2.

WPE has a number of characteristics that make it very attractive for reverberant speech recognition. WPE is based on a rigorous modeling of reverberation and can thus substantially reduce reverberation. Moreover, WPE is known to be relatively robust to the ambient noise that is inevitably present in distant speech recordings, making it suitable for realistic situations. Finally, the WPE algorithm shortens the room impulse responses between a speaker and microphones to achieve dereverberation [19]. This means that this algorithm allows multi-channel noise reduction techniques based on beamforming to be effectively performed after dereverberation.

3.1.2 Beamforming

The second component constituting our SE front-end is an MVDR beamformer. This component is designed to filter out additive noise that remains in the dereverberated signals. In this work, the MVDR beamformer was implemented by using the scheme proposed in [20] by using noise covariance matrices estimated from the initial and final 10 frames of each utterance.

3.1.3 Model-based SE

Finally, we further reduce background noise using a model-based SE approach. We employ the dominance-based locational and power-spectral characteristics integration (DOLPHIN) approach [22], which combines spectral feature-based SE [32] with a source location feature-based approach [33, 34] into a single optimization framework. This is performed by introducing a dominant source index (DSI), which indicates whether noise or speech is dominant at each time/frequency bin. The DSI is estimated using the EM algorithm by exploiting a source location feature and spectral feature models of speech and noise.

The parameters of the source location feature models for speech and noise are estimated on a per-utterance basis from the multi-channel output of the WPE. The spectral feature models, composed of Gaussian mixture models (GMMs), are trained on the clean speech training data and then adapted to the test conditions in an unsupervised manner using the output of the MVDR. The adaptation is performed using all the utterances of a given acoustic condition (full batch mode)[1] following the procedure described in [22]. The noise spectral model parameters are estimated on a per utterance basis. Noise reduction is finally performed on the MVDR output.

In contrast to the other components of the SE front-end, DOLPHIN is based on non-linear processing, i.e., it processes speech based on frame-wise spectral modification possibly introducing time-varying distortions that may be difficult for the ASR back-end to cope with. However, the use of pre-trained spectral models of speech ensures that the characteristics of the processed speech remain close to that of clean speech.[2]

3.2 ASR back-end

Our ASR back-end relies on neural network techniques for acoustic and language modeling to achieve high recognition performance. Moreover, we also employ unsupervised environmental adaptation to mitigate the mismatch between training and testing acoustic conditions.

3.2.1 Acoustic model

We employ a conventional context-dependent DNN hidden Markov model (CD-DNN-HMM)-based acoustic model. The input features consist of log mel filterbank features that were processed with global mean and variance normalization. The DNN was initialized using layer-wise restricted Boltzmann machine (RBM) pre-training [35]. The DNN was then fine-tuned using stochastic gradient descent (SGD) to optimize the cross entropy criterion. We used the backpropagation algorithm with labels obtained by performing an HMM state alignment of the clean speech training data using a GMM-based recognizer trained with the maximum likelihood criterion.

We trained the DNN using multi-condition training data. The REVERB challenge provided a baseline multi-condition training data set that consists of simulated reverberant speech with additional noise. The reverberant speech of the baseline multi-condition training data set is thus relatively close to that of the SimData test sets. However, the acoustic conditions of the multi-condition training data differ greatly from that of RealData. Therefore, to increase recognition performance especially for the more challenging RealData sets, we extended the training data by increasing the acoustic conditions seen during training. This was simply performed by creating different versions of the multi-condition training data at different SNRs and also adding clean speech and speech recorded by a desktop microphone. Consequently, the *extended training set* consists of the same utterances, and any variation originates solely from different acoustic conditions.

Note that we did not use enhanced speech to train the DNN but instead used the unprocessed distant noisy and reverberant speech. This was shown to provide more robust acoustic models as we discuss using the experiments described in Section 4.3.2. This approach also enabled us to use the same acoustic model for different configurations of the SE front-end, which allows a rapid experimental turnaround.

3.2.2 Unsupervised adaptation

There are two main causes for the mismatch between training and testing conditions, which affect recognition performance. First, as mentioned in the previous section, we use distant unprocessed speech for training the model but enhanced speech for recognition. Consequently, small artifacts or distortions caused by the SE front-end may affect recognition performance. Another cause of the mismatch is the different acoustic conditions seen during training and testing, which is particularly noticeable with the RealData set.

We employed environmental adaptation to mitigate the mismatch between the training and testing conditions. Here, we had to perform an unsupervised adaptation as we do not possess adaptation data with transcriptions. The adaptation of DNNs to speakers or environments has recently attracted much interest [36–41]. Here, we investigate four simple but effective ways to adapt the DNN by retraining its parameters or some of them using labels obtained from a first recognition pass, i.e., a linear input network (LIN) [36][3], retraining the first layers, the last layer, or the whole DNN [37]. The experiments described in Section 4.3.4 reveal that retraining the first two layers can already greatly reduce the mismatch caused by the acoustic conditions.

3.2.3 Language model

As mentioned previously, the different acoustic conditions between training and testing induce a mismatch that affects recognition performance. However, we can consider that the linguistic characteristics do not vary with the acoustic conditions. Consequently, improving language modeling is expected to improve performance, as long as the use of the system remains the same.

Recently, RNNLMs have been used to improve language modeling as they can model the long-term dependency between words that cannot be captured by conventional n-gram models [42]. However, RNNLMs make decoding prohibitively expensive because the word probability distribution represented by RNNLM is dependent on the entire word history, making the search space grow exponentially with the length of the recognition hypotheses. Therefore, RNNLM can usually be used only for N-best lattice rescoring, which then requires two-pass decoding. Here, we use a recently proposed approach to incorporate RNNLM into a weighted finite-state transducer (WFST)-based decoder by using an on-the-fly rescoring strategy [43]. This approach enables one-pass decoding without greatly increasing the computational cost. The algorithm is detailed in [43].

The RNNLM was trained using sentences extracted from the WSJ text corpora [28] distributed by the Linguistic Data Consortium (LDC), while ensuring that the sentences in the evaluation and development sets were not employed. The training data set for RNNLM consists of 716,951 sentences.

It is common practice when using RNNLM to compute the word probabilities by interpolating the probabilities from the RNNLM and a conventional trigram language model to enhance the word prediction performance. We also used this approach here and determined the interpolation coefficient using the development set.

4 Experiments

Before presenting the experimental results, we summarize the settings of our proposed SE front-end and ASR back-end. We then discuss the results of preliminary experiments on the development set that justify the implementation choices we made. Final results obtained with the evaluation set are provided in Section 4.4.

4.1 Experimental settings

Table 1 summarizes the main settings for the different components of the SE front-end. We discuss the impact that the choice of these parameters may have on the recognition performance in Section 4.2.

Table 2 details the experimental settings of the ASR back-end. To train the DNN we used an HMM state alignment obtained using the clean training data with an GMM-HMM based ASR system trained with the maximum likelihood (ML) criterion. Note that this can be performed because the training data was generated with simulation and therefore we have access to clean/reverberant stereo data. We may expect some decrease in performance if we would use reverberant speech to perform the alignment [44]. During SGD fine tuning, we used a validation set to decide when to reduce the learning rate and stop training. The validation set was created by randomly extracting 5 % of the training utterances.

We used two multi-condition training sets, namely the baseline training data set and the extended training data

Table 1 Settings for the SE Front-end

WPE
$T_{\perp}=3$, $T_{\top}=40, 30, 7$ for 1ch, 2ch and 8ch, respectively
Window length: 32 ms, frame shift : 8 ms
Number of FFT points: 512 (number of frequency bands: 257)
MVDR
Window length: 32 ms, frame shift : 8 ms
DOLPHIN see [22] Section V. A. 2) and Section V. B. 2) for details
Window length: 100 ms, frame shift : 25 ms
Spectral feature model:
GMM with 256 components, features: MFCC (13 dimensions)
Source location model:
Watson mixture model (4 components), features: normalized complex spectrum

set described in Subsection 3.2.1. Unless otherwise specified, all preliminary experiments were performed using the baseline training data set. For decoding we used two types of language model, a conventional trigram, which is provided with the WSJ corpus, and the RNNLM that was trained using the WSJ corpus. All the preliminary results were obtained with the trigram language model.

For unsupervised batch adaptation, we performed a few backpropagation iterations assuming that the labels obtained from a first recognition pass were true references.

Table 2 Settings for the ASR back-end

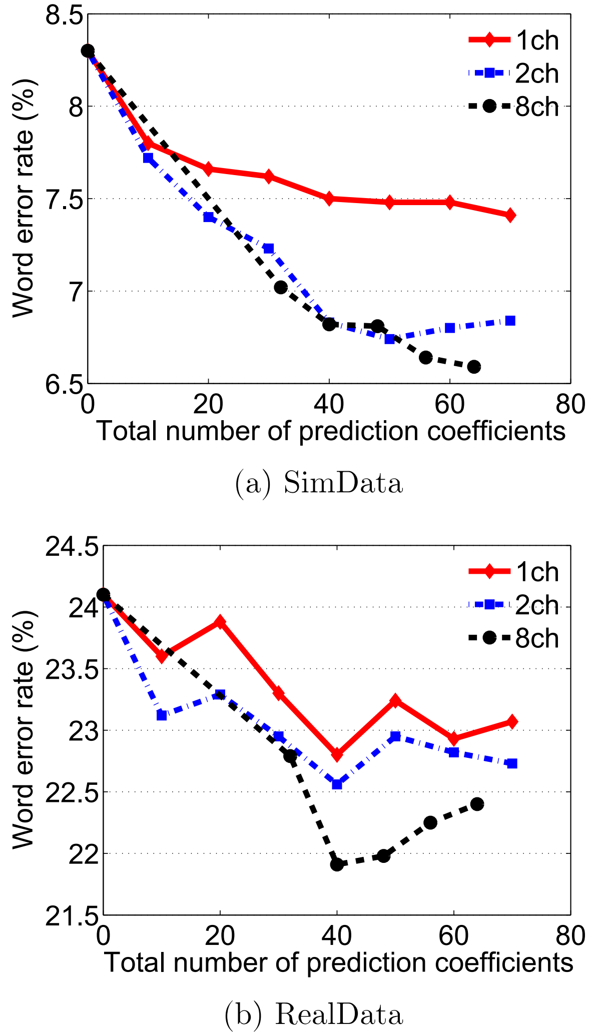
Input features
40 log mel filterbank coefficients + Δ + $\Delta\Delta$ (120 coef.)
Global mean and variance normalization + utterance level CMN
5 left and 5 right context (11 frames)
Acoustic model
DNN-HMM
7 hidden layers, 2048 hidden units,
1320 visible units, 3129 output units (HMM states)
Training data
(1) Baseline multi-condition training data (17h)
(2) Extended multi-condition training data (85h)
Language model
TRI : Trigram language model available with the WSJ corpus [28]
RNNLM : RNN-based language model (interpolation coef. 0.5)
Decoding parameters
Language model weight:11
Beam: 400

4.2 Preliminary experiments on SE front-end

We first investigate the influence on ASR performance of characteristics of the SE configurations, such as the prediction filter length, the scheme for reverberation reduction (linear filtering or spectral subtraction) and the processing order of WPE and MVDR. All parameters were tuned using the development set.

4.2.1 Influence of prediction filter length

Figure 2 plots the WER as a function of the total number of prediction coefficients of WPE for 1ch, 2ch and 8ch. The total number of prediction coefficients is the prediction filter order multiplied by the number of channels. Note that here 1 tap corresponds to 8 ms.



(a) SimData

(b) RealData

Fig. 2 Influence of prediction filter length. The figure plots the WER as a function of the total number of prediction coefficients of WPE for 1ch, 2ch, and 8ch for **a** SimData and **b** RealData, for the development set. The total number of prediction coefficients is the prediction filter order multiplied by the number of channels

For SimData, we observe that performance continues to improve up to a total of about 40 coefficients for 1ch and 2ch cases and longer for 8ch. These results suggest that for acoustic environments covered by the REVERB challenge, i.e., an RT60 up to about 700 ms, filter length of about 300 ms may be sufficient in practice. Longer filters do not significantly improve the average results, however, we observed that for sentences with a longer reverberation time, longer filters tend to provide better performance.

For RealData the performance peaks at 40 and degrades for longer filters, especially for the 8ch case. This may be due to the higher level of noise present in the RealData, which affects the accuracy of the prediction filters. In such a case, long prediction filters may amplify the background noise, resulting in degradation of the ASR performance.

We investigated the prediction filter length in detail after submitting our results to the REVERB challenge. The values that we used for the challenge correspond to filter lengths of 40, 30, and 7 taps for 1ch, 2ch, and 8 ch, respectively (this corresponds to total numbers of coefficients of 40, 60, and 56). Note that according to Fig. 2, these settings are not optimal for RealData, but for consistency with the system that we submitted to the REVERB challenge, we also used these settings for the other experiments described in this paper.

4.2.2 Linear filtering vs spectral subtraction

It is possible to reduce late reverberation components estimated by WPE using linear filtering or spectral subtraction as discussed in Section 3.1.1. Table 3 shows the WER as a function of the processing strategy used to reduce late reverberation for 1ch WPE. The results clearly show that linear filtering greatly outperforms spectral subtraction. Note that we also tested spectral subtraction in the amplitude domain and several parameters for spectral subtraction such as the over-subtraction factor but found consistently that spectral subtraction performed worse than linear filtering.

These results show that linear filtering, which causes less distortion than spectral subtraction, can lead to better recognition performance, confirming our intuition.

4.2.3 Processing order

When considering the physical model for reverberant speech as shown in Eq. 1, it would appear natural to perform denoising prior to dereverberation. Nevertheless, our proposed system performs dereverberation before denoising. There are two main reasons for choosing this processing order. First, model-based SE employs nonlinear processing that would destroy the linear filtering model of reverberation that WPE relies on. Second, WPE can process multi-channel signals while preserving source location information and outputs multi-channel signals. In contrast, the output of an MVDR beamformer is a single channel signal. Therefore, when applying MVDR before WPE, WPE cannot exploit spatial information provided by the multi-channel inputs, and performances are thus not optimal.

Table 3 WER for different strategies to remove late reverberations, i.e., linear filtering and power domain spectral subtraction (SS). The results are given for the development set

Processing	SimData	RealData
Distant	8.3 %	24.1 %
WPE(1ch) - *linear filtering*	7.5 %	22.8 %
WPE(1ch) - *SS*	7.8 %	23.9 %

We confirmed the above intuitions experimentally. Figure 3 plots the WER for different processing orders of WPE and MVDR. For SimData, WPE alone significantly outperforms MVDR and performing dereverberation before MVDR works better for both 2ch and 8ch cases. For RealData, we also observe that performing dereverberation prior to denoising works significantly better, although MVDR alone outperforms WPE alone. This may be due to the lower SNR of the RealData set, which makes denoising more relevant in this case.

Note that it is possible to use dereverberation before denoising because prediction filters estimated with WPE are not too sensitive to noise at the levels observed in the REVERB challenge.

4.3 Preliminary results on ASR back-end

Let us now analyze different factors influencing the ASR back-end such as the number of hidden layers and the size of the input context, the influence of training data, and the choice of adaptation strategy.

4.3.1 Influence of number of hidden layers and input context size

The results presented so far were obtained for DNN with 7 hidden layers and a context window of 11 frames. Here, we investigate the influence of the number of hidden layers and input context size on recognition performance.

Figure 4 plots WER as a function of the number of hidden layers for two input context settings (11 and 15 frames) for speech recorded by a distant microphone without an SE front-end. We clearly observe that the DNN with 7 hidden layers provides the best results, therefore we used that configuration for all other experiments.

Note that when considering reverberant speech recognition, we may consider using a longer context to handle reverberation [3]. We also tested increasing the acoustic context to 15 frames for numbers of hidden layers between 5 and 8, but we observed no performance improvement for SimData, and for RealData, in fact the performance even degraded. This indicates over-fitting to the training conditions that consist of simulated reverberant speech.

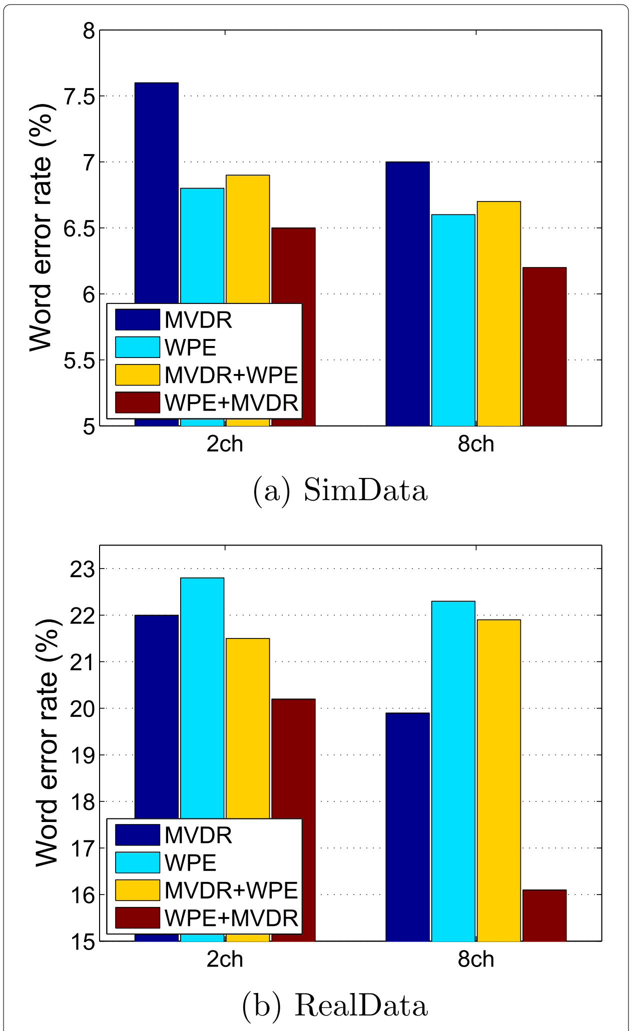


Fig. 3 Influence of processing order. The figure plots the WER for MVDR beamformer and WPE dereverberation and different combinations of both approaches for 2ch and 8ch for **a** SimData and **b** RealData, for the development set

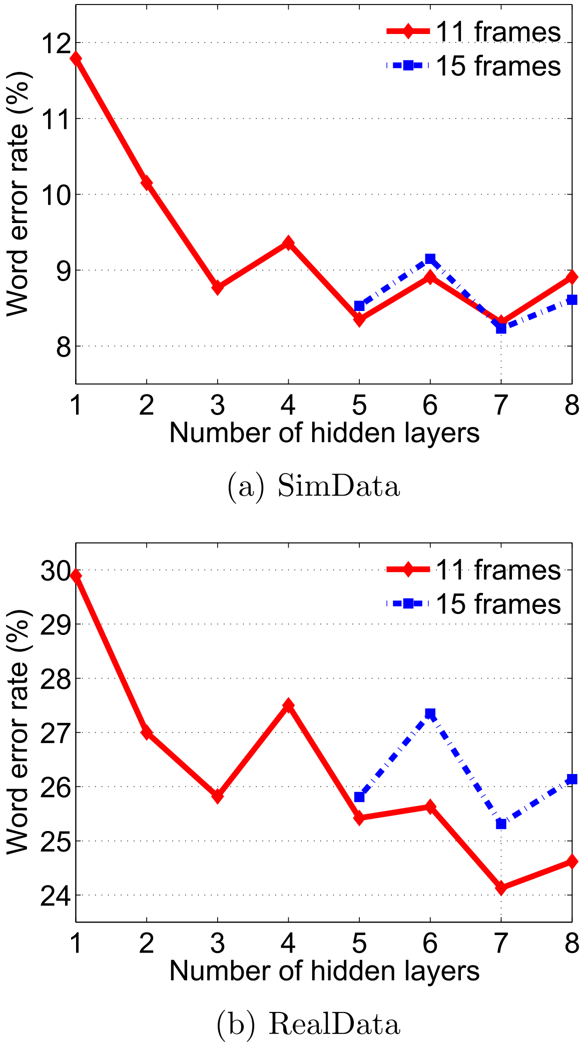


Fig. 4 Influence of the number of hidden layers and input context size. The figure plots the WER as a function of the number of hidden layers for context windows of 11 and 15 frames for **a** SimData and **b** RealData, for the development set

4.3.2 Training with distant speech vs enhanced speech

Table 4 shows the WER obtained with acoustic models trained with and without processing the multi-condition training data with the SE front-end. The first case consists of using a matched SE front-end for the training and testing conditions. In that case, the RBM pre-training and fine tuning were both performed using enhanced speech data.

The results in Table 4 clearly reveal that using enhanced training data improves the performance for SimData but greatly degrades the performance for RealData. This demonstrates again that the performance on SimData set is less sensitive to overfitting to the training data because the acoustic conditions are close to those of the multi-condition training data. The model trained with enhanced speech appears less robust and therefore performs poorly on RealData. Others have already reported that training

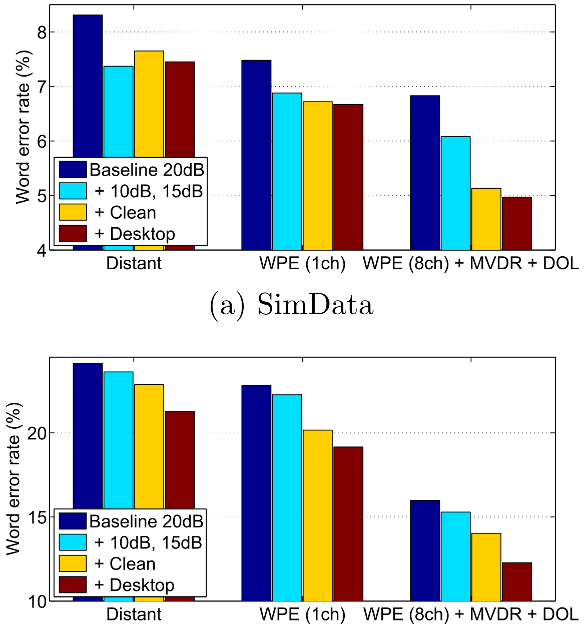
Table 4 WER when using acoustic models trained with ('✓') and without ('-') processing the training data with the SE front-end used during testing. The results are averaged over the acoustic conditions of SimData and RealData, for the development set

Processing	Train w/ SE front-end	SimData	RealData
Distant	-	8.3 %	24.1 %
WPE(1ch)	-	7.5 %	22.8 %
	✓	7.0 %	24.4 %
WPE(8ch) + MVDR	-	6.2 %	16.1 %
	✓	5.0 %	20.5 %

on enhanced speech may reduce the acoustic variations seen during training and therefore may negatively affect the generalization of the DNN [44–46]. Note that the results may differ if we use the extended multi-condition training data set that covers more acoustic variations as used in the next subsection. However, processing a large amount of training data with the SE front-end and training different acoustic models for each SE front-end is expensive. All subsequent results were obtained without processing the training data with the SE front-end.

4.3.3 Influence of training data

A simple approach for mitigating the mismatch between training and RealData and therefore alleviating the overfitting issue is to increase the number of acoustic conditions seen during training. Figure 5 plots the WER for different training data sets. We gradually increased the amount of training data by adding data with different acoustic variations to the multi-condition training data set. Each addition consists of exactly the same spoken utterances and the only differences relate to the acoustic conditions (reverberation, noise, or recording microphone). All the data in the extended training data set are part of WSJCAM0 [29] or can be generated using the tools provided by the REVERB challenge to generate multi-condition training data [25]. Note that this approach is related to previous work on data augmentation for DNN training such as vocal tract length perturbation (VTLP) [23, 24]. Indeed, we also artificially created modified versions of the training data while preserving the labels. The difference is that we modified the data by changing the SNR instead of modifying the vocal tract characteristics.



(a) SimData

(b) RealData

Fig. 5 Influence of extended training data. The figure plots the WER as a function of the training data used for **a** SimData and **b** RealData, for the development set. The results are given for different SE front-ends, i.e., no SE front-end ('Distant'), 1ch WPE ('WPE (1ch)') and the combination of 8ch WPE with MVDR and DOLPHIN ('WPE (8ch) + MVDR + DOL'). The *left bars* ('Baseline 20 dB') show the results for the baseline multi-condition training data set. The *other bars* show the results when the amount of training data is gradually increased by adding distant speech at SNRs of 10 and 15 dB ('+10 dB, 15 dB'), the clean training data ('+ clean'), and the training data recorded with a desktop microphone ('desktop')

The results in Fig. 5 clearly reveal that increasing the amount of training data improves the performance, especially for the RealData set. Adding clean data is particularly effective when using enhanced speech for recognition, especially when the SE front-end can greatly reduce reverberation and noise such as in the 8ch case. Moreover, although adding data recorded by the desktop microphone of WSJCAM0 does not improve the performance significantly for SimData, it contributes to a large improvement in the RealData case. This may be attributed to the benefit of using speech recorded with microphones with different characteristics. In addition, the use of real recordings obtained with a somewhat distant microphone in the training set may cover factors that are difficult to model with a simulation such as head movements. We used the last training set ('20 dB + 10 dB + 15 dB + clean + desktop') to obtain the optimal performance shown in the final results in Section 4.4.2. That extended training data set consists of approximately 5 times the amount of training data compared with the baseline multi-condition training data set of the REVERB challenge, i.e., about 85 h.

4.3.4 Unsupervised environmental adaptation

We have also investigated several strategies for adapting the DNN to the acoustic conditions. Figure 6 plots the WER as a function of the number of epochs for different adaptation strategies with the RealData. The results are given for the development set using an acoustic model trained with the baseline multi-condition training data. We compared four adaptation approaches namely, (1) retraining the weights and bias parameters associated with the last hidden layer of the DNN, (2) retraining only the first layer, (3) retraining the first two layers, (4) retraining all the parameters, and (5) training a global feature transformation, i.e., LIN [36]. LIN is implemented by inserting an additional hidden layer with linear activation between the input layer and the first hidden layer. The weights of the linear feature transformation are initialized as a unit matrix and are trained using adaptation data. Note that since LIN simply introduces a hidden layer with linear activation, the network adapted with LIN is structurally equivalent to the original network with re-trained

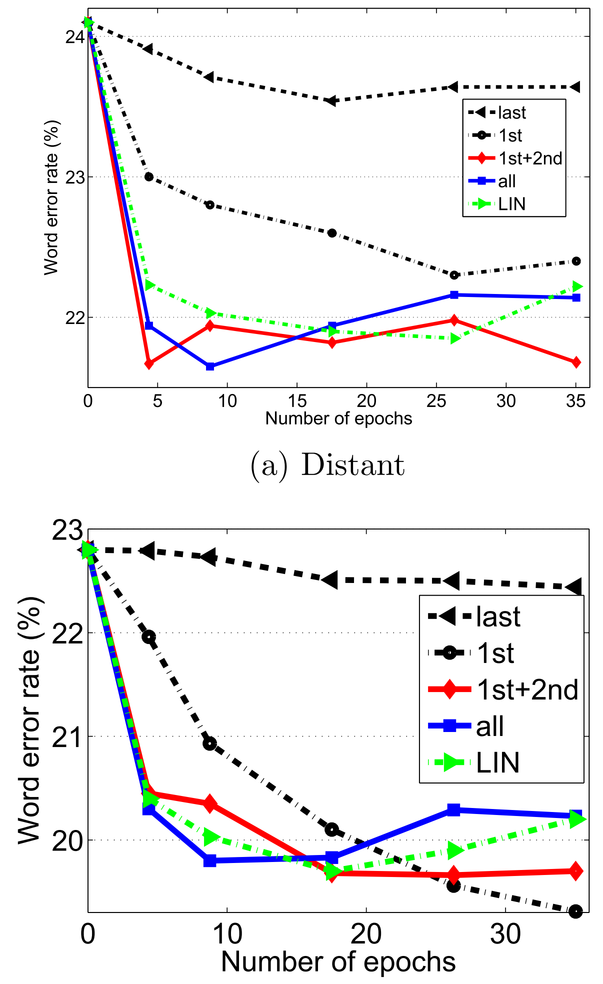


Fig. 6 Comparison of different adaptation strategies. The figure plots the WER as a function of the number of epochs for different adaptation strategies, i.e., (1) retraining the weights and bias parameters associated with the last hidden layer of the DNN ('last'), (2) retraining only the first layer ('1st'), (3) retraining the first two layers ('1st + 2nd'), (4) retraining all parameters ('all'), and (5) training a global feature transformation ('LIN'). The results are given for the RealData set of the development set, for (**a**) distant speech and for (**b**) speech processed with 1ch WPE

input weights. However, the matrix size of the linear transform is smaller, and the number of adaptation coefficients LIN requires is only about 65 % the number of coefficients of the first layer.

We tested different values for the learning rate and found that the best performance was obtained with a learning rate of 0.0001 for LIN and 0.0005 for the other adaption strategies. For all the experiments, we used a fixed momentum value, which was set at the same value as used for the DNN fine-tuning, i.e., 0.9.

In this experiment, we performed environmental adaptation by adapting the DNN using all the test data corresponding to a given acoustic condition after front-end processing. That corresponds to 'full batch mode' according to the instructions of the REVERB challenge [25]. Since several speakers are included in the test set of each acoustic condition, this process mostly performs environmental adaptation and only partly performs speaker adaptation. The adaptation data amounts to about 9 min for the results in Fig. 6.[4]

The results in Fig. 6 reveal that there is no large performance difference between retraining the first layer, retraining the first two layers, retraining the whole DNN, or using LIN. This shows that most of the gains from DNN retraining come from adaptation of the first hidden layers. Moreover, the performance is relatively stable after about 15 epochs except when retraining only the first layer. For all subsequent experiments, we performed adaptation by retraining the first and second hidden layers for about 15 epochs.

4.4 Final results for evaluation set

4.4.1 Results using baseline multi-condition training data set

Table 5 shows the WER for the evaluation set using an acoustic model trained with the baseline training data set. These results are provided to allow fair comparison with the other systems submitted to the REVERB challenge that employed the same training data set [47–50]. The results are detailed for all components of the SE front-end, for trigram (TRI) language model, and RNNLM and with and without adaptation.

Looking at the results of Table 5, we confirmed that unsupervised environmental adaptation and the use of RNNLM provide consistent improvement. As regards the SE front-end, dereverberation with WPE is responsible for most of the performance gains, with the relative improvement ranging from 15 to 29 %. The improvement is particularly large for the multi-channel cases. When combining dereverberation with noise reduction for the 8ch scenario (system VII-d), we obtained a relative improvement of up to 50 % compared with our system without any SE front-end (system 0-d). This demonstrates the potential of a well-designed multi-channel SE front-end to improve the performance of DNN-based ASR back-ends.

4.4.2 Results using extended multi-condition training data set

Table 6 shows the WER for the evaluation set when using the extended training data set. Note that the results in Table 6 differ slightly from those submitted to the REVERB challenge [26] because of minor differences in the configuration of the ASR back-end.

Table 5 Results for the evaluation set using an acoustic model trained with the baseline multi-condition training data set

				SimData							RealData		
	Proc.	LM	Adap.	Room1		Room2		Room3		Ave.	Room1		*Ave.*
				Near	Far	Near	Far	Near	Far	-	Near	Far	-
0 - a	Distant	TRI	-	6.1	6.7	7.4	11.2	7.8	12.2	8.6	29.2	27.8	28.5
b			✓	5.6	6.5	7.2	10.2	7.5	11.3	8.0	23.9	23.3	23.6
c		RNN	-	5.2	6.0	5.8	9.9	6.4	9.9	7.2	26.4	26.4	26.4
d			✓	4.7	5.8	6.0	9.5	6.7	10.1	7.1	21.6	22.3	22.0
I - a	WPE (1ch)	TRI	-	6.2	6.4	6.9	9.9	7.0	9.5	7.7	25.4	25.0	25.2
b			✓	5.4	5.8	6.2	9.1	6.4	8.6	6.9	20.1	20.2	20.1
c		RNN	-	5.1	5.5	5.2*	8.1	5.7	7.9	6.3	23.7	23.1	23.4
d			✓	4.7*	5.2*	5.4	8.0*	5.7*	7.7*	6.1*	18.6*	18.9*	18.7*
II - a	WPE (2ch)	TRI	-	6.3	6.3	6.7	8.5	6.7	8.4	7.1	23.5	22.9	23.2
b			✓	5.5	6.1	6.4	8.0	6.2	7.7	6.6	18.3	18.3	18.3
c		RNN	-	5.1	5.4	5.4	7.0	5.7	6.8	5.9	21.4	21.6	21.5
d			✓	4.7*	5.1	5.4	7.0	5.9	7.3	5.9	17.7	16.9	17.3
III - a	II + MVDR	TRI	-	6.3	6.6	6.5	7.3	6.1	7.4	6.7	21.1	20.5	20.8
b			✓	5.5	5.8	5.9	7.2	5.8	7.2	6.2	16.8	15.9	16.4
c		RNN	-	5.8	5.6	5.1	6.2	5.0	5.8	5.6	18.9	19.0	19.0
d			✓	4.8	5.0*	5.0*	6.3	5.3	6.3	5.4*	16.3	15.2	15.7
IV - a	III + DOL	TRI	-	6.6	6.8	6.4	7.1	5.8	7.0	6.6	20.3	18.7	19.5
b			✓	5.6	6.0	5.9	7.0	5.7	6.9	6.2	16.7	14.7*	15.7
c		RNN	-	5.7	5.5	5.0*	6.1*	4.8*	5.7*	5.5	17.9	18.2	18.1
d			✓	5.0	5.0*	5.0*	6.2	5.1	6.2	5.4*	15.3*	15.7	15.5*
V - a	WPE (8ch)	TRI	-	6.3	6.3	6.9	7.9	6.6	8.2	7.0	22.8	22.1	22.5
b			✓	5.6	5.9	6.4	7.3	6.0	7.6	6.5	17.4	17.1	17.2
c		RNN	-	5.3	5.5	5.1	6.1	5.4	6.9	5.7	21.9	20.5	21.2
d			✓	4.6*	4.9*	5.2	6.2	5.7	6.9	5.6	15.8	15.7	15.7
VI - a	V + MVDR	TRI	-	7.2	7.3	6.1	6.0	6.4	7.1	6.7	15.6	15.6	15.6
b			✓	5.5	6.1	5.6	5.5	5.9	6.5	5.8	12.5	12.7	12.6
c		RNN	-	6.2	6.3	4.8	5.3*	5.2*	5.7*	5.6	14.2	14.5	14.3
d			✓	5.1	5.1	4.6*	5.3*	5.3	5.8	5.2*	11.1	11.1*	11.1*
VII - a	VI + DOL	TRI	-	7.7	7.6	6.7	6.7	7.1	7.2	7.2	14.5	15.7	15.1
b			✓	5.9	6.3	5.8	5.8	6.0	6.6	6.0	12.2	13.9	13.0
c		RNN	-	6.7	6.1	5.6	5.6	5.9	6.4	6.1	12.2	13.9	13.0
d			✓	5.1	5.2	4.7	5.3*	5.4	5.8	5.3	9.8*	12.4	11.1*

*Best performance for 1ch, 2ch, and 8ch

The results are presented for the different components of the SE front-end and for different configurations of the ASR back-end, such as the language model (LM) used (trigram (tri) or RNNLM (RNN)) or with (✓) or without (-) adaptation

Table 6 Results for the evaluation set using an acoustic model trained with extended training data

				SimData							RealData		
	Proc.	LM	Adap.	Room1		Room2		Room3		Ave.	Room1		*Ave.*
				Near	Far	Near	Far	Near	Far	-	Near	Far	-
	Clean/Headset mic	RNN	✓	-	-	-	-	-	-	3.5	-	-	6.1
	Lapel mic	RNN	✓	-	-	-	-	-	-	-	-	-	7.3
0 - a	Distant	TRI	-	5.0	5.7	6.3	11.1	7.2	11.2	7.8	25.8	26.5	26.1
b			✓	4.5	5.4	6.1	9.9	7.0	10.2	7.2	19.7	22.1	20.9
c		RNN	-	4.2	5.0	5.3	9.2	5.7	9.7	6.5	23.0	24.4	23.7
d			✓	3.8	4.9	4.9	8.1	5.6	8.6	6.0	18.5	19.9	19.2
I - a	WPE (1ch)	TRI	-	4.8	5.4	6.2	8.8	6.1	8.6	6.6	20.3	20.3	20.3
b			✓	4.7	5.0	5.9	8.2	5.9	8.1	6.3	16.8	17.3	17.0
c		RNN	-	3.8	4.3	4.7	7.6	5.3	7.4	5.5	19.2	18.7	19.0
d			✓	3.6*	3.9*	4.4*	6.6*	4.8*	6.8*	5.0*	15.2*	16.7*	15.9*
II - a	WPE (2ch)	TRI	-	4.9	5.1	6.0	7.2	6.0	7.3	6.1	18.0	18.1	18.1
b			✓	4.6	4.9	5.8	6.8	5.8	6.7	5.8	14.5	16.0	15.2
c		RNN	-	4.0	4.3	4.9	6.1	5.0	5.8	5.0	16.5	16.5	16.5
d			✓	3.7*	4.0*	4.4	5.7	4.7	5.4	4.6	13.4	13.1	13.2
III - a	II + MVDR	TRI	-	4.9	5.1	5.8	6.8	5.7	6.7	5.8	15.6	15.9	15.8
b			✓	4.6	4.9	5.4	6.2	5.6	6.6	5.5	12.9	14.2	13.5
c		RNN	-	4.2	4.4	4.2	5.5	4.9	5.4	4.8	14.3	14.8	14.6
d			✓	3.9	4.2	4.1*	5.0*	4.5	5.0*	4.4*	11.6	12.8	12.2
IV - a	III + DOL	TRI	-	4.8	5.1	5.6	6.5	5.8	6.4	5.7	15.8	15.9	15.8
b			✓	4.7	5.0	5.4	6.1	5.6	6.2	5.5	12.8	14.0	13.4
c		RNN	-	4.1	4.7	4.2	5.2	4.9	5.4	4.7	14.4	14.6	14.5
d			✓	3.7	4.1	4.3	5.0*	4.4*	5.2	4.4*	11.1*	12.7*	11.9*
V - a	WPE (8ch)	TRI	-	4.6	5.1	5.8	6.5	5.8	6.8	5.8	16.7	17.0	16.9
b			✓	4.42	4.9	5.8	6.2	5.5	6.5	5.5	13.5	13.9	13.7
c		RNN	-	3.8	4.2	4.6	5.5	4.8	5.4	4.7	16.0	15.7	15.8
d			✓	3.5*	3.9*	4.1	5.0	4.3	5.3	4.3	12.7	13.1	12.9
VI - a	V + MVDR	TRI	-	4.8	5.2	5.2	5.5	5.5	5.9	5.3	11.9	12.4	12.2
b			✓	4.6	4.9	5.0	5.2	5.5	5.8	5.2	10.0	10.2	10.1
c		RNN	-	4.0	4.2	3.8	4.3	4.4	4.8	4.2	10.4	11.9	11.1
d			✓	3.7	3.9*	3.5*	4.2*	4.2*	4.9	4.1*	9.0	9.6	9.3
VII - a	VI + DOL	TRI	-	5.0	5.3	5.2	5.5	5.7	5.8	5.4	11.1	12.3	11.7
b			✓	4.8	4.8	5.0	5.3	5.6	5.6	5.2	9.7	9.9	9.8
c		RNN	-	4.1	4.2	3.9	4.3	4.3	4.8	4.3	10.0	11.4	10.7
d			✓	3.9	4.0	3.7	4.2*	4.3	4.6*	4.1*	8.9*	9.3*	9.1*

*Best performance for 1ch, 2ch and 8ch

The results are presented for the different components of the SE front-end and for different configurations of the ASR back-end, such as the language model (LM) used (trigram (tri) or RNNLM (RNN)) or with (✓) or without (-) adaptation

We observe the same tendency as in Table 5 but with an average performance that is about 1 point better for SimData and 2–5 points better for RealData. Here, we also observe large gains with our SE front-end, i.e., our best SE front-end achieved a relative performance improvement of 50 % over a strong DNN-based ASR back-end. This performance improvement should be compared to the improvement provided by the use of DNN instead of GMM for the acoustic models. Comparing the results of Table 5 for unprocessed speech with those reported in [48], we observe that our DNN-based system achieves about 10 % relative improvement over a state-of-the-art GMM-based recognizer (using Kaldi with fMMLR, BMMI training, MBR decoding, etc).

Table 6 also shows the WER for Clean/Headset microphone speech and for speech recorded with a label microphone. For SimData, the performance tends to saturate for all rooms to a level close to the WER of clean speech. For RealData, there still remains a large gap between our best performance and the performance obtained with a headset. Nevertheless, with 8ch, we achieve a WER close to that obtained using a lapel microphone.

Note that the performance difference between clean speech and headset microphone, which is close to clean speech, indicates that the performance gap between SimData and RealData does not solely originate from the different acoustic conditions but may also be due to other factors related to the spoken utterances such as speaking style. The fact that speaker adaptation techniques were shown to be very effective for the REVERB task [49, 51] is another indication that part of the mismatch may be due to speaker-related factors. These observations suggest that incorporating speaker adaptation techniques such as adding *i* vectors to the input features of our ASR back-end could provide additional gains [39, 49]. This will form part of our future investigations.

4.5 Discussions

4.5.1 Computational complexity and latency

When designing our system, our implementation choices were guided mainly by the objective of improving the recognition performance. In practice, the computational complexity and latency would also be important factors to consider when deploying such a recognition system. Currently, the front-end represents most of the computational cost. The real-time factors (RTFs) of WPE is about 0.2, 0.5, and 2.8 for 1ch, 2ch, and 8ch, respectively. The RTF of the MVDR beamformer is negligible (about 0.03) and that of DOLPHIN is about 6.1 and 10.5 for 2ch and 8ch, respectively. As for the latency, WPE and the MVDR beamformer operate in utterance-batch mode, therefore the latency corresponds to the length of a utterance, i.e., several seconds. DOLPHIN and environmental adaptation operate in full-batch mode, meaning that they process all data corresponding to an acoustic condition in a batch manner. Extension of WPE for online processing as well as reduction of the computational cost and latency of the components of our front-end is part of our future work.

4.5.2 Comparison with other approaches

Table 5 allows us to compare our results with those of other participants who used the same training data. Our 1ch system without adaptation (system I-c) performs slightly better than the second best system [47] on the RealData set. However, if we allow full-batch processing, with adaptation, we obtained 5 points of WER reduction (system I-d). For 8ch case, our system greatly outperforms the second best system by more than 4 points for utterance-based batch processing (system VI-c) and by 7.5 points for full-batch processing (system VI-d and VII-d). This confirms that even without using the extended training data set, our recognition system could achieve good performance.

There are many differences between the system used in [47] and our system that may explain the performance differences. However, both systems use similar SE front-ends, i.e., they both use beamforming and dereverberation. A major difference between the SE front-end in [47] and ours is that [47] starts with beamforming followed by single channel dereverberation. We used a different dereverberation approach that can exploit multi-channel signals and a MVDR beamformer after dereverberation. The results of Fig. 3 suggest that the fact that our dereverberation approach can exploit multi-channels may be one reason for our superior performances especially for the RealData 2ch and 8ch cases.

5 Conclusion

We have discussed strategies for designing a distant ASR system robust to reverberation. The system we present was developed for the REVERB challenge. It consists of an SE front-end, which uses long-term linear prediction-based dereverberation, an MVDR beamformer, and a model-based SE. We then employ an ASR back-end that uses neural networks for acoustic and language modeling and unsupervised environmental adaptation. We have analyzed the influence of the implementation parameters of our SE front-end, which revealed that using dereverberation prior to denoising and linear filtering to reduce reverberation were keys to achieving high recognition performance. Moreover, we have discussed ways to improve the recognition performance of the ASR back-end by using extended training data and adaptation. Our system achieved high performance in the REVERB challenge. In particular, using 8 microphones, we achieved recognition performance close to that obtained when using a lapel microphone.

Although our system was developed for the REVERB challenge, the strategies discussed in this paper could be effectively adopted in other distant speech recognition tasks. For example, [52] provides a detailed investigation of the various front-end techniques for DNN-HMM acoustic models in the AMI meeting transcription task. Moreover, it would also be possible to extend our strategies to deal with overlapping speech that is frequently observed in real meetings, by integrating other SE components such as blind source separation [53]. Future work directions may include the investigation of such complex acoustic scenes as well as further research to improve performance especially in the most challenging single microphone case.

Endnotes

[1] An acoustic condition is defined as the combination of a given room and a given microphone-speaker distance. The REVERB task consists of reverberant speech in 4 rooms over 2 speaker-microphone distances (near and far), i.e., 8 acoustic conditions in total. According to the REVERB challenge terminology [25], 'full-batch mode' means that all data for a given acoustic condition are processed in batch mode.

[2] Note that for the SE evaluation of the REVERB challenge, we also submitted an SE front-end that used an extension of vector Taylor series (VTS) based SE [21] instead of DOLPHIN. That approach performed particularly well in terms of subjective evaluation [27, 54].

[3] Note that LIN is a variant of the more recently reported feature discriminative linear regression (fDLR) [55, 56]. LIN simply performs a linear transformation of context expended input features.

[4] Note that the amount of adaptation data varies for the different test sets, i.e., it consists of 30 and 48 min for the development and evaluation sets of SimData and 9 and 18 minutes for the development and evaluation sets of RealData [17].

Competing interests

The authors declare that they have no competing interests.

Authors' information

Yotaro Kubo is now with Amazon and Takaaki Hori is now with Mitsubishi Electric Research Laboratories (MERL).

Received: 29 January 2015 Accepted: 27 June 2015
Published online: 19 July 2015

References

1. J Li, L Deng, Y Gong, R Haeb-Umbach, An overview of noise-robust automatic speech recognition. IEEE Trans. Audio Speech Lang. Process. **22**(4), 745–777 (2014)
2. R Haeb-Umbach, A Krueger, in *Techniques for Noise Robustness in Automatic Speech Recognition*, ed. by T Virtanen, R Singh, and B Raj. Reverberant speech recognition (Wiley, 2012). Chap. 10 ISBN: 978-1-119-97088-0
3. T Yoshioka, A Sehr, M Delcroix, K Kinoshita, R Maas, T Nakatani, W Kellermann, Making machines understand us in reverberant rooms: robustness against reverberation for automatic speech recognition. IEEE Signal Process. Mag. **29**(6), 114–126 (2012)
4. H Kuttruff, *Room Acoustics, Fifth Edition*. (Spon Press, Abingdon, Oxon, 2009)
5. PJ Moreno, Speech recognition in noisy environments. Technical report, Ph. D. Dissertation, CMU (Carnegie Mellon University 1996)
6. M Woelfel, J McDonough, *Distant Speech Recognition*. (John Wiley and Sons Ltd, UK, 2009)
7. K Eneman, J Duchateau, M Moonen, DV Compernolle, HV Hamme, in *Conference: 8th European Conference on Speech Communication and Technology, EUROSPEECH 2003 - INTERSPEECH 2003*. Assessment of dereverberation algorithms for large vocabulary speech recognition systems (Geneva, Switzerland, 2003)
8. I Tashev, in *in Proc. HSCMA'05*. Revereberation reduction for improved speech recognition (Piscataway, USA, 2005)
9. PA Naylor, ND Gaubitch, *Speech Dereverberation*, 1st edn. (Springer, Berlin, 2010)
10. BW Gillespie, LE Atlas, in *Proc. of ICASSP'03*. Strategies for improving audible quality and speech recognition accuracy of reverberant speech, vol. 1, (2003), pp. 676–679. doi:10.1109/ICASSP.2003.1198871
11. K Kinoshita, M Delcroix, T Nakatani, M Miyoshi, Suppression of late reverberation effect on speech signal using long-term multiple-step linear prediction. IEEE Trans. Audio Speech Lang. Process. **17**(4), 534–545 (2009)
12. T Nakatani, T Yoshioka, K Kinoshita, M Miyoshi, B-H Juang, in *Proc. of ICASSP'08*. Blind speech dereverberation with multi-channel linear prediction based on short time Fourier transform representation (Las Vegas, NV, 2008), pp. 85–88
13. A Krueger, R Haeb-Umbach, Model-based feature enhancement for reverberant speech recognition. IEEE Trans. Audio Speech Lang. Process. **18**(7), 1692–1707 (2010)
14. M Wolfel, Enhanced speech features by single-channel joint compensation of noise and reverberation. IEEE Trans. Audio Speech Lang. Process. **17**(2), 312–323 (2009)
15. A Sehr, R Maas, W Kellermann, Reverberation model-based decoding in the logmelspec domain for robust distant-talking speech recognition. IEEE Trans. Audio Speech Lang. Process. **18**(7), 1676–1691 (2010)
16. Y-Q Wang, MJF Gales, in *Proc. of ASRU'11*. Improving reverberant VTS for hands-free robust speech recognition (Hawaii, USA, 2011), pp. 113–118
17. K Kinoshita, M Delcroix, T Yoshioka, T Nakatani, E Habets, A Sehr, W Kellermann, S Gannot, R Maas, R Haeb-Umbach, V Leutnant, B Raj, in *Proc. of WASPAA'13*. The REVERB challenge: a common evaluation framework for dereverberation and recognition of reverberant speech (New Paltz, NY, USA, 2013)
18. T Yoshioka, T Nakatani, M Miyoshi, HG Okuno, Blind separation and dereverberation of speech mixtures by joint optimization. IEEE Trans. Audio Speech Lang. Process. **19**(1), 69–84 (2011)
19. T Yoshioka, T Nakatani, Generalization of multi-channel linear prediction methods for blind MIMO impulse response shortening. IEEE Trans. Audio Speech Lang. Process. **20**(10), 2707–2720 (2012)
20. M Souden, J Benesty, S Affes, On optimal frequency-domain multichannel linear filtering for noise reduction. IEEE Trans. Audio Speech Lang. Process. **18**(2), 260–276 (2010)
21. M Fujimoto, S Watanabe, T Nakatani, in *Proc. of ICASSP'12*. Noise suppression with unsupervised joint speaker adaptation and noise mixture model estimation (Kyoto, 2012), pp. 4713–4716
22. T Nakatani, T Yoshioka, S Araki, M Delcroix, M Fujimoto, Dominance based integration of spatial and spectral features for speech enhancement. IEEE Trans. Audio Speech Lang. Process. **21**(12), 2516–2531 (2013)
23. N Jaitly, GE Hinton, Vocal tract length perturbation (VTLP) improves speech recognition. Proc. ICML Workshop on Deep Learning for Audio, Speech and Language (2013)
24. X Cui, V Goel, B Kingsbury, in *Proc. of ICASSP'14*. Data augmentation for deep neural network acoustic modeling (Florence, 2014). pp. 5582–5586
25. REVERB Challenge. http://reverb2014.dereverberation.com. Cited Aug. 18 2014
26. M Delcroix, T Yoshioka, A Ogawa, Y Kubo, M Fujimoto, N Ito, K Kinoshita, M Espi, T Hori, T Nakatani, A Nakamura, in *Proc. of REVERB'14*. Linear prediction-based dereverberation with advanced speech enhancement and recognition technologies for the REVERB challenge (Florence, Italy, 2014). http://reverb2014.dereverberation.com/proceedings.html

27. M Delcroix, T Yoshioka, A Ogawa, Y Kubo, M Fujimoto, N Ito, K Kinoshita, M Espi, S Araki, T Hori, T Nakatani, in *Proc. of GlobalSIP'14*. Defeating reverberation: advanced dereverberation and recognition techniques for hands-free speech recognition (Atlanta, GA, 2014)
28. DB Paul, JM Baker, in *Proc. SNL'92*. The design for the Wall Street Journal-based CSR corpus (Association for Computational Linguistics Morristown, NJ, USA, 1992), pp. 357–362
29. T Robinson, J Fransen, D Pye, J Foote, S Renals, in *Proc. of ICASSP'95*. WSJCAM0: A British English speech corpus for large vocabulary continuous speech recognition (IEEE ICASSP'95 (Detroit,USA), 1995), pp. 81–84
30. M Lincoln, in *Proc. of ASRU'05*. The multi-channel Wall Street Journal audio-visual corpus (MC-WSJ-AV): Specification and initial experiments (San Juan, 2005), pp. 357–362
31. M Delcroix, T Hikichi, M Miyoshi, Precise dereverberation using multichannel linear prediction. IEEE Trans. Audio Speech Lang. Process. **15**(2), 430–440 (2007)
32. S Roweis, in *Proc. of EUROSPEECH'03*. Factorial models and refiltering for speech separation and denoising (Eurospeech 2003 (Geneva, Switzerland), 2003), pp. 1009–1012
33. O Yilmaz, S Rickard, Blind separation of speech mixtures via time-frequency masking. IEEE Trans. Signal Process. **52**(7), 1830–1847 (2004)
34. H Sawada, S Araki, S Makino, Underdetermined convolutive blind source separation via frequency bin-wise clustering and permutation alignment. IEEE Trans. Audio Speech Lang. Process. **19**(3), 516–527 (2010)
35. G Hinton, A practical guide to training restricted Boltzmann machines. Technical report (2010) ISBN:978-3-642-35288-1
36. J Neto, L Almeida, M Hochberg, C Martins, L Nunes, S Renals, T Robinson, in *Proc. of EUROSPEECH'95*. Speaker-adaptation for hybrid HMM-ANN continuous speech recognition system, (1995), pp. 2171–2174. http://hdl.handle.net/1842/1274
37. H Liao, in *Proc. of ICASSP'13*. Speaker adaptation of context dependent deep neural networks (New York, NY, USA, 2013), pp. 7947–7951
38. D Yu, K Yao, H Su, G Li, F Seide, in *Proc. of ICASSP'13*. KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition (ICASSP 2013 (Vancouver, Canada), 2013), pp. 7893–7897
39. G Saon, H Soltau, D Nahamoo, M Picheny, in *Proc. ASRU'13*. Speaker adaptation of neural network acoustic models using i-vectors (Olomouc, 2013), pp. 55–59
40. T Yoshioka, A Ragni, MJF Gales, in *Proc. of ICASSP'14*. Investigation of unsupervised adaptation of DNN acoustic models with filter bank input (ICASSP 2014 (Florence, Italy), 2014), pp. 6344–6348. http://www.icassp2014.org/home.html
41. S Xue, O Abdel-Hamid, H Jiang, L Dai, in *Proc of ICASSP'14*. Direct adaptation of hybrid DNN/HMM model for fast speaker adaptation in LVCSR based on speaker code (ICASSP 2014 (Florence, Italy), 2014), pp. 6339–6343. http://www.icassp2014.org/home.html
42. M Thomáš, Statistical language models based on neural networks (2012). PhD thesis, Brno University of Technology
43. T Hori, Y Kubo, A Nakamura, in *Proc. of ICASSP'14*. Real-time one-pass decoding with recurrent neural network language model for speech recognition (Florence, 2014)
44. M Delcroix, Y Kubo, T Nakatani, A Nakamura, in *Proc. of INTERSPEECH'13*. Is speech enhancement pre-processing still relevant when using deep neural networks for acoustic modeling? (Interspeech 2013 (Lyon, France), 2013), pp. 2992–2996. http://www.interspeech2013.org/
45. ML Seltzer, D Yu, Y Wang, in *Proc. of ICASSP'13*. An investigation of deep neural networks for noise robust speech recognition (Vancouver, BC, 2013), pp. 7398–7402
46. JT Geiger, F Weninger, JF Gemmeke, M Wollmer, B Schuller, G Rigoll, Memory-enhanced neural networks and NMF for robust ASR. IEEE Trans. Audio Speech Lang. Process. **22**(6), 1037–1046 (2014)
47. Y Tachioka, T Narita, F Weninger, S Watanabe, in *Proc. of REVERB'14*. Dual system combination approach for various reverberant environments with dereverberation techniques (Florence, Italy, 2014). http://reverb2014.dereverberation.com/proceedings.html
48. F Weninger, S Watanabe, JL Roux, JR Hershey, Y Tachioka, J Geiger, B Schuller, G Rigoll, in *Proc. of REVERB'14*. The MERL/MELCO/TUM system for the REVERB challenge using deep recurrent neural network feature enhancement (Florence, Italy, 2014). http://reverb2014.dereverberation.com/proceedings.html
49. MJ Alam, V Gupta, P Kenny, P Dumouchel, in *Proc. of REVERB'14*. Use of multiple front-ends and I-vector-based speaker adaptation for robust speech recognition (Florence, Italy, 2014). http://reverb2014.dereverberation.com/proceedings.html
50. X Xiao, S Zhao, DHH Nguyen, X Zhong, DL Jones, ES Chng, H Li, in *Proc. of REVERB'14*. The NTU-ADSC systems for reverberation challenge 2014 (Florence, Italy, 2014). http://reverb2014.dereverberation.com/proceedings.html
51. X Feng, K Kumatani, J McDonough, in *Proc. of REVERB'14*. The CMU-MIT REVERB challenge 2014 system: description and results (Florence, Italy, 2014). http://reverb2014.dereverberation.com/proceedings.html
52. T Yoshioka, MJF Gales, Environmentally robust {ASR} front-end for deep neural network acoustic models. Comput. Speech Lang. **31**(1), 65–86 (2015)
53. T Hori, S Araki, T Yoshioka, M Fujimoto, S Watanabe, T Oba, A Ogawa, K Otsuka, D Mikami, K Kinoshita, T Nakatani, A Nakamura, J Yamato, Low-latency real-time meeting recognition and understanding using distant microphones and omni-directional camera. IEEE Trans. Audio Speech Lang. Process. **20**(2), 499–513 (2012)
54. J Barker, E Vincent, N Ma, C Christensen, P Green, The PASCAL CHiME speech separation and recognition challenge. http://www.dcs.shef.ac.uk/spandh/chime/challenge.html. Cited April 24 2012
55. V Abrash, H Franco, A Sankar, M Cohen, in *Proc. of Eurospeech'95*. Connectionist speaker normalization and adaptation, (1995), pp. 2183–2186. doi:10.1109/72.182692
56. F Seide, G Li, X Chen, D Yu, in *Proc. of ASRU'11*. Feature engineering in context-dependent deep neural networks for conversational speech transcription (Hawaii, USA, 2011), pp. 24–29