THE MERL/MELCO/TUM SYSTEM FOR THE REVERB CHALLENGE USING DEEP RECURRENT NEURAL NETWORK FEATURE ENHANCEMENT

Felix Weninger[1,2*], *Shinji Watanabe*[1], *Jonathan Le Roux*[1], *John R. Hershey*[1], *Yuuki Tachioka*[3], *Jürgen Geiger*[2], *Björn Schuller*[2], *Gerhard Rigoll*[2]

[1] Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA 02139, USA
[2] MMK, Technische Universität München, 80290 Munich, Germany
[3] Information Technology R&D Center, Mitsubishi Electric Corp., Kamakura, 247–8501 Japan

ABSTRACT

This paper describes our joint submission to the REVERB Challenge, which calls for automatic speech recognition systems which are robust against varying room acoustics. Our approach uses deep recurrent neural network (DRNN) based feature enhancement in the log spectral domain as a single-channel front-end. The system is generalized to multi-channel audio by performing single-channel feature enhancement on the output of a sum-and-delay beamformer with direction of arrival estimation. On the back-end side, we employ a state-of-the-art speech recognizer using feature transformations, utterance based adaptation, and discriminative training. Results on the REVERB data indicate that the proposed front-end provides acceptable results already with a simple clean trained recognizer while being complementary to the improved back-end. The proposed ASR system with eight-channel input and feature enhancement achieves average word error rates (WERs) of 7.75 % and 20.09 % on the simulated and real evaluation sets, which is a drastic improvement over the Challenge baseline (25.26 and 49.16 %). Further improvements can be obtained by system combination with a DRNN tandem recognizer, reaching 7.02 % and 19.61 % WER.

Index Terms— De-reverberation, feature enhancement, recurrent neural networks, automatic speech recognition

1. INTRODUCTION

The REVERB Challenge [1] calls for automatic speech recognition (ASR) systems that are robust against reverberant environments with stationary noise. It is a notable trend that successful approaches for robust ASR in realistic conditions typically modify multiple parts of the basic ASR system, including multi-channel front-ends, feature extraction, unsupervised model adaptation to unseen test conditions, advanced acoustic modeling such as by deep neural networks (DNN), multi-condition training (MCT) with noisy data, and improved language modeling to take into account more context information. These techniques are usually found to be complementary to each other, and thus all have to be considered for optimal performance [2].

In line with previous successful systems for highly noise- and reverberation-robust ASR [3, 4], our approach combines multiple techniques for robustness. Apart from standard techniques including multi-condition and discriminative training (DT), adaptation, and feature transformations, we employ an advanced front-end that combines multi-channel processing, using direction of arrival estimation and subsequent beamforming, with single-channel spectral feature enhancement by a neural network. Deep and recurrent neural networks (DRNN) using Long Short-Term Memory (LSTM) are used, motivated by their recent success in ASR tasks in the past years – ranging from front-end enhancement to language modeling [5–7]. We will show that our front-end delivers drastic gains in ASR performance with a simple clean trained recognizer while being complementary to state-of-the-art back-end techniques. By investigating a late fusion approach, we also show that DRNN based feature enhancement and acoustic modeling deliver complementary performance gains. The next section is devoted to describing the components of our system in detail, before giving details of the experimental setup and outlining the results.

2. SYSTEM DESCRIPTION

2.1. Overview

Figure 1 shows a schematic overview of the proposed ASR techniques. Single- or multi-channel audio is transformed to the time-frequency domain. In case that multiple channels are available, the direct sound is enhanced by estimating the direction of arrival (cross-spectrum phase analysis, CSP) and subsequent delay-and-sum (DS) beamforming. The resulting complex spectrum is converted to a power spectrum and passed through a Mel filterbank. The logarithmic filterbank (Log FB) outputs are passed to a DRNN for enhancement. ASR features can be generated directly from the enhanced Log FB features, by applying feature transformations including DCT, unsupervised adaptation, etc. (cf. below). These ASR features are modeled by a GMM acoustic model (AM), whose likelihoods are combined with the language model (LM) for decoding. Alternatively, a DRNN AM can be used on top of enhanced Log FB features. In this case, the GMM and DRNN AMs are fused by a multi-stream HMM (Tandem approach).

2.2. Beamforming after DoA Estimation

To enhance the direct sound from the source, a frequency domain delay-and-sum beamformer is applied [8]. Given K microphones, the complex STFT spectra $\mathbf{z}_t(m), m = 1, \ldots, K$ are summed to the enhanced complex spectrum $\hat{\mathbf{z}}_t$,

$$\hat{\mathbf{z}}_t = \sum_m \mathbf{z}_t(m) \odot \exp(-\jmath \boldsymbol{\omega} \tau_{1,m}), \quad (1)$$

where t is the index of the current frame, $\odot$ is an element-wise multiplication, and $\boldsymbol{\omega}$ is a set of angular frequencies. The arrival time delay of the m-th microphone from the first microphone $\tau_{1,m}$ is

* Felix Weninger performed the work as an intern at MERL. Correspondence should be addressed to weninger@tum.de

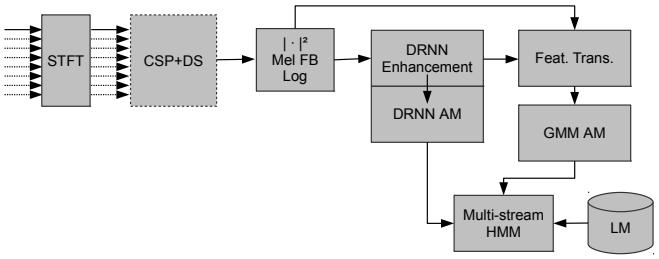


Fig. 1: Flowchart of the proposed system, using GMM and/or DRNN acoustic models (AM) after front-end enhancement.

related to the direction of arrival (DoA) and is estimated by the cross-spectrum phase (CSP) analysis, which uses a cross-power spectrum between two microphones [9] as

$$\tau_{1,m} = \arg\max \mathcal{S}^{-1} \left[\frac{\mathbf{z}_t(1) \odot \mathbf{z}_t(m)^*}{|\mathbf{z}_t(1)||\mathbf{z}_t(m)|} \right], \tag{2}$$

where $\mathcal{S}$ is the STFT operation and * denotes the complex conjugate. To improve the performance of the original CSP method, we used a peak-hold process [10] and noise component suppression, which sets the cross power spectrum to zero when the estimated SNR is below 0 dB [11]. Using three or more microphones reduces noise influence by synchronously adding pair-wise CSP coefficients [12]. For the purpose of further processing, the power spectrum $\hat{\mathbf{x}}_t = |\hat{\mathbf{z}}_t|^2$ is computed.

2.3. Single-Channel Feature Enhancement

In this study we use our spectral enhancement method based on deep neural networks introduced for de-reverberation in [13]. Enhancement is applied in the log-Mel domain, which tends to give better performance in deep neural network based ASR than the Mel-frequency cepstral coefficient (MFCC) domain [14].

To model the context needed for compensating late reverberation, we use bidirectional Long Short-Term Memory (LSTM) recurrent neural networks (RNNs), which deliver state-of-the-art performance in reverberation- and noise-robust ASR [15] and feature enhancement [5]. In the LSTM approach, an estimate $\tilde{\mathbf{y}}_t$ of the clean speech features $\mathbf{y}_t$ is computed from a sequence of observed speech features $\tilde{\mathbf{x}}_t \in \mathbb{R}^M$, $t = 1, \ldots, T$ by a non-linear mapping which is defined by the following iteration (*forward pass*) for levels $n = 1, \ldots, N$:

$$\mathbf{h}_0^{(1,\ldots,N)} := \mathbf{0}, \mathbf{c}_0^{(1,\ldots,N)} := \mathbf{0}, \tag{3}$$

$$\mathbf{h}_t^{(0)} := \tilde{\mathbf{x}}_t, \tag{4}$$

$$\mathbf{f}_t^{(n)} := \sigma(\mathbf{W}^{f,(n)}[\mathbf{h}_t^{(n-1)}; \mathbf{h}_{t-1}^{(n)}; \mathbf{c}_{t-1}^{(n)}; 1]), \tag{5}$$

$$\mathbf{i}_t^{(n)} := \sigma(\mathbf{W}^{i,(n)}[\mathbf{h}_t^{(n-1)}; \mathbf{h}_{t-1}^{(n)}; \mathbf{c}_{t-1}^{(n)}; 1]), \tag{6}$$

$$\mathbf{c}_t^{(n)} := \mathbf{f}_t^{(n)} \otimes \mathbf{c}_{t-1}^{(n)} + \mathbf{i}_t^{(n)} \otimes \tanh\left(\mathbf{W}^{c,(n)}[\mathbf{h}_t^{(n-1)}; \mathbf{h}_{t-1}^{(n)}; 1]\right), \tag{7}$$

$$\mathbf{o}_t^{(n)} := \sigma(\mathbf{W}^{o,(n)}[\mathbf{h}_t^{(n-1)}; \mathbf{h}_{t-1}^{(n)}; \mathbf{c}_t^{(n)}; 1]), \tag{8}$$

$$\mathbf{h}_t^{(n)} := \mathbf{o}_t^{(n)} \otimes \tanh(\mathbf{c}_t^{(n)}), \tag{9}$$

$$\tilde{\mathbf{y}}_t := \mathbf{W}^{(N+1)}[\mathbf{h}_t^{(N)}; 1]. \tag{10}$$

In the above, $\mathbf{h}_t^{(n)}$ denotes the hidden feature representation of time frame t in the level n units ($n = 0$: input layer). Analogously, $\mathbf{c}_t^{(n)}$, $\mathbf{f}_t^{(n)}$, $\mathbf{i}_t^{(n)}$, and $\mathbf{o}_t^{(n)}$ denote the dynamic cell state, forget gate, input gate, and output gate activations. $\mathbf{W}^{\cdot,(n)}$ denote the corresponding weight matrices at level n ($n = N + 1$: output layer). $\sigma(\cdot)$ and $\tanh(\cdot)$ are the (element-wise) logistic and hyperbolic tangent functions. For simplicity, we write $[\mathbf{a}; \mathbf{b}] := (\mathbf{a}^T, \mathbf{b}^T)^T$ for row-wise concatenation.

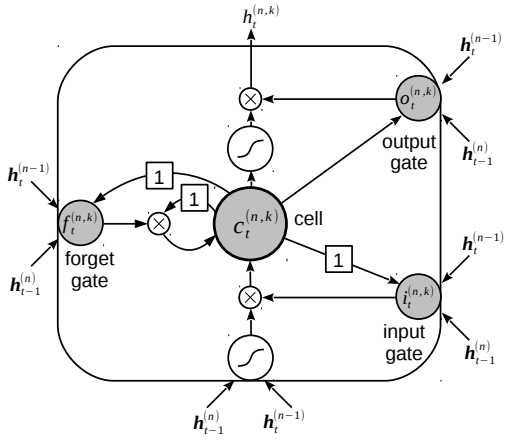


Fig. 2: Visualization of the k-th cell in the n-th layer of an LSTM-RNN. Arrows denote data flow and 1 denotes a delay of one timestep.

The cell state variables $\mathbf{c}_t^{(n)}$ serve to provide memory to the recurrent neural network [16], which is controlled by input and forget gates [17], denoted by $\mathbf{f}_t^{(n)}$ and $\mathbf{i}_t^{(n)}$ in Eqn. 7. The hidden layer activations correspond to the state variables, 'squashed' by the tanh activation function and scaled by the output gate activations (Eqn. 9). Figure 2 shows a visualization of a single LSTM cell (index k in layer n), which calculates its hidden activation $h_t^{(n,i)}$ from $\mathbf{h}_t^{(n-1)}$ and $\mathbf{h}_{t-1}^{(n)}$. $c_t^{(n,k)}$, $i_t^{(n,k)}$, $o_t^{(n,k)}$, $f_t^{(n,k)}$ denote the state, input gate, output gate, and forget activation of the cell k in layer n.

Note that in an LSTM-RNN, by applying Eqns. 5–9, the input features are weighted by coefficients calculated at run-time, instead of static coefficients as in a normal RNN. In turn, the matrices required for computing the coefficients are learned from data. This is done by minimizing the error $E\{\sum_t (\tilde{\mathbf{y}}_t - \mathbf{y}_t^*)^2\}$ on the training set, where $\mathbf{y}_t^*$ is a clean speech feature vector. Hence, our approach is similar to the de-noising auto-encoder (DAE) principle where hidden layers are trained to generate various levels of feature representations by mapping noisy input to clean output features [18]. Similar to previous studies on DAE in ASR [5, 19, 20], we directly use the output of the auto-encoder as features, not the hidden layer activations – this allows usage in a 'black-box' scenario where only the ASR feature extraction but not the ASR back-end is known.

In our approach, we use logarithmic filterbank features, i.e., $\tilde{\mathbf{x}}_t = \log \mathbf{M}\mathbf{x}_t$, $\mathbf{y}_t^* = \log \mathbf{M}\mathbf{y}_t$ where $\mathbf{M}$ is the matrix transforming power to Mel spectra, $\mathbf{x}_t$ is the power spectrum of the reference channel and $\mathbf{y}_t$ is the power spectrum of the clean speech. We also consider input features computed from beamformed power spectra $\hat{\mathbf{x}}_t$, i.e., $\tilde{\mathbf{x}}_t = \log \mathbf{M}\hat{\mathbf{x}}_t$.

By the transformation to the logarithmic Mel domain, the relationship between clean speech $\mathbf{y}_t$ and reverberated speech $\tilde{\mathbf{x}}_t$ becomes non-linear. However, it is known that deep neural networks can exploit such non-linear relationships in training [21]. We add delta coefficients to the filterbank features to capture dynamics at the feature level, which gives a slight performance gain.

For utterance-based processing in ASR, we can also exploit future

context within a sequence. This is done by adding a second set of layers which process the input feature sequences backwards, from $t = T$ to $t = 1$. This concept leads to bidirectional LSTM (BLSTM)-RNNs. In a deep BLSTM-RNN, activations from both directions are collected in a single activation vector before passing them on as inputs to the next layer [6].

2.4. GMM-HMM Back-End

With respect to the ASR baseline provided by the REVERB Challenge organizers, we add several state-of-the-art ASR techniques to improve the performance. All of them are implemented in the Kaldi toolkit [22] (cf. below). The REVERB baseline uses a simple GMM-HMM system with multi-condition training and Constrained Maximum Likelihood Linear Regression (CMLLR) adaptation to each test condition – the latter making it suitable only for batch processing. Since CMLLR can be viewed as a feature-space transformation, we will refer to it as fMLLR.

First, we use an advanced front-end considering context expansion ('frame stacking') and subsequent transformation by Linear Discriminant Analysis (LDA) and Semi-Tied Covariance (STC) matrices [23].

Second, after conventional ML training of the GMM parameters, we employ discriminative training by the boosted Maximum Mutual Information (bMMI) criterion. The training objective f_b for the acoustic model parameters λ is to maximize the mutual information between the correct transcription and the decoding lattice, given by

$$f_b(\lambda) = \log \sum_u \frac{p(\mathbf{X}^u|\lambda, h^{w_u^*})^\alpha p_L(w_u^*)}{\sum_{w_u} p(\mathbf{X}^u|\lambda, h^{w_u})^\alpha p_L(w_u) e^{-b\varrho(w_u, w_u^*)}}$$

where the outer sum is taken over all training utterances u, $\mathbf{X}^u$ denotes the acoustic features of utterance u, w_u^* is the reference transcription, w_u is a word hypothesis, and h^w denotes the HMM state sequence corresponding to the hypothesis w. The set $\{w_u\}$ in the denominator corresponds to the lattice of word hypotheses, and α is the acoustic model weight while p_L denotes the language model likelihood. $\varrho(w_u, w_u^*)$ denotes the phoneme accuracy of w_u with respect to the reference w_u^*. Thus, the term $e^{-b\varrho(w_u, w_u^*)}$ with a 'boosting factor' $b > 0$ emphasizes the weight of wrong hypotheses in the denominator calculation.

Third, we replace the full batch processing by the REVERB baseline by considering basis fMLLR [24] for adaptation, which performs well even on very small amounts of adaptation data [24]. In this approach, the acoustic features $\mathbf{X}$ of a speech utterance are transformed by a matrix $\mathbf{A}$,

$$\mathbf{X} \mapsto \mathbf{A}[\mathbf{X}; 1]$$

which itself is a linear combination of basis matrices:

$$\mathbf{A} = \mathbf{A}_0 + \sum_{i=1}^{B} \beta_i \mathbf{B}_i.$$

The matrices $\mathbf{B}_i$ are estimated by Princial Component Analysis (PCA) on statistics derived from the fMLLR matrices obtained on the training utterances. At test time, the number B of basis matrices is varied depending on the utterance length. To this end, the top B eigenvectors $\mathbf{b}_i$ (representing row-stacked matrices $\mathbf{B}_i$) are used as basis. It is shown in [24] that for any $B \leq M(M+1)$ this is equivalent to Maximum Likelihood estimation under reasonable assumptions. B is chosen proportial to the amount of frames in the utterance to be decoded, i.e., $B = \min\{\eta T, M(M+1)\}, 0 < \eta \ll 1$ where M is the acoustic feature dimension. The corresponding Maximum Likelihood coefficients β_i are estimated by Newton's method. Thus, if ηT is small compared to $M(M+1)$, the amount of adaptation parameters to be estimated is greatly reduced compared to conventional fMLLR (which requires an $M \times (M+1)$ matrix to be estimated). This enables robust adaptation on single utterances as short as three seconds [24].

Finally, instead of using the standard Maximum A-Posteriori (MAP) approach to speech decoding, we use Minimum Bayes Risk (MBR) decoding. In MBR decoding, we look for the hypothesis $\tilde{w}$ fulfilling

$$\tilde{w} = \arg\min_w \sum_{w'} P(w'|\mathbf{x}) D(w, w')$$

where $D(w, w')$ is the Levenshtein (edit) distance of w and w'. The intuition is that it is 'risky' to choose a hypothesis that is far from other hypotheses that are also likely given the model, and we want to minimize that risk. It can be shown that under assumption of model correctness, the above is equivalent to minimizing the expected word error rate (WER), while MAP corresponds to minimizing expected sentence error rate [25]. Thus, if we agree on WER as the ASR performance measure, MBR decoding will improve the results over standard MAP. As for MAP decoding, efficient approximations are needed since calculating the above sum requires exponential time. In [25], an efficient forward-backward algorithm operating on lattices is described, which we use in our study.

2.5. Tandem DRNN-GMM-HMM Back-End

As an extension to the GMM-HMM back-end, we consider DRNN acoustic modeling in a Tandem multi-stream HMM approach similar to [3]. A DRNN is trained on a frame-wise phoneme classification task, using an output layer with a softmax activation function. Thus, the output activations $\tilde{\mathbf{y}}_t$ of the DRNN correspond to pseudo posteriors $\tilde{\mathbf{y}}_t \in [0, 1]^P$ where P is the number of phonemes. From this, a frame-wise phoneme prediction $b_t \in \{1, \ldots, P\}$ is derived as

$$b_t = \arg\max_i \tilde{y}_{t,i}.$$

The phoneme prediction is decoded along with the MFCC feature vector in a multi-stream HMM. We obtain the joint probability of observing an MFCC feature vector and a DRNN phoneme prediction in the HMM state s_t as

$$p(\mathbf{x}_t, b_t|s_t) = p(\mathbf{x}_t|s_t)^\mu p(b_t|s_t)^{2-\mu}$$

where $\mu \in (0, 2)$ is the MFCC stream weight. The emission probabilities $p(\mathbf{x}_t|s_t)$ are modeled by conventional GMMs whereas the probabilities $p(b_t|s_t)$ are determined from the row-normalized state-phoneme confusion matrix on a held out part of the training data.

To integrate feature enhancement with phoneme recognition, instead of simply cascading both as in our previous work [3], we exploit the fact that feature enhancement is performed by a DRNN and stack the enhancement layers with the recognition layers, which allows backpropagation of the recognition error to the enhancement layers. More precisely, given a feature enhancement DRNN with N hidden layers, we first train the weight matrices $\mathbf{W}^{(N+2)}, \ldots, \mathbf{W}^{(N+N'+2)}$ of a DRNN with $N + N' + 1$ hidden layers (the output layer of the feature enhancement DRNN becomes the $N+1$-th hidden layer of the stacked DRNN). The error function is the cross-entropy between phoneme posteriors and phoneme labels. After convergence, *all* weight matrices $\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(N+N'+2)}$ are re-trained until convergence, using the same error function. Thus, the feature enhancement

network is re-trained discriminatively so as to enable good phoneme classification instead of optimizing the squared error in the enhanced features. Our 'stacking' procedure also resembles pre-training of the first N hidden layers in a de-noising auto-encoder scheme [18].

In our system, the GMM stream emission probabilities $p(\mathbf{x}_t|s_t)$ are always calculated using features before DRNN enhancement. In particular, the GMM model parameters exactly correspond to those obtained using multi-condition discriminative training (on beamformed data in the 8-channel case). This improved performance over using enhancement in both streams, probably because it makes both streams carry more complementary information.

3. EXPERIMENTAL SETUP

3.1. Evaluation Database

Our methods are evaluated on the official development and evaluation sets of the 2014 REVERB Challenge[1] [1]. Since the dataset is described in detail by [1], let us just summarize the most important figures. The task is to recognize read medium vocabulary (5 k) speech in eight different acoustic conditions, of which six are simulated by convolving the WSJCAM0 corpus [26] with three measured room impulse responses at near and far microphone distance, and adding stationary noise recordings from the same rooms at an SNR of 20 dB (SIMDATA). The other two conditions (REALDATA) correspond to real recordings of speakers in a reverberant meeting room at two microphone distances with ambient, stationary noise (mostly from the air conditioning), taken from the MC-WSJ-AV corpus [27]. The reverberation times (T60) range from 0.25 to 0.7 s, but are assumed to be unknown at test time. In all conditions, eight channels of a circular microphone array are available, of which the first is used as a reference channel and to train the GMM-HMM ASR back-end. We also investigate an ASR system that only uses the reference channel in signal enhancement. The SIMDATA set has 1 484 / 2 176 utterances from 20 / 20 speakers in the development and evaluation data. The REALDATA set has 179 / 372 utterances from five / ten speakers. For multi-condition ASR training and DAE training, the Challenge multi-condition training set is used, which also contains artificially distorted data similar to the SIMDATA set. It is of the same size as the clean WSJCAM0 training set, containing 7 861 utterances from 92 speakers. Impulse responses and noise types are chosen randomly, with equal probability.

Since the room impulse responses and noises in the training set, SIMDATA, and REALDATA sets differ, the setup of the Challenge provides a testbed to assess generalization of algorithms trained only on artificial data to real-life conditions which are of similar nature, but whose exact parameters are unknown.

3.2. ASR Back-End

The REVERB Challenge baseline is implemented in HTK [28]. Since many state-of-the-art ASR techniques are not available in HTK, yet we want to stick to an open-source ASR back-end for maximum reproducibility, we re-implemented the REVERB Challenge baseline in Kaldi [22], which delivers almost the same results as the official baseline (cf. below).

The GMM-HMM ASR back-end uses an LDA-STC front-end where nine consecutive frames of 13 MFCC features (coefficients 0–12) are reduced to 40 components. STC transforms are estimated after every other iteration of model training up to iteration 10. The standard 5 k WSJ bigram and trigram language models are used. In most of our experiments, the language model weight is fixed at $1/\alpha = 15$. For the Tandem back-end, the stream weight is set to $\mu = 1.2$.

A clean triphone recognizer is trained on the WSJCAM0 training set, while a multi-condition triphone recognizer is trained by repeating the GMM-HMM training steps using the REVERB multi-condition training set, including bMMI training of the GMMs. In case of multi-condition training, LDA and STC matrices are estimated on multi-condition training data while they are trained on clean data for the clean recognizer. Similarly, the fMLLR bases $\mathbf{B}_i$ are estimated on either the clean or multi-condition training set. To use multi-condition training with front-end enhancement (beamforming and DRNN), the multi-condition set is processed by the same enhancement steps. Using the original multi-condition training set in combination with enhancement delivered inferior results, sometimes even falling below the clean training result.

3.3. Network Training

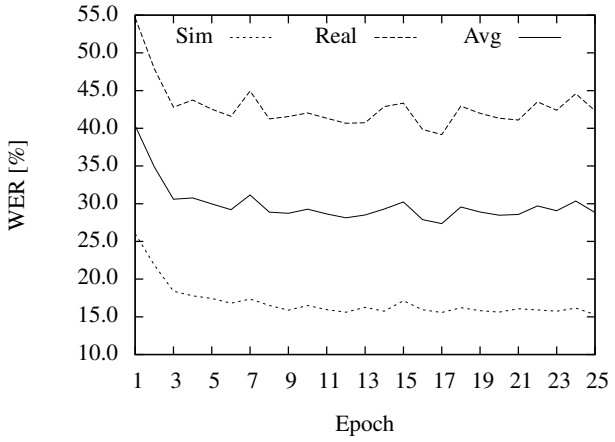
The weight matrices for DRNN feature enhancement are estimated on the task to map the multi-condition set of the REVERB Challenge to the clean WSJCAM0 training set, in a frame-by-frame manner. To obtain the training inputs for the network in the eight-channel case, the beamformer is applied on the multi-condition set. Log Mel filterbank outputs with 23 Mel bands are used as input and output features, and mean normalization is performed per utterance, so that cepstral mean normalized (CMN) MFCCs can be obtained simply by applying a discrete cosine transformation (DCT) to the DRNN output $\tilde{\mathbf{y}}_t$. In practice, performing another CMN after DCT turned out to give better performance, because the actual network outputs tend not to have exact zero mean. Input and output features are globally variance normalized on the training set. Thus, all feature transformations at test time only need the current test utterance ('utterance-based processing').

The network topology used in this study was determined based on earlier feature enhancement experiments on the CHiME Challenge data [5] and limited tuning on the REVERB development set. In the case of beamformed input, networks have two hidden layers each consisting of 128 LSTM units for each direction ($N = 2$). For single-channel input, an additional hidden layer is used ($N = 3$), reflecting the fact that the single-channel enhancement task is arguably more complex. All weights are randomly initialized with Gaussian random numbers ($\mu = 0$, $\sigma = 0.1$).

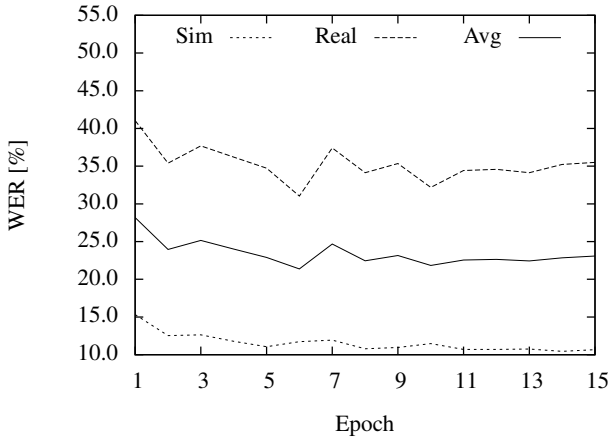
We train the DRNNs through stochastic on-line gradient descent with a learning rate of 10^{-6} and a momentum of 0.9. Weights are updated after 'mini-batches' of 50 utterances (feature sequences). Zero mean Gaussian noise ($\sigma = 0.1$) is added to the inputs in the training phase. An early stopping strategy is used to minimize overfitting, by evaluating an error function on the development set for each training epoch and selecting the best network accordingly. The sum of squared errors, which is used in backpropagation, cannot be evaluated on the development set by the Challenge guidelines, as it would require using the clean development data. Thus, instead the ASR performance in terms of WER is used directly. We train the networks for a maximum of 50 epochs and measure the WER with the clean trained LDA-STC recognizer and enhanced features on the SIMDATA and REALDATA development for every training epoch. The best network in terms of the sum of the SIMDATA and REALDATA WERs is used as the final network. We found that the optimal performance is obtained after only 6 epochs for the beamformed input, and after 17 epochs for the single-channel input. The corresponding WER curves are shown in Figures 3a and 3b.

For DRNN acoustic modeling, we use three hidden layers ($N' =$

[1] http://reverb2014.dereverberation.com/



(a) 1-channel input



(b) 8-channel (BF) input

Fig. 3: DRNN feature enhancement training: WER curves on development set, clean training, LDA-STC, no adaptation, tri-gram LM

3) with 50 LSTM units for each direction on top of the enhancement layers. A learning rate of 10^{-5} and input noise with standard deviation $\sigma = 0.6$ is used. Phoneme alignments are obtained by the LDA-STC recognizer on the Challenge multi-condition training set. Early stopping is done on the frame-wise phoneme error obtained on a held out part of the multi-condition training data, consisting of utterances from 10 speakers (it is not allowed to use the phoneme error on the development data, as per Challenge regulations). Our GPU enabled DRNN training software is publicly available[2].

3.4. Search parameter optimization

For the final systems, we tune the language model weight on the (unweighted) average WER on the development SIMDATA and REALDATA sets, by rescoring decoding lattices accordingly. The best development set language model weight (without adaptation) is also used for obtaining the first-pass hypothesis in fMLLR transformation estimation. We also increase the width of the beam search both in first-pass decoding and lattice generation to avoid search errors which might cause performance drops.

Table 1: Results on the REVERB development set, on SIMDATA and REALDATA. GMM-HMM recognizers, single-channel MFCC front-end, without pre-processing. bg/tg: bi-gram/tri-gram language modeling. [1] batch fMLLR per test condition. [2] basis fMLLR per utterance.

LDA -STC	fMLLR	MCT	DT	LM	MBR	WER [%] SIM	WER [%] REAL
REVERB Challenge baselines							
✗	✗	✗	✗	bg	✗	51.86	88.38
✗	✗	✓	✗	bg	✗	28.94	52.29
✗	✓[1]	✓	✗	bg	✗	25.16	47.23
Our baselines							
✗	✗	✗	✗	bg	✗	51.23	88.81
✗	✗	✓	✗	bg	✗	28.62	54.04
✗	✓[2]	✓	✗	bg	✗	23.60	47.14
✓	✗	✗	✗	bg	✗	48.22	91.72
✓	✗	✓	✗	bg	✗	23.41	47.80
✓	✓[2]	✓	✗	bg	✗	19.42	41.42
✓	✗	✓	✓	bg	✗	17.34	46.48
✓	✓[2]	✓	✓	bg	✗	15.53	40.60
✓	✓[2]	✓	✓	tg	✗	12.28	31.05
✓	✓[2]	✓	✓	tg	✓	**12.05**	**30.73**

3.5. System combination

For system combination, we use the Recognizer Output Voting Error Reduction (ROVER) scheme implemented in NIST's scoring toolkit[3]. First, 1-best hypotheses with word level posteriors ('confidences') are generated from the decoding lattices of each system. Then, alignment of the hypotheses is performed by dynamic programming. Finally, for each aligned segment a weighted majority vote is taken.

4. RESULTS AND DISCUSSION

4.1. Baseline ASR results

In a first step, we assess the performance gained by replacing the Challenge recognizer by more up-to-date acoustic and language modeling, as well as adaptation. The first three rows in Table 1 show the results obtained by the Challenge recognizer, and the next three rows those by the equivalent 'Kaldi baselines'. These systems correspond to clean training, multi-condition training, and multi-condition training plus fMLLR adaptation. The only difference is that the Kaldi system using fMLLR does so on an utterance level by basis fMLLR, which actually improves performance on the SIMDATA. As expected, there is no significant difference in the other results.

The three subsequent rows show the same systems but with LDA-STC feature transformation. It can be seen that in combination with multi-condition training and/or adaptation, a significant performance gain is obtained (roughly 5-6 % absolute). The result with clean training and LDA-STC (48.22 % / 91.72 % WER) is given for completeness and as a clean recognizer baseline for evaluation of feature enhancement.

Performing MCT with the bMMI criterion (discriminative training) gives another boost on SIMDATA (6 % absolute without adapta-

[2] https://sourceforge.net/p/currennt

[3] ftp://jaguar.ncsl.nist.gov/pub/sctk-1.2c.tgz

Table 2: WER obtained by single- and multi-channel front-ends with and without feature enhancement, on the REVERB development set. ASR back-ends using with LDA-STC and basis fMLLR, tri-gram LM, MBR. # ch: number of channels; Enh: DRNN enhancement. Optimized: tuning of LM weight and beam width in decoding. [1] bMMI training using clean data.

WER [%]		**Back-End**							
Front-End		Clean trained		MCT		+ bMMI		Optimized	
# ch	Enh	SIM	REAL	SIM	REAL	SIM	REAL	SIM	REAL
1	✗	33.21	77.76	14.92	35.20	12.05	30.73	11.22	30.77
1	✓	13.99	35.03	13.51	32.69	10.77	28.30	10.44	26.30
8	✗	16.42	54.49	**9.77**	26.34	7.94	23.82	**7.49**	23.91
8	✓	**9.72**	**26.49**	9.94	**24.25**	**7.91**	**22.04**	7.67	**21.39**
Oracle		*5.96*	*10.12*	–	–	*5.01*[1]	*10.12*[1]	*5.06*[1]	*9.91*[1]

tion, 4 % absolute with adaptation), but only a slight gain (≈ 1 %) on REALDATA, probably because of the mismatched condition.

Next, choosing a tri-gram language model instead of a bi-gram drastically improves performance by about 21/24 % relative on SIMDATA / REALDATA. This shows the importance of adding domain knowledge to achieve increased robustness. Finally, using MBR decoding slightly improves WER both on SIMDATA and REALDATA.

All in all, the performance gain just by improving the ASR back-end are quite impressive, resulting in 52 % / 35 % relative reduction in WER compared to the multi-condition / fMLLR REVERB baseline. This improved ASR back-end ('Kaldi baseline') is made publicly available under the Apache 2.0 open-source license [4].

4.2. Results with beamforming and/or spectral enhancement

Table 2 shows the results with single- and multi-channel enhancement. We first use a clean recognizer (with fMLLR adaptation) to show how well the front-end performs in a recognizer that has never seen noisy data in training (recall that for the clean recognizer, the fMLLR basis is computed on clean data as well). It can be seen that DRNN enhancement on the single-channel input gives reasonable results with the clean recognizer, significantly outperforming beamforming on its own. However, combination of both in a straightforward cascade gives by far the best result, improving by 71 % / 66 % relative on SIMDATA and REALDATA over the baseline without front-end processing.

When the back-end is trained with multi-condition data, DRNN enhancement improves for all cases except the 8-channel SIMDATA. Furthermore, if we optimize the search parameters, the performance difference between features with and without DRNN enhancement becomes larger on REALDATA (both for 1-channel and 8-channel). This probably shows that for DRNN enhanced features, there are some search errors in the baseline ASR due to a different dynamic range of acoustic model likelihoods. Furthermore, we found that they gave generally higher acoustic likelihoods, which requires adjusting the LM weight.

All in all, while the performance gains by enhancement are notable, they are still far from the performance obtained in clean conditions, especially on REALDATA. It has to be noted, though, that the ASR task of REALDATA itself seems much harder than the SIMDATA task, when eliminating reverberation and noise as confounding factors – the WER in clean conditions is about twice as high. This could be due to a mismatch in accent and/or speaking style, since the MC-WSJ-AV corpus was recorded in a different site and dialect region than the WSJCAM0 corpus, and a different recording protocol was used (speakers standing in a meeting room, rather than sitting in a sound-proof booth).

[4] `http://www.mmk.ei.tum.de/~wen/REVERB_2014/kaldi_baseline.tar.gz`

4.3. Test set evaluation and system combination

Table 3 shows the detailed results on the evaluation set obtained by the 1-channel and 8-channel systems, with and without DRNN enhancement. In the 1-channel case, our best result (submitted to the Challenge) is 10.21 / 26.73 % WER on the evaluation set. Comparing the results obtained at the 'near' microphone distance with the 8-channel front-end on SIMDATA with the oracle results, we observe that the performance is already quite close (to be fair, one has to look at the GMM-HMM results). However, at the 'far' distance a significant difference remains. The results on REALDATA are less encouraging, due to the aforementioned mismatch in training and test data.

To investigate whether DRNN enhancement and acoustic modeling are complementary, let us first outline the results with the Tandem DRNN+GMM-HMM recognizer. It obtains 6.48 / 7.28 % WER on SIMDATA (development / evaluation set), which is our best single system result for SIMDATA. However, the performance on REALDATA is lower than the one of the DRNN enhancement GMM-HMM system. This is arguably due to the discriminative training of the DRNN acoustic model, which leads to a better modeling of the SIMDATA which is close to the training data while worsening the results on the mismatched REALDATA – recall that we made similar observations for ML vs. bMMI GMM training, cf. Table 1.

We now investigate the combination of two or three of the 8-channel systems. As it seems, the best combination on the development set is the DRNN acoustic model with the DRNN enhancement system. This combination achieves the best development set WER on SIMDATA and REALDATA, indicating that both are complementary approaches although they use similar modeling techniques (involving a DRNN and a GMM). This combination also achieves the best average WER on the evaluation set, reaching down to 7.02 and 19.61 % WER on SIMDATA and REALDATA.

5. CONCLUSIONS AND OUTLOOK

We presented the MERL/MELCO/TUM system for the REVERB Challenge using a combination of beamforming, single-channel feature enhancement and acoustic modeling, the latter two by DRNNs. The system architecture allows both multi-channel and single-channel processing. In particular, the proposed integration of (physical) model based multi-channel and data based single-channel processing has the advantage that the models do not have to be re-trained for different microphone array setups (as would be the case if we just concatenated the input features from the eight channels for DRNN enhancement).

Table 3: REVERB development and evaluation set results for selected 1-channel and 8-channel systems, as well as system combination. Evaluation set results are given per room and microphone distance (near / far). All with MCT, LDA-STC, basis fMLLR, bMMI, tri-gram LM, MBR, optimized search parameters. BF: beamforming. Enh: DRNN enhancement.

WER [%]	**Dev. set avg.**		**Evaluation set**									
	SIM	REAL	SIMDATA							REALDATA		
			Room 1		Room 2		Room 3		Avg	Room 1		Avg
			near	far	near	far	near	far		near	far	
1-channel systems												
REVERB baseline	25.16	47.23	16.23	18.71	20.50	32.47	24.76	38.88	25.26	50.74	47.57	49.16
GMM-HMM	11.22	30.77	6.37	7.67	8.76	16.22	10.66	20.20	11.65	31.84	30.93	31.39
GMM-HMM (Enh)	**10.44**	**26.30**	6.39	7.52	8.41	14.15	9.47	15.30	**10.21**	25.39	28.06	**26.73**
8-channel systems (BF)												
GMM-HMM (I)	7.49	23.91	5.39	5.93	6.38	9.71	6.87	12.47	7.79	20.25	23.16	21.71
DRNN+GMM-HMM (II)	6.48	22.07	5.32	5.76	6.19	9.00	6.65	10.78	7.28	19.74	23.63	21.69
GMM-HMM (Enh) (III)	7.67	21.39	5.49	6.12	6.80	9.69	7.13	11.28	7.75	17.66	22.52	20.09
8-channel systems (BF) – System combination												
ROVER I+II	6.58	22.60	5.00	5.44	6.04	8.95	6.70	11.04	7.20	19.26	22.01	20.64
ROVER II+III	**6.44**	**20.24**	5.08	5.66	5.95	8.58	6.72	10.10	**7.02**	16.96	22.25	**19.61**
ROVER I+III	7.15	21.20	5.29	5.98	6.24	9.34	6.98	11.19	7.50	17.57	21.10	*19.34*
ROVER I+II+III	6.75	21.62	5.30	5.88	6.06	9.00	6.89	11.07	7.37	17.76	22.62	20.19
Oracle enhancement												
GMM-HMM	*5.06*	*9.91*	*5.34*		*5.55*		*6.07*		*5.65*	*8.47*		

Our system has been designed to allow utterance based processing, but needs multiple recognition passes at this stage. It is thus suitable, e.g., for server-based ASR systems. Most of the system components could be used in an on-line ASR system as-is or with small modifications. The CSP+DS front-end is fully on-line capable. Low-latency enhancement could be done by using unidirectional DRNNs (possibly with a small window of future context); it remains to evaluate the performance of this setup.

From our results, it is evident that there is a fundamental limitation to the performance of training-based approaches in a mismatched condition setup, such as the REALDATA scenario. In a practical application one could perform semi-supervised training of the DRNN enhancement and acoustic model on 'field data' (in the Challenge scenario, this would be other data from the MC-WSJ-AV corpus, which was not allowed to be used). This, along with other methods to improve generalization, such as weight noise for DRNNs [29], will be a promising direction for future research.

6. REFERENCES

[1] K. Kinoshita, M. Delcroix, T. Yoshioka, T. Nakatani, E. Habets, R. Haeb-Umbach, V. Leutnant, A. Sehr, W. Kellermann, R. Maas, S. Gannot, and B. Raj, "The REVERB Challenge: A common evaluation framework for dereverberation and recognition of reverberant speech," in *Proc. of WASPAA*, New Paltz, NY, USA, 2013.

[2] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, "The second CHiME speech separation and recognition challenge: an overview of challenge systems and outcomes," in *Proc. of ASRU*, Olomouc, Czech Republic, 2013.

[3] J.T. Geiger, F. Weninger, A. Hurmalainen, J.F. Gemmeke, M. Wöllmer, B. Schuller, G. Rigoll, and T. Virtanen, "The TUM+TUT+KUL approach to the CHiME Challenge 2013: Multi-stream ASR exploiting BLSTM networks and sparse NMF," in *Proc. of 2nd CHiME Workshop*, Vancouver, Canada, 2013, pp. 25–30, IEEE.

[4] Y. Tachioka, S. Watanabe, J. Le Roux, and J. Hershey, "Discriminative methods for noise robust speech recognition: A CHiME challenge benchmark," in *Proc. of 2nd CHiME Workshop*, 2013, pp. 19–24.

[5] F. Weninger, J. Geiger, M. Wöllmer, B. Schuller, and G. Rigoll, "Feature enhancement by deep LSTM networks for ASR in reverberant multisource environments," *Computer Speech and Language*, 2014, to appear, doi:10.1016/j.csl.2014.01.001.

[6] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. of ICASSP*, Vancouver, Canada, 2013, pp. 6645–6649, IEEE.

[7] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling," in *Proc. of INTERSPEECH*, Portland, OR, USA, 2012.

[8] D. Johnson and D. Dudgeon, *Array Signal Processing*, Prentice-Hall, 1993.

[9] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, pp. 320–327, 1976.

[10] T. Suzuki and Y. Kaneda, "Sound source direction estimation based on subband peak-hold processing," *The Journal of the Acoustical Society of Japan*, vol. 65, no. 10, pp. 513–522, 2009.

[11] Y. Tachioka, T. Narita, and T. Iwasaki, "Direction of arrival estimation by cross-power spectrum phase analysis using prior distributions and voice activity detection information," *Acoustical Science and Technology*, vol. 33, pp. 68–71, 2012.

[12] T. Nishiura, T. Yamada, T. Nakamura, and K. Shikano, "Localization of multiple sound sources based on a CSP analysis with a microphone array," in *Proc. of ICASSP*, Istanbul, Turkey, 2000, vol. 2, pp. 1053–1056.

[13] F. Weninger, S. Watanabe, Y. Tachioka, and B. Schuller, "Deep recurrent de-noising auto-encoder and blind de-reverberation for reverberated speech recognition," in *Proc. of ICASSP*, Florence, Italy, 2014, to appear.

[14] A. Mohamed, G. Hinton, and G. Penn, "Understanding how deep belief networks perform acoustic modelling," in *Proc. of ICASSP*, Kyoto, Japan, 2012, pp. 4273–4276.

[15] M. Wöllmer, F. Weninger, J. Geiger, B. Schuller, and G. Rigoll, "Noise robust ASR in reverberated multisource environments applying convolutive NMF and Long Short-Term Memory," *Computer Speech and Language, Special Issue on Speech Separation and Recognition in Multisource Environments*, vol. 27, no. 3, pp. 780–797, 2013.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] F. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[18] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. of ICML*, Helsinki, Finland, 2008, pp. 1096–1103.

[19] T. Ishii, H. Komiyama, T. Shinozaki, Y. Horiuchi, and S. Kuroiwa, "Reverberant speech recognition based on denoising autoencoder," in *Proc. of INTERSPEECH*, Lyon, France, 2013, pp. 3512–3516.

[20] A.L. Maas, T.M. O'Neil, A.Y. Hannun, and A.Y. Ng, "Recurrent neural network feature enhancement: The 2nd CHiME challenge," in *Proc. of 2nd CHiME Workshop*, Vancouver, Canada, June 2013, pp. 79–80, IEEE.

[21] M.L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proc. of ICASSP*, Vancouver, Canada, 2013, pp. 7398–7402.

[22] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlícek, Y. Qian, P. Schwarz, et al., "The Kaldi speech recognition toolkit," in *Proc. of ASRU*, Big Island, HI, USA, 2011.

[23] M. Gales, "Semi-tied covariance matrices for hidden Markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 7, pp. 272–281, 1999.

[24] D. Povey and K. Yao, "A basis method for robust estimation of Constrained MLLR," in *Proc. of ICASSP*, Prague, Czech Republic, 2011, pp. 4460–4463.

[25] H. Xu, D. Povey, L. Mangu, and J. Zhu, "Minimum bayes risk decoding and system combination based on a recursion for edit distance," *Computer Speech & Language*, vol. 25, no. 4, pp. 802 – 828, 2011.

[26] T. Robinson, J. Fransen, D. Pye, J. Foote, and S. Renals, "WSJ-CAM0: A British English speech corpus for large vocabulary continuous speech recognition," in *Proc. of ICASSP*, Detroit, MI, USA, 1995, pp. 81–84.

[27] M. Lincoln, I. McCowan, J. Vepa, and H. Maganti, "The multi-channel Wall Street Journal audio visual corpus (MC-WSJ-AV): Specification and initial experiments," in *Proc. of ASRU*, San Juan, PR, USA, 2005, pp. 357–362.

[28] S.J. Young, G. Evermann, M.J.F. Gales, D. Kershaw, G. Moore, J.J. Odell, D.G. Ollason, D. Povey, V. Valtchev, and P.C. Woodland, *The HTK book version 3.4*, Cambridge University Engineering Department, Cambridge, UK, 2006.

[29] A. Graves, "Practical variational inference for neural networks," in *Advances in Neural Information Processing Systems*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds., vol. 24, pp. 2348–2356. MIT Press, 2011.