THE TUM SYSTEM FOR THE REVERB CHALLENGE: RECOGNITION OF REVERBERATED SPEECH USING MULTI-CHANNEL CORRELATION SHAPING DEREVERBERATION AND BLSTM RECURRENT NEURAL NETWORKS

Jürgen T. Geiger, Erik Marchi, Björn Schuller[1] *and Gerhard Rigoll*

Institute for Human-Machine Communication, Technische Universität München, Germany
[1] also with the Department of Computing, Imperial College London, UK

geiger@tum.de

ABSTRACT

This paper presents the TUM contribution to the 2014 REVERB Challenge: we describe a system for robust recognition of reverberated speech. In addition to an HMM-GMM recogniser, we use bidirectional long short-term memory (LSTM) recurrent neural networks. These networks can exploit long-range temporal context by using memory cells in the hidden units, which increases the robustness against reverberation. The LSTM is trained with phonemes as targets, and the predictions are converted into observation likelihoods and used as an acoustic model. Furthermore, we apply a dereverberation method called correlation shaping on the 8-channel recordings. This method applies a reduction of the long-term correlation energy in the received reverberant speech. The linear prediction residual, which generally contains information about reverberation, is processed to suppress the long-term correlation that is mostly due to the speaker-to-receiver impulse response. Using dereverberation as a front-end of the GMM in combination with the LSTM predictions leads to substantial improvements of the word error rate, achieving 11.19 % (relative improvement of about 35 %) and 28.13 % (improvement of about 30 %) with simulated and real data test sets, respectively. In the single-channel case, in which the dereverberation technique can not be applied, improvements of about 20 % (for simulated data) and 7 % (for real data) are obtained with the LSTM technique.

Index Terms— Dereverberation, BLSTM recurrent neural networks, multi-channel correlation shaping

1. INTRODUCTION

Reverberation severely degrades the performance of automatic speech recognition. The REVERB Challenge [1] addresses the problem of reverberated speech by providing a testbed for speech enhancement and speech recognition methods in a reverberant environment. Methods for robust speech recognition can be categorised into two groups: the first group involves methods of front-end enhancement, enhancing either the waveforms or extracted features by removing noise and reverberation [2]. It is possible to employ feature adaptations to transform the corrupt features, or to use noise-robust features directly. The other group of methods comprises improved recognition back-end systems. Here, one method is to adapt the models to noisy features, e. g., using multi-condition training or methods such as vector Taylor series. On the other hand, robust models are applied, where especially systems making use of deep Neural Networks (DNNs) were successful in the last years [3].

Suitable schemes for modelling reverberation are broadly applied such as the source-image method [4, 5]. Generally a reverberant scenario consists of a source speech signal which propagates through an acoustic channel and is then captured by a microphone. The microphone signal, however, contains a reverberated version of the source signal. Thus, dereverberation algorithms are applied on the microphone signal and output an estimate of the source signal. A plethora of dereverberation algorithms have been developed over the last two decades [6]. Several strategies have been proposed, ranging from linear prediction residual processing [7] to multiple microphone array-based techniques [8, 9]. Further approaches addressed blind system identification [10] by using subspace decomposition [11] and adaptive filters [12].

In our system we compare two multi-channel dereverberation techniques: the first technique, phase-error based filtering (PEF), relies on time-delay estimation with time-frequency masking [13, 14]. The second technique, namely correlation shaping (CS) [15], is based on linear prediction and reduces the length of the equalised speaker-to-receiver impulse response.

As a robust recognition back-end, our system employs bidirectional long short-term memory (LSTM) recurrent neural networks (RNNs) for phoneme prediction. One shortcoming of conventional RNNs is that the amount of context they use decays exponentially over time (the well-known *vanishing gradient problem* [16]). To overcome this problem, the LSTM concept has been introduced [17]. An LSTM-RNN exploits a self-learnt amount of temporal context, which makes it especially suited for a speech recognition task involving reverberation and additive noise. The application of LSTM networks in a double-stream system has first been introduced in [18] for conversational speech recognition, where LSTM phoneme predictions improved a simple triphone HMM system. In the first and second CHiME Speech Separation and Recognition Challenges [19, 20], the task was to recognise speech in a reverberant environment with highly non-stationary additive noise. Previous versions of the GMM-LSTM double-stream system that is also used in the present work showed a high performance in these recognition tasks [21, 22]. In this approach, an LSTM network is used to generate frame-wise phoneme predictions, largely improving the performance of the maximum likelihood (ML) trained HMM baseline system.

A short introduction to the REVERB Challenge is given in the next section, followed by a description of our recognition system. The experimental results are described in Section 4, before the paper

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 289021 (ASC-Inclusion). Thanks to Felix Weninger for providing the Kaldi recognition system.

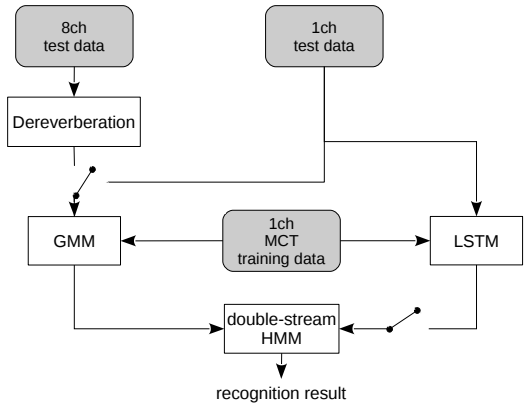

Fig. 1: System overview: a double-stream HMM system combining GMM and LSTM, and dereverberation using the 8-channel (ch) recordings

ends with some conclusions.

2. THE REVERB CHALLENGE

Let us just shortly review the REVERB Challenge. The goal of the 2014 REVERB Challenge [1] is to evaluate methods for speech enhancement and robust speech recognition in reverberant environments. Thus, there are two tasks in the challenge (enhancement and recognition). Our contribution is limited to the recognition track, where the task is to recognise read medium vocabulary (5 k) speech in different reverberant environments (reverberation times T60 ranging from 0.25 to 0.7 s). There are eight different environments, whereof six (called the SIM condition) are simulated by convolving the WSJCAM0 corpus [23] (which is a British English version of the WSJ corpus [24]) with measured room impulse responses. The impulse responses were measured in three different rooms, each at a near (50 cm) and far (200 cm) microphone distance. Additionally, stationary noise from the same rooms is added at an SNR of 20 dB. The other two conditions (called the REAL condition) correspond to recordings from the MC-WSJ-AV corpus [25]. This database contains real recordings of speakers standing in a reverberated room, measured at two distances (near =~ 100cm and far =~ 250cm). For all data (SIM and REAL), 8-channel recordings from a microphone array are available. In addition, it is also possible to evaluate one-channel systems. In this case, only the recording from the first microphone is taken. For training the recognition system, the WSJCAM0 training set containing 7 861 utterances from 92 speakers is provided. In addition, a multi-condition training set is available, which is created similarly like the SIM data, from the WSJCAM0 training set. Test experiments are performed using data from the eight different environments, where the six conditions from the SIM data together have 1 484 and 2 176 utterances in the development and test set, respectively, each from 20 speakers. The REAL data consist of 179 and 372 utterances (development and test) from five/ten speakers. Systems are evaluated using the word error rate (WER), counting the number of word substitutions, insertions and deletions as a fraction of the number of target words.

3. SYSTEM DESCRIPTION

Figure 1 shows an overview of the evaluated system. In addition to a standard HMM-GMM system, the HMM can make use of phoneme predictions from an LSTM network in a double-stream architecture. This LSTM network predicts phonemes and the predictions are converted to observation likelihoods for HMM decoding. Compared to the baseline HMM-GMM, we use a slightly improved system, which uses a different method for adaptation, and the main difference is that this system uses a trigram language model instead of the bigram.

The GMM is trained either with clean or multi-condition training data, while the LSTM uses multi-condition training data in all experiments. Furthermore, we apply a dereverberation method, processing the 8-channel recordings, and the GMM is either fed with the 1-channel reverberated test data or with the 8-channel processed test data. Here, we compare two different dereverberation techniques, namely phase-error based filtering (PEF) and correlation shaping (CS).

3.1. HMM-GMM recognition system

In addition to the REVERB baseline recognition system, which is implemented in HTK [26], we perform experiments with a (slightly improved) re-implementation with the Kaldi toolkit [27].

The baseline recogniser is a HMM-GMM system that employs tied-state HMMs with 10 Gaussian components per state and is trained according to the maximum-likelihood criterion. As features, standard MFCCs (computed every 10 ms from windows of 25 ms) including delta and delta-delta coefficients are used. Two methods are utilised to address the reverberation in the audio recordings. First, multi-condition training is employed by training the recogniser not only with clean training data, but also with the reverberated version of the training data. Second, constrained maximum-likelihood linear regression (MLLR) adaptation (in batch processing) is used to adapt the features to each test condition. The WSJ0 bi-gram language model (LM) is used during decoding.

A re-implementation using the Kaldi toolkit of this system is also used for our experiments. Instead of CMLLR, the Kaldi system employs basis feature space MLLR [28] for adaptation. This method performs well even on small amounts of adaptation data and thus is used for utterance-based batch processing instead of full batch processing. This means that the implementation is not capable of on-line processing since it always waits for the end of the current utterance. The biggest improvement that is made compared to the baseline system is the introduction of a trigram LM instead of the bigram LM that is used in the baseline.

3.2. LSTM Double-Stream Recogniser

In addition to GMM acoustic modelling, an LSTM network is used to generate frame-wise phoneme estimates, as first proposed in [18]. From these phoneme estimates, the observation likelihoods for the acoustic model are derived. These are used together with the GMM in a multi-stream architecture.

3.2.1. LSTM Recurrent Neural Networks

LSTM networks were introduced in [17]. Compared to a conventional RNN, the hidden units are replaced by so-called memory blocks. These memory blocks can store information in the cell variable $\boldsymbol{c}_t$. In this way, the network can exploit long-range temporal context. Each memory block consists of a memory cell and three gates: the input gate, output gate, and forget gate, as depicted in Fig. 2. These gates control the behaviour of the memory block. The forget gate can reset the cell variable which leads to 'forgetting' the

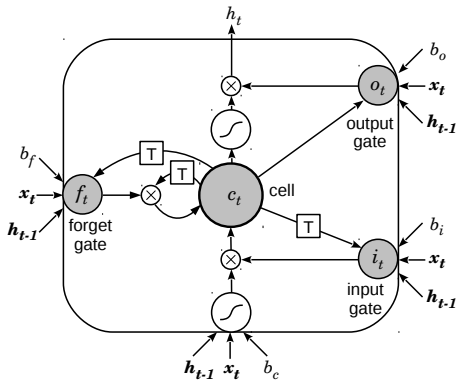

Fig. 2: Long Short-Term Memory block, containing a memory cell and the input, output and forget gates

stored input $\boldsymbol{c}_t$, while the input and output gates are responsible for reading input from $\boldsymbol{x}_t$ and writing output to $\boldsymbol{h}_t$, respectively:

$$\boldsymbol{c}_t = \boldsymbol{f}_t \otimes \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \otimes \tanh(\boldsymbol{W}_{xc}\boldsymbol{x}_t + \boldsymbol{W}_{hc}\boldsymbol{h}_{t-1} + \boldsymbol{b}_c) \quad (1)$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \otimes \tanh(\boldsymbol{c}_t) \quad (2)$$

where $\otimes$ denotes element-wise multiplication and $\tanh$ is also applied in an element-wise fashion. The variables $\boldsymbol{i}_t$, $\boldsymbol{o}_t$ and $\boldsymbol{f}_t$ are the output of the input gates, output gates and forget gates, respectively, $\boldsymbol{b}_c$ is a bias term, and $\boldsymbol{W}$ is the weight matrix. Each memory block can be regarded as a separate, independent unit. Therefore, the activation vectors $\boldsymbol{i}_t$, $\boldsymbol{o}_t$, $\boldsymbol{f}_t$ and $\boldsymbol{c}_t$ are all of same size as $\boldsymbol{h}_t$, i. e., the number of memory blocks in the hidden layer. Furthermore, the weight matrices from the cells to the gates are diagonal, which means that each gate is only dependent on the cell within the same memory block.

In addition to LSTM memory blocks, we use bidirectional RNNs [29]. A bidirectional RNN can access context from both temporal directions, which makes it suitable for speech recognition, where whole utterances are decoded. This is achieved by processing the input data in both directions with two separate hidden layers. Both hidden layers are then fed to the output layer. The combination of bidirectional RNNs and LSTM memory blocks leads to bidirectional LSTM networks [30], where context from both temporal directions is exploited. It has to be noted that using bidirectional LSTM networks makes it impossible to use the system for online processing.

A network composed of more than one hidden layer is referred to as a deep neural network (DNN) [3]. By stacking multiple (potentially pre-trained, but not in our system) hidden layers on top of each other, increasingly higher level representations of the input data are created (deep learning). When multiple hidden layers are employed, the output of the network is (in the case of a bidirectional RNN) computed as

$$\boldsymbol{y}_t = \boldsymbol{W}_{\overrightarrow{\mathbf{h}^N}y}\overrightarrow{\boldsymbol{h}_t^N} + \boldsymbol{W}_{\overleftarrow{\mathbf{h}^N}y}\overleftarrow{\boldsymbol{h}_t^N} + \boldsymbol{b}_y, \quad (3)$$

where $\overrightarrow{\boldsymbol{h}_t^N}$ and $\overleftarrow{\boldsymbol{h}_t^N}$ are the forward and backward activations of the N-th (last) hidden layer, respectively. Furthermore, a softmax activation function is used at the output, with

$$p(b^{(j)}|\boldsymbol{x}_t) = \frac{\exp(y_t^{(j)})}{\sum_{j'=1}^{P}\exp(y_t^{(j')})}, \quad (4)$$

to generate phoneme probabilities for all possible phonemes $j = 1, \ldots, P$. The LSTM is trained with on-line gradient descent using backpropagation through time, with cross entropy as error function. Our GPU enabled LSTM software is publicly available[1].

3.2.2. LSTM Phoneme Prediction

The LSTM is trained with phonemes as targets, as determined by a forced alignment with the HMM system. During decoding, discrete phoneme predictions are derived from the network output activations. These frame-wise phoneme predictions are used to obtain the likelihood $p(b_t|s_t)$ for the acoustic model in the following way: using a validation set, the frame-wise phoneme predictions are evaluated and all confusions are counted and stored in the phoneme confusion table $\boldsymbol{C}$ as row-normalised probabilities. The likelihood $p(\boldsymbol{x}_t|s_t)$ (observation given HMM state) is then obtained from this conditional probability table by using the mapping $b = \mathrm{m}(s)$ from HMM states to phonemes. Since the LSTM works with monophones, triphone structures are ignored here, by mapping triphone HMM states to the corresponding monophones. Thus, instead of directly predicting the probability $p(s_t|\boldsymbol{x}_t)$ with the network and using Bayes' theorem to obtain observation likelihoods, as in a typical hybrid system, the confusions of the network are 'learnt' in the conditional probability table $\boldsymbol{C}$ and used to derive the observation likelihoods $p(\boldsymbol{x}_t|s_t)$. With this method, the RNN needs fewer output nodes (as compared to predicting state posteriors), which makes it easier to train.

3.2.3. Double-Stream Decoding

In order to combine GMM acoustic modelling and LSTM phoneme predictions, we employ a double-stream HMM system. In every time frame t, the double-stream HMM has access to two independent information sources, $p_G(\boldsymbol{x}_t|s_t)$ and $p_L(\boldsymbol{x}_t|s_t)$, the acoustic likelihoods of the GMM and the LSTM predictions, respectively. The double-stream emission probability is then computed as

$$p(\boldsymbol{x}_t|s_t) = p_G(\boldsymbol{x}_t|s_t)^{\lambda} \cdot p_L(\boldsymbol{x}_t|s_t)^{2-\lambda}, \quad (5)$$

where the variable $\lambda \in [0, 2]$ denotes the stream weight of the GMM stream.

3.3. Dereverberation

We apply and compare two multi-channel dereverberation techniques: phase-error based filtering (PEF) [13, 14] and correlation shaping (CS) [15].

3.3.1. Phase-Error Based Filtering

PEF involves time-varying, or time-frequency (TF), phase-error filters based on estimated time-difference of arrival (TDOA) of the speech source and the phases of the signals acquired by the microphones. The phase variance [14] between two speech signals is defined as

$$\psi_\beta = \sum_{k=1}^{N}\sum_{\omega=-\omega_s}^{\omega_s}\theta_{\beta,k}^2(\omega), \quad (6)$$

where

$$\theta_{\beta,k}(\omega) = \angle X_{1,k}(\omega) - \angle X_{2,k}(\omega) - \omega\beta \quad (7)$$

indicates the level of noise and reverberation present in the entire speech signal. $\angle X_{1,k}$ and $\angle X_{2,k}$ are the phase spectra of the input

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

signals at frame k, and $\theta_{\beta,k}(\omega)$ is the minimised phase-error when β equals the TDOA, N indicates the number of segments in the speech signal, and ω_s is the highest frequency of interest. The phase-error measures the time misalignment at each frequency bin. The overall phase-error can be reduced to:

$$\theta_{\beta,k}(\omega) = \angle X_{1,k}(\omega) - \angle X_{2,k}(\omega) \tag{8}$$

with the assumption that the input signals are time-aligned. The phase error is used as a reward-punish criteria to removing noise from multi-microphone speech signals. Time-frequency blocks with large phase-error are scaled down in amplitude, whereas, blocks with low phase-error are preserved. First, the phase-error is computed from the two phase spectra. Then, a masking function is applied as a weighting function for the amplitude spectrum of each channel. Spectra are later summed up similarly to delay-and-sum. The parametrised scaling function,

$$\eta(\omega) = \frac{1}{1 + \gamma\theta^2_{\beta,k}(\omega)} \tag{9}$$

is proposed in [14] as a masking function to attenuate the time-frequency blocks, where γ is a fixed value. Higher values of γ reduce high phase-error blocks prominently with a consequent improved performance in low SNR scenarios and worse performance in high SNR situations. Phase-error based filtering is transferred to multi-microphone signals by applying the parametrised scaling function on all possible pairs of microphones. Each microphone pair i and j is processed by the following masking function

$$\eta_{ij}(\omega) = \frac{1}{1 + \gamma\theta^2_{ij}(\omega)} \tag{10}$$

which is extended from Equation (9). A detailed analysis [13] proposed the use of a modified geometric mean of the time-varying functions as follows:

$$\Phi_i(\omega) = \Big(\prod_{j=1, j\neg i}^{M} \eta_{ij}(\omega) \Big)^{\frac{1}{m}}, \tag{11}$$

where M is the number of microphones and m is a value which, for a standard geometric mean, would be equal to M. In this case it represents a factor affecting the aggressiveness of the algorithm. Using this approach, the estimation of high phase-error values is relevant in the mask averaging process, in fact, provided that a pair of microphones results in a very high phase-error for a certain time-frequency block, the resulting scaling value will be close to zero. The zero value is then kept in the geometrical averaging with the masking values for other pairs of microphones. The enhanced spectrum $\hat{S}(\omega)$ is obtained by summing up the enhanced spectra processed by the multi-channel mask $\phi_i(\omega)$, as defined in Equation (12).

$$\hat{S}(\omega) = \sum_{i=1}^{M} \Phi_i(\omega) X_i(\omega). \tag{12}$$

3.3.2. Correlation Shaping

CS reduces the long-term correlation in the linear prediction (LP) residual of reverberant speech. This approach improves both the audible quality and ASR accuracy of reverberant speech [15]. CS modifies the correlation structure of the processed speech signal y. Assuming that an array of M microphones records a speech source, the signal observed by the mth microphone x_m is processed by an adaptive linear filter $\boldsymbol{g}_m$ in order to minimise the weighted mean square error (MSE) between the actual output autocorrelation sequence $\boldsymbol{R}_{yy}$, and the desired output autocorrelation sequence $\boldsymbol{R}_{dd}$. The adaptive linear filters are continuously adjusted via a set of feedback functions in order to minimise the MSE.

Gradient descent is used to perform the minimisation via the adaptive filters. The gradient relies on the output autocorrelation $\boldsymbol{R}_{yy}$, the cross-correlation between the output and input, $\boldsymbol{R}_{yx_m}$, and the desired output autocorrelation $\boldsymbol{R}_{dd}$.

The autocorrelation sequence $R_{x_m x_m}(\tau)$ of the multi-channel input sequence $x_m(n)$ is given by

$$R_{x_m x_m}(\tau) = \sum_{n=0}^{N-1} x_m(n) x_m(n-\tau). \tag{13}$$

CS is implemented as a multi-input single-output linear filter, defined as

$$y(n) = \sum_{m=0}^{M-1} \boldsymbol{g}_m^T(n) \boldsymbol{x}_m(n). \tag{14}$$

The autocorrelation sequence $R_{yy}(\tau)$ of the output signal $y(n)$ is expressed as follows:

$$R_{yy}(\tau) = \sum_{n=0}^{N-1} y(n) y(n-\tau), \tag{15}$$

where N is the number of samples over which autocorrelation is computed, τ is the correlation lag.

The scope of CS is to minimize the weighted MSE given by

$$e(\tau) = W(\tau)\Big(R_{yy}(\tau) - R_{dd}(\tau)\Big)^2, \tag{16}$$

where $W(\tau)$ is a real value weight. The larger $W(\tau)$ is, the more relevant the error at a specific lag τ is.

For dereverberation purposes, the linear prediction residual is fed into the correlation shaping processor, and the target output correlation is set to be $R_{dd}(\tau) = \delta(\tau)$. By further exploiting autocorrelation symmetry, the gradient can be simplified as

$$\nabla_m(l) = \sum_{\tau>0} W(\tau) R_{yy}(\tau)\Big(R_{yx_m}(l-\tau) + R_{yx_m}(l+\tau)\Big). \tag{17}$$

This gradient is used in the following filter update equation

$$g_m(l, n+1) = g_m(l, n) - \mu \nabla'_m(l), \tag{18}$$

where μ is the learning rate parameter and $\nabla'_m(l)$ is given by

$$\nabla'_m(l) = \frac{\nabla_m(l)}{\sqrt{\sum_m \sum_l \nabla^2_m(l)}} \tag{19}$$

The dereverberated speech signal is obtained by applying the equaliser $g(l, n)$ onto the input signal. Considering that the reverberation time affects significantly audio quality and automatic speech recognition accuracy [15], a 'don't care' region is introduced. The 'don't care' region is applied to autocorrelation lags closed to the zeroth lag in order to improve the suppression of long-term components. This region modifies the gradient in Equation (17) and controls the value of the first autocorrelation lag.

4. EXPERIMENTS

We first describe the configuration of the parts of our recognition system before presenting and discussing the experimental results. In order to give detailed analysis of the contribution of different system components to the final results, we will provide extensive results using the development set and the test set.

4.1. System Configuration

4.1.1. HMM

The Kaldi HMM system is tested in similar configurations as the Challenge baseline system. First, a clean triphone recogniser is trained with the WSJCAM0 training set. Then, the reverberated training set is used to train a multi-condition acoustic model. For this model, the bases for MLLR adaptation are estimated, and finally, the trigram LM is used for decoding with this model. In the case of using front-end dereverberation, the employed method is always only applied on the test data, while the original acoustic model is used.

4.1.2. LSTM

Instead of MFCCs, the LSTM uses Mel filterbank features, complemented by their delta coefficients. This follows other recent studies that use NNs for speech recognition [3, 31]. We use 26 log filterbank coefficients (plus root-mean-square energy) covering the frequency range from 20–8 000 Hz, computed with a frame size of 25 ms and frame shift of 10 ms. Thus, in total, the dimension of features for the LSTM is 54. Features for the LSTM are extracted from the one-channel recordings. As an additional preprocessing step, we consider a per-utterance peak normalisation of the waveforms of the audio recordings. To this end, the recording is amplified to set the largest occurring absolute value to -3 dB of the maximum amplitude. This was necessary because the recordings from the REAL dataset are badly adjusted.

The topology of the tested bidirectional LSTM network is as follows: as the dimension of the feature vector is 54, this is also the size of the input layer. Three hidden layers are employed, where we tested two systems, with 100 or 200 LSTM blocks. The number of output units corresponds to the number of phonemes, which is 45 in our system. For training the networks, the multi-condition training set is employed. The networks are trained through online gradient descent with a learning rate of 10^{-5} and momentum of 0.9. During training, zero mean Gaussian noise with standard deviation 0.6 is added to the inputs in order to further improve generalisation. All weights were randomly initialised from a Gaussian distribution with mean 0 and standard deviation 0.1. After every training epoch, the average cross-entropy error per sequence on a validation set is evaluated. Training is aborted as soon as no improvement on the validation set can be observed during 10 epochs. This validation is a held-out part of the multi-condition training set, consisting of the utterances from 10 speakers. The stream weight for double-stream decoding is set to $\lambda = 1.2$.

4.1.3. Dereverberation

First, we evaluated PEF by using a frame size of 1 024 samples as in [14]. Smaller frame sizes result in less reliable phase estimates causing artifacts and distortions in the reconstructed signal. A frame shift of 10 ms was applied. γ was set to 0.01 in order to avoid an aggressive masking that is suitable only in low SNR conditions. In fact, the more γ steps up, the more WER increases rapidly. k was set to M in order to obtain the geometric mean of the signal and avoid severe speech distortions. Next, we performed CS by estimating autocorrelation functions on the whole speech segment. We applied 62.5 ms long equalisers, a 18.7 ms long 'don't care' region and exponential weighting. Correlation shaping was performed up to τ_{max} equals 62.5 ms.

Table 1: *Baseline recogniser vs. improved Kaldi system (WER on the development set). For decoding, either a brigram (bg) or trigram (tg) language model (LM) is used.*

Recogniser			WER [%]	
Adapt	MCT	LM	SIM	REAL
Baseline system				
-	-	bg	51.86	88.51
✓	-	bg	39.57	83.82
-	✓	bg	28.94	52.29
✓	✓	bg	25.16	47.23
Kaldi system				
-	-	bg	50.61	88.50
-	✓	bg	27.85	53.00
✓	✓	bg	22.07	45.52
✓	✓	tg	16.85	38.33

Table 2: *Baseline recogniser: influence of CS dereverberation, WER (in %) on the development set*

Baseline GMM					+CS	
Adapt	MCT	LM	SIM	REAL	SIM	REAL
-	-	bg	51.86	88.51	34.66	70.04
✓	-	bg	39.57	83.82	23.90	56.61
-	✓	bg	28.94	52.29	21.79	42.40
✓	✓	bg	25.16	47.23	19.48	37.66

4.2. Results for the Improved HMM-GMM

First, we replaced the baseline recognition system by a slightly improved version (from now on called the Kaldi system) as described in Section 3.1. A comparison of the performance of these two systems can be seen in Table 1. We used the Kaldi system in similar configurations as the baseline system, concerning multi-condition training and adaptation. The unadapted systems (clean and MCT) achieve similar results, while the adaptation implemented in the Kaldi system is slightly better. Furthermore, using the trigram LM leads to a large improvement in WER.

4.3. Influence of Dereverberation

Next, we investigate the influence of our dereverberation methods. This is firstly tested in combination with the baseline recognition system, in order to make it comparable to other systems that keep the back-end fixed and only improve the front-end of the system. The results of the experiments employing CS for dereverberation together with the baseline recogniser (using the development set) can be seen in Table 2. Similar improvements are obtained with all

Table 4: *Dereverberation: multi-channel correlation shaping (CS) and phase-error based filtering (PEF), development set*

Kaldi baselines			CS		PEF	
Adapt	MCT	LM	SIM	REAL	SIM	REAL
-	-	bg	32.33	66.81	31.57	68.22
-	✓	bg	20.85	37.42	20.34	38.45
✓	✓	bg	16.44	33.67	16.37	33.69
✓	✓	tg	12.05	27.70	12.11	28.43

configurations of the recogniser.

The results of using CS as a front-end to the Kaldi recogniser can be seen in Table 3. Generally, the same trends are visible as in combination with the baseline recognition system. For the best configuration, the results are improved by 28 % relatively for both the SIM and REAL datasets.

We compare the two employed dereverberation methods as a front-end to the Kaldi recognition system. The experimental results (using the Kaldi recogniser) are listed in Table 4. CS achieves slightly better results than PEF. For the REAL data, this is clearly illustrated in the results, while for the SIM data, this is at least the case for the best recognition back-end (last row in Table 4). Therefore, in all other experiments, we use this dereverberation method. This can be explained considering the difference between the two approaches: PEF aims to reduce noise and reverberation by minimising the mean phase variance while CS was exclusively designed for dereverberation and it is known that can effectively improve audible quality and ASR accuracy [15]. Furthermore, CS was implemented by estimating autocorrelation functions on the whole speech segment.

4.4. LSTM

Experimental results for combining the Kaldi recognition system (with or without front-end dereverberation) in the double-stream setup with the LSTM predictions are also listed in Table 3. Note that in all cases, the LSTM parameters are estimated using the multi-condition training set. For the SIM condition, including LSTM predictions leads to a similar improvement as with dereverberation. Apart from that, the improvements with the REAL data are smaller. Here, the mismatch between training and test data has a larger influence on the LSTM recognition performance. Table 3 also includes the results for using dereverberation and LSTM predictions in combination. Adding LSTM predictions to the GMM system with dereverberation leads to a further 15 % relative improvement (down to 10.21 %) for SIM, while the best system is not improved for the REAL data.

We tested different configurations of the LSTM recognition system and evaluated the frame-wise phoneme classification performance on the development set. The results are listed in Table 5. A smaller and a larger LSTM network were considered, and we investigated the influence of the audio normalisation that is described in Section 4.1.2. First of all, the results show that the normalisation had a positive effect on the results for the REAL data, while the SIM results are unaffected. Increasing the number of LSTM units in the hidden layers to 200 brought a small improvement to the SIM data. Since the LSTM was validated with a small partition of the original MCT training set (using a forced alignment of the development data for system training is not allowed in the challenge), which is comparable to the SIM data, it was decided to use the larger network in the other experiments. The large phoneme error rate with the REAL data is also reflected in the WER, where only a small improvement is obtained by using the LSTM predictions (cf. Table 3). This discrepancy between SIM and REAL data may indicate overtraining of the LSTM.

Table 5: *LSTM size: phoneme classification error (in %) on development*

			Phoneme Error	
Norm.	Network	weights	SIM	REAL
-	3x100	170 k	25.91	67.79
✓	3x100	170 k	25.55	51.39
✓	3x200	600 k	24.97	52.35

Table 6: *Baseline recogniser: Influence of CS dereverberation, test set*

Baseline GMM					+CS	
Adapt	MCT	LM	SIM	REAL	SIM	REAL
-	-	bg	51.68	88.53	34.56	72.88
✓	-	bg	39.16	81.53	24.29	59.41
-	✓	bg	29.51	56.94	24.12	45.65
✓	✓	bg	25.25	48.85	20.62	38.83

4.5. Test Set Results

Finally, experimental results with the test set are listed in Table 6 for the baseline recognition system (with and without speech dereverberation) and in Table 7 for the Kaldi system. Overall, the results are comparable to the development set results, and the same tendencies are visible.

To give a detailed coverage of the results on the test set, Table 8 includes test set results for five different system configurations, broken down into the eight different recording conditions. By looking at these results, it can be observed that, while the relative improvement from the LSTM predictions is similar for all (simulated) room conditions, the employed dereverberation technique works better with higher reverberation times. This is due to the fact that CS is penalising long-term reverberation energy more effectively. Thus, we can observe a better dereverberation under long impulse responses.

Row five in Table 8 represents our best system working with 1-channel recordings, while row six corresponds to the best 8-channel system. These two results were our official submissions in the two different conditions.

5. CONCLUSIONS

This paper presented the TUM system for the 2014 REVERB Challenge for recognition of reverberated speech. We use an LSTM network for phoneme prediction in addition to the GMM acoustic model, which increases the robustness of the system. In addition, a dereverberation method called correlation shaping is applied, using 8-channel audio recordings to estimate and filter the reverberation. Experiments were performed according to the official REVERB Challenge guidelines with the provided datasets. In addition

Table 3: *Kaldi recogniser: influence of CS dereverberation and LSTM, development set*

Kaldi GMM					+CS		+LSTM		+CS, +LSTM	
Adapt	MCT	LM	SIM	REAL	SIM	REAL	SIM	REAL	SIM	REAL
-	-	bg	50.61	88.50	32.33	66.81	38.46	79.90	22.85	59.16
-	✓	bg	27.85	53.00	20.85	37.42	21.07	47.20	16.59	37.16
✓	✓	bg	22.07	45.52	16.44	33.67	17.38	42.70	13.90	33.65
✓	✓	tg	16.85	38.33	12.05	27.70	12.98	36.14	10.21	28.16

Table 7: *Kaldi recogniser: influence of CS dereverberation and LSTM, test set*

Kaldi GMM					+CS		+LSTM		+CS, +LSTM	
Adapt	MCT	LM	SIM	REAL	SIM	REAL	SIM	REAL	SIM	REAL
-	-	bg	49.95	88.50	32.88	70.98	36.80	79.81	23.92	62.06
-	✓	bg	27.53	53.78	22.06	40.37	20.79	48.97	17.63	37.89
✓	✓	bg	22.36	46.14	17.75	34.06	17.77	43.83	14.82	34.39
✓	✓	tg	17.26	39.76	13.20	28.15	13.75	36.78	11.19	28.13

to the baseline recogniser, a slightly improved HMM-GMM was also tested. The results showed that all employed methods are highly effective for the recognition of reverberated speech. The correlation shaping approach led to slightly better performances than phase-error based filtering. This corroborates common wisdom that reducing the length of the equalised speaker-to-receiver impulse response can improve audible quality and ASR accuracy.

Regarding multi-channel results, the correlation shaping method gives significant improvements with a reduction of more than 25 % in WER. This is achieved at a low computational complexity, as the LSTM does not really improve these results (at least not on real data). On simulated data, the LSTM gives an additional improvement of about 15 % but this is achieved at a tremendous computational expense (LSTM training and decoding). In the single-channel case (i. e. without applying correlation shaping), the WER reduction is around 7 % on real data with the LSTM technique. It is about 20 % on simulated data.

Further improvements are possible with a full integration of all system components. In the current version, speech dereverberation is not applied on the multi-condition training set, which might bring another small improvement. In addition, the input to the LSTM network is also unenhanced. However, it is not yet confirmed in the literature, whether speech enhancement is still relevant for deep neural network based systems; this has to be shown in future work, especially also for LSTM systems. A detailed comparison of LSTMs (used as an acoustic model in a hybrid system) and similar DNNs without LSTM cells is also to be done in the future. Beyond that, the employed HMM-GMM does not yet use all state-of-the-art techniques; discriminative training will further improve the system.

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. IEEE WASPAA*, New Paltz, NY, USA, 2013.

[2] T. Virtanen, R. Singh, and B. Raj, *Techniques for noise robustness in automatic speech recognition*, Wiley, 2012.

[3] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[4] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *J. Acoust. Soc. Am.*, vol. 65, pp. 943, 1979.

[5] P. M. Peterson, "Simulating the response of multiple microphones to a single acoustic source in a reverberant room," *J. Acoust. Soc. Am.*, vol. 80, pp. 1527, 1986.

[6] P. A. Naylor and N. D. Gaubitch, "Speech dereverberation," in *Proc. IEEE IWAENC*, Eindhoven, The Netherlands, 2005.

[7] B. Yegnanarayana and P. S. Murthy, "Enhancement of reverberant speech using LP residual signal," *IEEE Audio, Speech, Language Process.*, vol. 8, no. 3, pp. 267–281, 2000.

[8] S. Griebel and M. Brandstein, "Wavelet transform extrema clustering for multi-channel speech dereverberation," in *Proc. IEEE IWAENC 99*, Pocono Manor, USA, 1999, pp. 27–30.

[9] D. Ward and M. Brandstein, *Microphone Arrays: Signal Processing Techniques and Applications*, Springer, Berlin, Germany, 2001.

[10] G. Xu, H. Liu, L. Tong, and T. Kailath, "A least-squares approach to blind channel identification," *IEEE Signal Process.*, vol. 43, no. 12, pp. 2982–2993, 1995.

Table 8: Test set results (WER in %) for selected systems for all eight test conditions. The best result for each condition is marked bold. The baseline or the Kaldi recognisers are improved by using Correlation Shaping (CS) dereverberation and/or LSTM predictions. The last two rows represent our official challenge submissions for 1-channel and 8-channel audio processing, respectively.

	SIMDATA							REALDATA		
	Room 1		Room 2		Room 3		Avg	Room 1		Avg
System	near	far	near	far	near	far		near	far	
Baseline	16.23	18.71	20.50	32.47	24.76	38.88	25.25	50.14	47.57	48.85
Baseline + CS	16.03	17.66	17.37	24.10	19.38	29.26	20.62	38.87	38.79	38.83
Kaldi	10.23	12.26	12.96	23.34	15.25	29.53	17.26	40.56	38.96	39.76
Kaldi + CS	10.86	11.50	10.74	15.62	11.40	19.09	13.20	**28.04**	28.26	28.15
Kaldi + LSTM	**8.32**	9.98	10.63	18.66	12.21	22.67	13.75	36.38	37.17	36.78
Kaldi + CS + LSTM	8.50	**9.66**	**9.40**	**13.70**	**9.64**	**16.26**	**11.19**	28.27	**27.99**	**28.13**

[11] S. Gannot and M. Moonen, "Subspace methods for multimicrophone speech dereverberation," *EURASIP Journal on Applied Signal Processing*, vol. 2003, pp. 1074–1090, 2003.

[12] Y. Huang, J. Benesty, and J. Chen, "A blind channel identification-based two-stage approach to separation and dereverberation of speech signals in a reverberant environment," *IEEE Speech Audio Process.*, vol. 13, no. 5, pp. 882–895, 2005.

[13] C. Y.-K. Lai and P. Aarabi, "Multiple-microphone time-varying filters for robust speech recognition," in *Proc. ICASSP*, Montreal, Canada, 2004, pp. 230–233.

[14] P. Aarabi and G. Shi, "Phase-based dual-microphone robust speech enhancement," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 34, no. 4, pp. 1763–1773, 2004.

[15] B. W. Gillespie and A. Atlas, "Strategies for improving audible quality and speech recognition accuracy of reverberant speech," in *Proc. ICASSP*, Hong Kong, 2003, pp. 673–676.

[16] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," in *Field Guide to Dynamical Recurrent Networks*, S. C. Kremer and J. F. Kolen, Eds. IEEE Press, 2001.

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

[18] M. Wöllmer, F. Eyben, B. Schuller, and G. Rigoll, "A multi-stream ASR framework for BLSTM modeling of conversational speech," in *Proc. ICASSP*, Prague, Czech Republic, 2011, pp. 4860–4863.

[19] J. P. Barker, E. Vincent, N. Ma, H. Christensen, and P. D. Green, "The PASCAL CHiME speech separation and recognition challenge," *Computer Speech and Language*, vol. 27, no. 3, pp. 621–633, 2013.

[20] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, "The second 'chime' speech separation and recognition challenge: Datasets, tasks and baselines," in *Proc. ICASSP*, Vancouver, Canada, 2013, pp. 126–130.

[21] 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, pp. 780–797, 2013.

[22] 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 2nd CHiME Challenge: Multi-Stream ASR Exploiting BLSTM Networks and Sparse NMF," in *Proc. CHiME Workshop*, Vancouver, Canada, 2013, pp. 25–30.

[23] 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. ICASSP*, Detroit, MI, USA, 1995, pp. 81–84.

[24] D. B. Paul and J. M. Baker, "The design for the Wall Street Journal-based CSR corpus," in *Proc. of the Workshop on Speech and Natural Language (HLT-91)*, 1992, pp. 357–362.

[25] 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. ASRU*, San Juan, PR, USA, 2005, pp. 357–362.

[26] 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.

[27] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlícek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *Proc. ASRU*, Honolulu, HI, USA, 2011.

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

[29] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Signal Process.*, vol. 45, no. 11, pp. 2673–2681, 1997.

[30] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[31] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. ICASSP*, 2013, pp. 6645–6649.