Memory-Enhanced Neural Networks and NMF for Robust ASR

Jürgen T. Geiger, Felix Weninger, *Member, IEEE*, Jort F. Gemmeke, *Member, IEEE*, Martin Wöllmer, Björn Schuller, *Member, IEEE*, and Gerhard Rigoll, *Senior Member, IEEE*

***Abstract*—In this article we address the problem of distant speech recognition for reverberant noisy environments. Speech enhancement methods, e. g., using non-negative matrix factorization (NMF), are succesful in improving the robustness of ASR systems. Furthermore, discriminative training and feature transformations are employed to increase the robustness of traditional systems using Gaussian mixture models (GMM). On the other hand, acoustic models based on deep neural networks (DNN) were recently shown to outperform GMMs. In this work, we combine a state-of-the art GMM system with a deep Long Short-Term Memory (LSTM) recurrent neural network in a double-stream architecture. Such networks use memory cells in the hidden units, enabling them to learn long-range temporal context, and thus increasing the robustness against noise and reverberation. The network is trained to predict frame-wise phoneme estimates, which are converted into observation likelihoods to be used as an acoustic model. It is of particular interest whether the LSTM system is capable of improving a robust state-of-the-art GMM system, which is confirmed in the experimental results. In addition, we investigate the efficiency of NMF for speech enhancement on the front-end side. Experiments are conducted on the medium-vocabulary task of the 2nd 'CHiME' Speech Separation and Recognition Challenge, which includes reverberation and highly variable noise. Experimental results show that the average word error rate of the challenge baseline is reduced by 64% relative. The best challenge entry, a noise-robust state-of-the-art recognition system, is outperformed by 25% relative.**

***Index Terms*—Long short-term memory, multi-stream recognition, noise robust speech recognition, non-negative matrix factorization.**

Manuscript received October 10, 2013; revised February 03, 2014; accepted April 07, 2014. Date of publication April 18, 2014; date of current version May 06, 2014. This work was supported in part by the Federal Republic of Germany through the German Research Foundation (DFG) under Grant SCHU 2508/4-1 and in part by the project AAL-2009-2-049 "Adaptable Ambient Living Assistant" (ALIAS) co-funded by the European Commission and the German Federal Ministry of Education (BMBF) in the Ambient Assisted Living (AAL) programme. The work of J. F. Gemmeke was supported by IWT-SBO project Aladin, grant 100049. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Shinji Watanabe.

J. Geiger, F. Weninger, and G. Rigoll are with the Institute for Human-Machine Communication, Technische Universität München, 80333 Munich, Germany (e-mail: geiger@tum.de; weninger@tum.de; schuller@tum.de; rigoll@tum.de).

B. Schuller is with the Institute for Human-Machine Communication, Technische Universität München, 80333 Munich, Germany, and also with the Department of Computing, Imperial College London, London SW7 2AZ, U.K.

M. Wöllmer is with the BMW Group, 80807 Munich, Germany.

J. F. Gemmeke is with the Department ESAT, KU Leuven, 3000 Leuven, Belgium (e-mail: jgemmeke@amadana.nl).

I. Introduction

AUTOMATIC speech recognition (ASR) in realistic acoustic conditions, e. g., involving room reverberation and interfering noise sources, is still a major research challenge. System robustness can be achieved by several strategies at different levels [1]. On the one hand, the speech signal can be enhanced by denoising [2], [3]. Monaural signal separation techniques such as non-negative matrix factorization (NMF) [4] are especially useful for cases where multi-channel audio with a specified microphone placement is not available. Furthermore, robust features such as RASTA-PLP [5], feature enhancement techniques [6], or feature transformations such as Linear Discriminant Analysis (LDA) [7] can improve the system robustness. On the other hand, robust models and decoding methods are often employed, including multi-condition training and/or discriminative training, e. g., using the Maximum Mutual Information (MMI) principle [8]. In addition, methods such as vector Taylor series (VTS) can be applied to adapt the acoustic model to noisy speech [9]. Such approaches addressing the robustness of the back-end of the recognition system were mostly developed for conventional systems using Gaussian mixture models (GMMs). Recently, deep neural networks (DNNs) gained popularity in speech recognition due to improved acoustic modeling performance compared to GMMs [10]. In [11], the potential of DNNs for robust ASR was demonstrated. In this work we consider a system using a Long Short-Term Memory (LSTM) recurrent neural network (RNN) as an acoustic model and NMF-based speech enhancement for robust ASR. We want to study the effects of combining the LSTM network with a state-of-the-art GMM system in a double-stream architecture. Furthermore, we investigate the influence of speech enhancement on the different acoustic models.

A. Related Work

Recently, RNNs have been applied in a tandem system for robust ASR [12]. Deep RNNs with end-to-end training are also capable of being used for speech recognition on their own, without an HMM framework [13]. One shortcoming of conventional RNNs is that the amount of context they use decays exponentially over time (the well-known *vanishing gradient problem* [14]). To overcome this problem, the LSTM concept has been introduced [15]. An LSTM-RNN exploits a self-learned amount of temporal context, which makes it especially suited for a speech recognition task involving reverberation and additive noise. Previously, we suggested using LSTM networks for noise-robust spelling recognition in a tandem HMM-LSTM system [16]. The application of

LSTM networks in a double-stream system was first introduced in [17] for conversational speech recognition, where LSTM phoneme predictions improved a simple triphone HMM system. Multi-stream HMM systems were initially proposed to combine independent feature streams [18]. For example, in this way, GMMs can be fused with NNs [19] or with NMF-based sparse coding techniques [20] for increased robustness.

Building upon the first CHiME Speech Separation and Recognition Challenge [21], in its second installment [22], a medium-vocabulary speech recognition track was introduced by using the Wall Street Journal (WSJ0) read speech corpus. Together with degradation introduced by room reverberation and highly non-stationary additive noise, this proved to be a challenging recognition scenario. In our successful contributions to the 1st and 2nd CHiME challenges, we used a GMM-LSTM multi-stream system in combination with NMF speech enhancement [23]–[25]. An LSTM network was used to generate frame-wise phoneme predictions, largely improving the performance of the maximum likelihood (ML) trained HMM baseline system. The HMM system employed NMF speech enhancement in its front-end. However, up to now, the LSTM approach has never been combined with discriminatively trained HMM systems. Since in previous work, it was always combined with a ML-trained HMM-GMM system, it is not clear whether the LSTM approach will also lead to such large improvements in combination with a state-of-the-art discriminatively trained GMM system.

In the study presented in [26], a speech enhancement method using spatial and spectral cues was capable of improving a noise-robust small-vocabulary recognition system that utilized DNNs. In our work, we consider only spectral features (without additional information such as spatial cues), to enable a fair comparison. On the other hand, in [11], a DNN ASR system could not be improved by applying feature enhancement in the front-end.

B. Contribution

We now combine the LSTM approach with a state-of-the-art discriminatively trained ASR system, additionally making use of an NMF-based speech enhancement approach. In particular, we want to address the following research questions: (I), is the LSTM system capable of improving a state-of-the-art noise-robust HMM-GMM ASR system? Our experimental results will affirm this question. Furthermore, (II), what is the influence of speech enhancement in combination with our back-end recognition system? The robustness of the LSTM network has already been demonstrated (e. g., in [16]) and therefore it is unclear whether the combination of a state-of-the-art HMM-GMM and an LSTM system can be further improved by applying speech enhancement in the front-end. Our results will show that, while the employed speech enhancement method improves the GMM system, this is not the case for the LSTM system.

C. Overview

A flow chart of the evaluated ASR system is depicted in Fig. 1. On the back-end side, a double-stream architecture is used for acoustic modeling. In addition to a GMM acoustic model, a deep bidirectional LSTM network generates frame-wise phoneme estimates, which are converted into observation likelihoods to be used as an acoustic model in the HMM framework. Both acoustic models are always trained in a multi-condition fashion, using noise-free and noisy data. On the front-end side, our system can optionally use NMF speech enhancement (independently for both streams), exploiting an exemplar-based approach where noisy speech is decomposed into additive combinations of speech and noise training segments.

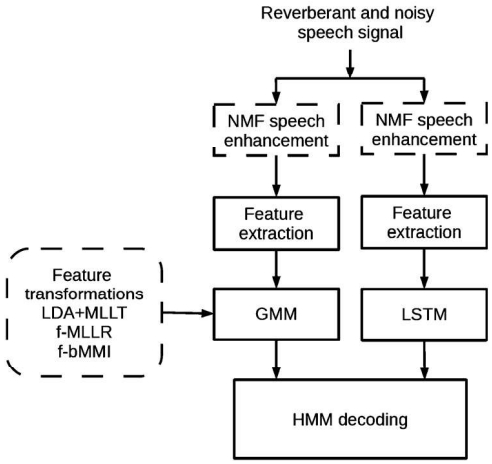

Fig. 1. Block diagram of the evaluated system: The central component is a multi-stream HMM fusing GMM and LSTM acoustic models. Speech enhancement, using NMF, is optionally applied prior to feature extraction. For the GMM stream, feature transformations (as explained in later sections) can be employed.

The described system is evaluated on the original medium-vocabulary task of the 2nd CHiME Speech Separation and Recognition Challenge [22]. We will demonstrate the influence of different system components on the recognition performance and show that our system strongly outperforms the challenge baseline as well as the best-performing challenge entry.

The employed methods (HMM-GMM, NMF speech enhancement and LSTM) are described in Sections II, III, and IV, respectively. Details about the experimental setup and parametrization of algorithms are given in Section V. The results of our experiments are presented and discussed in Section VI, before concluding in Section VII.

II. HMM-GMM-Based Speech Recognition

We use a state-of-the-art HMM-GMM ASR system, as it was described by Tachioka *et al.* in [27]. This system is implemented with the Kaldi speech recognition toolkit [28]. In addition to ML training, it uses discriminative learning (DL) and various feature transformation (FT) methods. Discriminative training is performed using boosted Maximum Mutual Information (bMMI) as proposed in [8]. The MMI principle aims at maximizing the posterior probabilities of the correct utterances, given the trained models. By applying bMMI, a weight is introduced, strengthening the influence of hypotheses with a higher error. For bMMI, the objective function is

$$\mathcal{F}_{bMMI}(\lambda) = \sum_{r=1}^{R} log \frac{p_\lambda(\mathcal{X}_r|\mathcal{M}_{s_r})^\kappa p_L(s_r)}{\sum_s p_\lambda(\mathcal{X}_r|\mathcal{M}_s)^\kappa p_L(s) e^{-bA(s,s_r)}}, \quad (1)$$

where $r = 1, \ldots, R$ are the training utterances, and $\mathcal{X}_r$ are the corresponding feature sequences. $\mathcal{M}_s$ is the HMM sequence of sentence s, s_r is the reference transcription of utterance r, κ

is the acoustic scale, p_λ is the likelihood of the acoustic model with the parameters λ, and p_L is the language model likelihood. The last term in the denominator is the boosting weight, where $b > 0$ is the boosting factor and $A(s, s_r)$ is the phone accuracy of sentence s given the reference s_r. In addition to model-space bMMI, we apply feature-space bMMI as well. The introduction of the boosting factor incorporates the concept of a soft margin that is proportional to the errors in a hypothesized sentence.

Furthermore, techniques for feature transformation are employed. Feature transformations can improve the class separation and address the speaker variability in the training data. Channel variability, such as different channels and additive noise or reverberation, can also be compensated by feature transformations. Linear discriminant analysis (LDA) is applied on 'stacked' MFCC vectors extracted from multiple signal frames (centered around the current frame) and reduces these high-dimensional features to a smaller dimension. The necessary class labels are obtained by aligning the triphone HMM states. There are too few data to train full-covariance models, because of the high-dimensional acoustic feature space. Therefore, diagonal-covariance models, which do not consider correlations between features, are used instead. We use a maximum likelihood linear transform (MLLT), as described in [29], for decreasing the correlations between features. The combination of LDA and MLLT exploits context to reduce the influence of non-stationary noise, and correlations between feature dimensions that were introduced by noise are removed. To address the problem of large variations among speakers, speaker adaptive training (SAT) is applied: During the ML training procedure, feature-space maximum likelihood linear regression (f-MLLR), which is the same as constrained MLLR [30], is applied to estimate a speaker-dependent transform. The estimated transform is subsequently used during model re-estimation. First, a tight-beam decoding is performed to re-estimate the SAT transform (the speaker identities are known), before doing a final decoding pass.

III. NMF Speech Enhancement

The speech enhancement component of our system uses exemplar-based spectrogram factorization algorithms previously employed in noise robust ASR experiments on the Aurora-2, SPEECON and CHiME/GRID datasets [25], [31]. In short, noisy Mel-magnitude spectra are decomposed as a sparse, non-negative linear combination of speech and noise dictionary atoms. The activations of the speech atoms are then used to obtain an estimate of the clean speech. In order to capture time context, atoms span multiple time frames and utterances are decoded using a sliding-window method:

$$\mathbf{V} \approx \hat{\mathbf{V}} = \Lambda^{(s)} + \Lambda^{(n)} = \sum_{j=1}^{J} \mathbf{W}_j^{(s)} h_j + \sum_{k=1}^{K} \mathbf{W}_k^{(n)} h_k, \quad (2)$$

where $\mathbf{V}$ is a $B \times T$ dimensional spectrogram representing the current window of the observed noisy speech, B is the number of spectral bands, and T the number of consecutive frames in a windowed spectrogram. The spectrograms $\mathbf{\Lambda}^{(s)}$ and $\mathbf{\Lambda}^{(n)}$ are estimates for its speech and noise content, respectively, $\mathbf{W}$ are $B \times T$ dimensional dictionary atoms, and h their *activation weights*. We denote the number of speech atoms by J and similarly the noise dictionary size by K.

The coefficients h_j and h_k are obtained through supervised NMF by minimizing the KL-divergence between $\mathbf{V}$ and $\hat{\mathbf{V}}$ regularized with a sparsity constraint on the activations [25]. After factorization, we estimate clean and noise estimates of the noisy speech spectra by overlap-adding the sliding windows. With these, we estimate the Wiener filter used to do speech enhancement [25]. The choice for speech enhancement rather than feature enhancement allows us more freedom in the feature extraction of the double-stream recognizer.

The dictionary atoms are formed by *exemplars*, spectrograms directly extracted from spectrograms [32]. Preliminary experiments on the CHiME development set revealed that the use of exemplars yields better results compared to the learnt representations of speech and noise used in previous and related work [33], [34].

IV. LSTM Acoustic Modeling

As an alternative to GMM acoustic modeling, an LSTM network is used to generate frame-wise phoneme estimates, as first proposed in [17]. The observation likelihoods are derived from these phoneme estimates.

A. LSTM RNN

LSTM networks were introduced in [15]. 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 behavior of the memory block. The activation vector of each gate is computed as, for example for the input gate,

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_{xi}\boldsymbol{x}_t + \boldsymbol{W}_{hi}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{ci}\boldsymbol{c}_{t-1} + \boldsymbol{b}_i) \quad (3)$$

where $\boldsymbol{W}$ is a weight matrix, $\boldsymbol{x}_t$ is the input vector at time step t, $\boldsymbol{h}_{t-1}$ is the hidden state vector of the previous time step, $\boldsymbol{b}_i$ denotes the input bias vector, and σ is a sigmoid function, causing each gate either to be open or closed. The forget gate can reset the cell variable which leads to 'forgetting' the 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:

$$\boldsymbol{c}_t = \boldsymbol{f}_t \otimes \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \otimes \tanh(\boldsymbol{W}_{xc}\boldsymbol{x}_t + \boldsymbol{W}_{hc}\boldsymbol{h}_{t-1} + \boldsymbol{b}_c) \quad (4)$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \otimes \tanh(\boldsymbol{c}_t) \quad (5)$$

where $\otimes$ denotes element-wise multiplication and $\tanh$ is also applied in an element-wise fashion. 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 [35]. A bidirectional RNN can access context from both

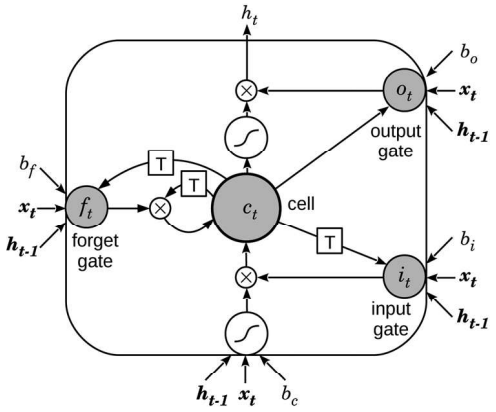

Fig. 2. Long Short-Term Memory block, containing a memory cell and the input, output, and forget gates.

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 [36], where context from both temporal directions is exploited. An NN composed of more than one hidden layer is referred to as a deep NN [10]. By stacking multiple (potentially pre-trained) 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

$$\boldsymbol{y}_t = \boldsymbol{W}_{\overrightarrow{\mathbf{h}^N}\, y} \overrightarrow{\boldsymbol{h}_t^N} + \boldsymbol{W}_{\overleftarrow{\mathbf{h}^N}\, y} \overleftarrow{\boldsymbol{h}_t^N} + \boldsymbol{b}_y, \tag{6}$$

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,

$$p(b^{(j)}|\boldsymbol{x}_t) = \frac{\exp\left(y_t^{(j)}\right)}{\sum_{j'=1}^{P} \exp\left(y_t^{(j')}\right)}, \tag{7}$$

to generate probabilities for the targets, in our case for all possible phonemes $j = 1, \ldots, P$.

For network training, we use on-line gradient descent by backpropagation through time, where weight changes are applied after processing one utterance in each training epoch. Training utterances are 'shuffled' (presented in random order) to improve generalization in on-line learning. The cross entropy is employed as an error function for training. Our LSTM software is publicly available[1].

B. LSTM Phoneme Prediction

For a phoneme prediction LSTM, the input vectors $\boldsymbol{x}_t$ of the network correspond to the employed acoustic features, whereas the output $\boldsymbol{y}_t$ represents frame-wise activations for each phoneme. In order to use phonemes as training targets, a forced alignment of the baseline HMM recognizer with clean data is obtained.

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

During decoding, a phoneme prediction is derived from the network output activations,

$$\hat{b}_t = \arg\min_j \left(p(b^{(j)}|\boldsymbol{x}_t)\right), j = 1, \ldots, P \tag{8}$$

leading to one phoneme prediction per frame. The process of LSTM decoding and generating the phoneme prediction is summarized in the function

$$\mathcal{L}(\boldsymbol{x}_t) = \hat{b}_t. \tag{9}$$

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 the development set, the frame-wise phoneme predictions are evaluated and all confusions are counted and stored in the phoneme confusion table $\boldsymbol{C}$ as row-normalized probabilities:

$$C(i,j) = p\left(\hat{b} = j | b = i\right). \tag{10}$$

Although the phoneme confusions are estimated on the development set, the performance generalizes well to the test set. The likelihood $p(\boldsymbol{x}_t|s_t)$ (observation given HMM state) is then obtained by using the mapping $b = \mathrm{m}(s)$ from HMM states to phonemes. Since the LSTM works with monophones, triphone structures are ignored here, mapping triphone HMM states to the corresponding monophones. The acoustic likelihoods are therefore computed as

$$p_L(\boldsymbol{x}_t|s_t) = C(\mathrm{m}(s_t), \mathcal{L}(\boldsymbol{x}_t)). \tag{11}$$

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 network converts the output scores $p(b_t|\boldsymbol{x}_t)$ to discrete phoneme predictions $\hat{b}_t$ using Eqn. (8). These phoneme predictions are evaluated on the development set. By storing the confusions in C and normalizing the rows of C, this matrix constitutes a discrete probability table for $p(\hat{b}_t|b_t)$. For HMM decoding, the likelihoods $p(\boldsymbol{x}_t|s_t)$ are required, which are now approximated by $p(\mathcal{L}(\boldsymbol{x}_t)|\mathrm{m}(s_t)) = p(\hat{b}_t|b_t)$, exploiting the surjective mapping from states to phonemes. Thereby, the confusions of the network are 'learned' in the conditional probability table C and used to derive the observation likelihoods $p(\boldsymbol{x}_t|s_t)$. These likelihoods are now expected to have a high discriminative power. With this method, the RNN needs fewer output nodes (as compared to predicting state posteriors), which makes it easier to train.

Phoneme classification experiments in [37] support our choice of using bidirectional LSTM RNNs instead of other network architectures. In that work, bidirectional LSTMs were shown to perform better than feedforward networks or traditional RNNs without LSTM cells. To underpin this statement, in our experimental section, we will additionally show results where a feedforward network is employed for phoneme prediction instead of an LSTM.

C. Double-Stream Decoding

In order to combine GMM acoustic modeling 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 computed as

$$p(\boldsymbol{x}_t|s_t) = p_G(\boldsymbol{x}_t|s_t)^{\lambda} \cdot p_L(\boldsymbol{x}_t|s_t)^{2-\lambda}, \tag{12}$$

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

V. Experimental Setup

A. Evaluation Database

Experiments are conducted on the medium-vocabulary task of the 2nd CHiME Challenge [22]. This database consists of utterances from the WSJ0 5 k vocabulary read speech corpus, convolved with real binaural impulse responses measured in a domestic environment, and mixed with realistic noise backgrounds recorded in the same environment. The impulse responses were measured for a fixed position 2 m in front of a head and torso simulator. The background noise contains a rich collection of sound sources from a lounge and kitchen such as electronic and kitchen appliances, noise produced by the inhabitants (such as footsteps, laughter or background speech), and noise from outside. Speech utterances are temporally placed in the background noise such that different signal-to-noise ratios (SNRs), from -6 to 9 dB, in steps of 3 dB, are obtained. The training set contains 7 138 utterances from 83 speakers summing up to 14.5 hours (forming the WSJ0 SI-84 training set), in clean, reverberated and reverberated+noisy form. For the development set, 409 noisy utterances from 10 other speakers are provided at all six different SNRs, leading to a total number of 2 454 utterances (4.5 h in total). The test set includes 330 noisy utterances from 12 speakers, at all SNRs (1 980 recordings or 4 h in total). All noisy utterances are also provided in an embedded form, where 10 s of surrounding background noise are included. Word error rate (WER) is used as an evaluation measure. For each evaluated system, we report the average WER across all SNRs, and for some systems also the WER for each of the six different SNR values.

B. Preprocessing and Feature Extraction

While the challenge data are stereophonic, we consider only single-channel signals. These signals are obtained by averaging over both channels. For the employed database, this corresponds to a delay-and-sum beam-forming, since the target speaker is located at a fixed position in front of the microphones (azimuth 90 degrees).

All features are extracted from frames of 25 ms and a frame shift of 10 ms. The baseline HMM-GMM system uses standard MFCCs, i. e., 13 coefficients with their delta and acceleration coefficients, whereas for the advanced HMM-GMM the features are processed using feature frame stacking and LDA projection (as described in the following section). The LSTM network uses logarithmic Mel filter bank (Log-FB) features (instead of MFCCs) that are also complemented by their delta and acceleration coefficients. The choice of features follows other recent studies that use NNs for speech recognition [10], [13], [38]. We use 26 Log-FB (plus root-mean-square energy) covering the frequency range from 20–8 000 Hz, computed with the same frame size and shift as applied for the MFCCs.

C. Parameterization

1) HMM-GMM Recognizer: Parameterization and training of HMM-GMM acoustic models in our system is the same as described in [27] and works as follows: 40 phonemes (including silence) are integrated in context-dependent triphone models with 2 500 states and a total number of 15 000 Gaussians. First, models are trained with clean training data applying the ML principle. Next, ML training is continued with reverberated training data, using the alignments and triphone tree structures from the clean models. Then, isolated noisy training data are used for training. In the experimental section, this basic system (using only ML training) is referred to as the ML GMM acoustic model. From this setup, an advanced system is created using discriminative training and feature transformations. First, another set of ML training iterations is performed after applying the described feature transformations, using the noisy training data. Here, the 13 static MFCC coefficients of nine consecutive frames are concatenated together and LDA is applied to reduce the resulting 117 dimensional vector to 40 dimensions. The LDA uses the 2 500 aligned triphone HMM states as classes. Subsequently, features are transformed using MLLT and model re-estimation is done. Afterwards, an f-MLLR transform is estimated for SAT, leading to another set of model re-estimation iterations. Based on the resulting acoustic models, discriminative training is performed with the noisy training data, using model-space and feature-space bMMI with a boosting factor of $b = 0.1$. During decoding, the language model weight is tuned for each system to minimize the average word error rate across all SNRs on the development set.

2) NMF Speech Enhancement: All factorization operates on Mel-magnitude spectra, with $B = 40$ bands. The window length is $T = 20$ frames, and a window shift of one frame is used.

From the reverberated isolated utterances in the training data, 10 000 speech exemplars were extracted by random sampling. Two noise dictionaries were used: a fixed noise dictionary of 4 000 exemplars randomly extracted from the embedded utterances in the noisy training set, and a noise dictionary extracted from the 10 seconds of embedding noise in the noisy utterance that is being decoded. This second noise dictionary consists of all exemplars that can be extracted from the 1 000 frames of noise: $2 \cdot 500 - T + 1 = 981$ exemplars. This brings the total number of exemplars in the dictionary to 14 981. An additional experiment is performed in order to demonstrate the effect of exploiting the embedding noise. For that experiment a fixed noise dictionary of 4 981 exemplars is used, without exploiting the knowledge of surrounding noise.

The sparsity for the speech was set at 0.075 times the average L_1 norm of the fixed part of the dictionary (speech and noise jointly). The noise sparsity was set at 0.5 times the speech sparsity. The number of iterations was kept constant at 400. These values were tuned using a small random subset of the AURORA-4 corpus.

3) LSTM Configuration: LSTM network parameters are estimated with multi-condition training, using the combination

of the reverberated noisy-free and noisy training sets. The inputs to the LSTM network are globally mean and variance normalized. To this end, the global means and variances are computed from the reverberated noise-free and noisy training set features. In addition to the input and output layers, the employed bidirectional LSTM network is made of three hidden layers (making it a deep NN), where 81, 128, and 90 hidden units are employed. These values correspond to the number of memory blocks in each of the two temporal directions. The number of input nodes corresponds to the length of the feature vector (81 in case of Log-FB), while the number of output nodes is equal to the number of phonemes, which is 40 in our case. LSTM topologies were chosen according to previously performed experiments on similar databases. The networks are trained through gradient descent with a learning rate of 10^{-5} and a 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 generalization. All weights were randomly initialized from a Gaussian distribution with mean 0 and standard deviation 0.1. The average cross entropy error per sequence on the development set is evaluated after every fifth epoch in the training phase. Using an early stopping strategy, training is aborted as soon as no improvement on the development set can be observed during 25 epochs.

VI. Experiments and Results

A. GMM vs. LSTM

First, we want to study the effects of combining the employed LSTM method with the two different GMM acoustic models: the standard system using only ML training or the advanced discriminatively trained system employing LDA, MLLT and SAT. Experimental results for the resulting four system combinations are displayed in Table I. Our ML-trained GMM acoustic model (first row) and the discriminatively trained system including all feature transformations (second row) correspond to the systems described by Tachioka *et al.* in [27], except that we apply beam-forming (cf. Section V-B), which brings an absolute improvement of about 7% in average WER. Combining GMM and LSTM acoustic modeling in the double-stream system (GMM stream weight $\lambda = 1.0$) leads to further large improvements in WER. The ML-trained HMM is improved by almost 30% relatively. More impressively, the discriminatively trained HMM can also vastly be improved (18% relative) by adding the LSTM predictions. The relative improvements are nearly the same for all SNRs. For comparison, results for a standard DNN, taken from [39], are listed in Table I. The DNN acoustic model had 3 hidden layers and 500 k parameters and is thus comparable to the LSTM employed in our study. Because it is not speaker-adapted (though it still uses the LDA+MLLT feature transformation), the DNN is not able to beat the GMM. Beyond that, the performance is also weaker than the GMM-LSTM double-stream system. A phoneme prediction DNN (employed in the same way as the LSTM, and described in more detail later in this section) performs significantly worse than the LSTM.

The stream weight λ in Eqn. (12) controls the trade-off between the influence of the GMM and LSTM acoustic model likelihoods. When setting $\lambda = 2$, the HMM uses only the GMM acoustic model (though with an exponent of 2). Accordingly, $\lambda = 0$ means that the system uses only the information from the LSTM stream. Fig. 3 shows the average WER for different stream weights. Of particular interest are the results with $\lambda = 0.0$ and $\lambda = 2.0$. The GMM alone (36.0%) performs better than the LSTM (43.4%), This might appear contrastive to the conclusion that DNN acoustic models perform better than GMMs [10]. However, only the advanced GMM (including DL+FT) beats the LSTM, while the LSTM approach outperforms the standard ML-trained GMM. In addition, the employed LSTM only models monophones. Further improvements are expected when modeling context-dependent HMM states. In this case, the model complexity needs to be increased and the resulting higher number of trainable parameters might require unsupervised pre-training of the network. The best performance of the double-stream system is achieved with a stream weight of $\lambda = 1.0$ (27.3%). In all other reported experiments, we therefore use a stream weight of $\lambda = 1.0$. These results show that, even if the LSTM acoustic model alone performs worse than the GMM, the GMM system can greatly benefit from the combination with the LSTM predictions in the double-stream setup.

In order to demonstrate the merits of the chosen LSTM RNN architecture, we trained different feedforward DNNs for phoneme recognition. Table II shows the framewise phoneme error rate on the development set for these experiments. A layer size of 400 hidden units was chosen for the DNNs, with either 3 or 4 hidden layers. Feature frame stacking (incorporating 7 neighboring frames) was applied to exploit temporal context. The phoneme recognition results show that the DNN cannot reach the performance of the LSTM network. What can also be seen is that adding the fourth layer to the DNN (and thereby adjusting the number of parameters to the LSTM) brings no improvement. In that case, the missing pre-training or initialization of the DNN becomes noticeable. Compared to the

TABLE I
WER (in %) on the Development Set When Combining Different GMM Acoustic Models With the LSTM

Acoustic Model		SNR [dB]						Mean
GMM	LSTM	-6	-3	0	3	6	9	
ML	-	68.5	59.0	50.3	44.3	39.7	34.5	49.4
DL+FT	-	52.9	43.0	34.6	26.7	23.5	19.0	33.3
ML	✓	53.7	43.2	35.2	29.8	26.7	22.1	35.1
DL+FT	✓	**45.2**	**34.4**	**27.5**	**21.5**	**19.2**	**15.7**	**27.3**
DNN [39]		57.2	45.9	36.2	30.6	26.4	23.3	36.6
DL+FT	DNN	50.9	40.6	32.6	25.8	22.6	18.7	31.9

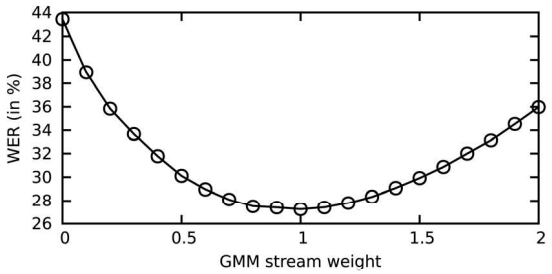

Fig. 3. Average WER (development set) for different HMM decoding stream weights λ. Values of 0.0 and 2.0 correspond to using only the LSTM or GMM acoustic model, respectively.

TABLE II
FRAMEWISE PHONEME ERROR RATE (PER) ON THE DEVELOPMENT SET, COMPARING AN LSTM WITH DIFFERENT DNNS WITH OR WITHOUT FEATURE FRAME STACKING

Network	Layers	# Weights	PER [%]
DNN	3x400	370k	59.5
DNN (feature stacking)	3x400	490k	51.8
DNN (feature stacking)	4x400	650k	52.1
LSTM	81-128-90	660k	35.8

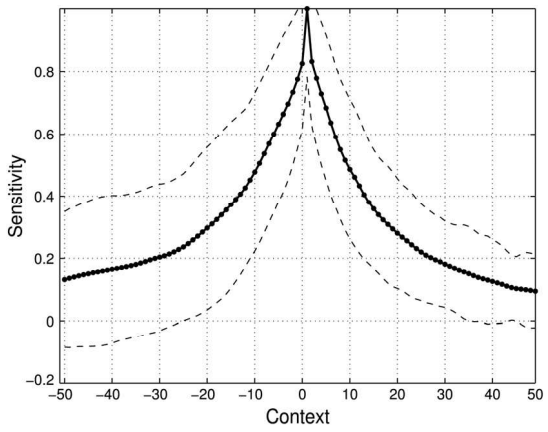

Fig. 4. Sensitivity (mean ± standard deviation) of network outputs to input nodes of neighboring time frames.

DNN, the LSTM is better structured and thus easier to train. The DNN with 3 layers and feature frame stacking was also used to obtain the results in the last row in Table I.

In [40], a methodology was proposed to analyze the amount of context that is exploited by an LSTM network. From the sequential Jacobian [37],

$$J_{ji}^{tt'} = \frac{\partial y_t^{(j)}}{\partial x_t'^{(i)}} \tag{13}$$

which corresponds to the derivative of the network outputs $y_t^{(j)}$ with respect to network inputs $x_{t'^{(i)}}$ at different time steps (given as a relative position compared to time step t), the sensitivity is computed by summing up the absolute magnitudes of the derivatives over all input units i and output units j and all time steps t and normalizing them:

$$S_t' = \frac{\sum_t \sum_j \sum_i |J_{ji}^{tt'}|}{\max \sum_t \sum_j \sum_i |J_{ji}^{tt'}|} \tag{14}$$

This sensitivity can be considered as a measure of the contribution of input nodes to the activity at the output of the network. Fig. 4 shows the sensitivity (mean ± standard deviation over time steps t) of a randomly chosen sequence (with SNR of -6 dB) in the development set. In particular, the plot shows the average sensitivity of the outputs with respect to the inputs from ± 50 frames of context. For example, considering a sensitivity threshold of 0.2, the network exploits roughly 30 frames (300 ms) of past and future information. The standard deviation (dashed lines) shows that there is a higher variability in using past context. In comparison to standard DNNs, which usually exploit context of around 10 frames via feature frame stacking [10], the employed LSTM architecture has access to a much larger amount of context information. For a standard DNN, the amount of context could be increased by using larger windows for feature frame stacking. However, this would increase the number of trainable parameters of the network. An advantage of the LSTM topology is that the amount of exploitable context is independent of the number of parameters.

TABLE III
INFLUENCE OF NMF ENHANCEMENT (ENHANCING TRAINING AND/OR TESTING DATA) ON THE TWO DIFFERENT GMM SYSTEMS (AVG. WER ON THE DEVELOPMENT SET). TWO NMF CONFIGURATIONS ARE TESTED, WHERE ONE OF THEM USES A CONTEXT NOISE DICTIONARY AND THE OTHER DOES NOT

	Enhancement		Context	
Acoustic Model	train	test	yes	no
GMM (ML)	-	-	49.4	
GMM (ML)	-	✓	50.3	51.5
GMM (ML)	✓	✓	42.4	44.3
GMM (DL+FT)	-	-	33.3	
GMM (DL+FT)	-	✓	**29.3**	30.8
GMM (DL+FT)	✓	✓	30.8	32.5

B. *Influence of Speech Enhancement*

Now we study the influence of NMF speech enhancement as a preprocessing step to GMM and/or LSTM training and/or decoding. Enhancing the training and test data can be regarded as feature-space noise-adaptive training. It can also be seen as a way of minimizing the mismatch between training and test data: enhancing only test data leads to a mismatch that may degrade recognition performance. Applying the speech enhancement only to the test data corresponds more to a 'plug-and-play' mode, where we regard the back-end of the recognition system as a constant and just enhance the input to the system. Furthermore, this simplifies the training procedure, since the signal enhancement approach is not required to be performed on the training set, which is generally much larger than the test set. In this setup, the system is not adapted to the artifacts introduced by the speech enhancement. A comparison of enhancing only the test data or also the training material (using only the GMM acoustic model) is shown in Table III. Rows 1 and 4 can also be found in Table I (as rows 1 and 2, respectively). We first discuss the results of the NMF configuration that exploits knowledge of context noise. In case of the ML model, when only the test data are enhanced, the system performance undergoes a slight degradation; the artifacts introduced by the NMF cancel out any improvements due to the enhancement. When creating matched conditions between the training and test data (through enhancing also the training set), NMF is able to decrease the WER of the ML model to 42.4%. While processing the training data increases the computational cost of model training, the process is trivially parallelizable and can easily be accelerated using modern GPU hardware [41] and advanced optimization methods [42]. If the GMM acoustic model is discriminatively trained and feature transformations (LDA, MLLT, SAT) are applied, the WER is slightly better when enhancing only the test data. In this case, the mismatch between training and test data (that is introduced by the signal enhancement) is compensated by the SAT transform; a speaker-dependent feature transform is estimated with f-MLLR in batch mode for the (enhanced) test data. Presumably, this feature transform not only adapts to the target speaker, but also on the enhancement. Not using the

TABLE IV
INFLUENCE OF APPLYING NMF ENHANCEMENT ON THE TRAINING AND/OR TESTING DATA FOR THE LSTM, IN COMBINATION WITH THE TWO DIFFERENT GMM SYSTEMS (WHICH USE ENHANCED (ENH.) DATA IN ALL CASES), SHOWING WER ON THE DEVELOPMENT SET

Acoustic Model	LSTM Enhancement train	LSTM Enhancement test	Mean WER
GMM (ML, enh.) + LSTM	-	-	32.2
GMM (ML, enh.) + LSTM	-	✓	35.6
GMM (ML, enh.) + LSTM	✓	✓	33.6
GMM (DL+FT, enh.) + LSTM	-	-	**25.1**
GMM (DL+FT, enh.) + LSTM	-	✓	27.4
GMM (DL+FT, enh.) + LSTM	✓	✓	25.9

TABLE V
TEST SET EVALUATION (WER IN %) OF OUR ASR SYSTEMS WITH NMF ENHANCEMENT (ENH.), WITH CONTEXT OR WITHOUT (NC), AND LSTM PHONEME PREDICTIONS AND COMPARISON TO RELATED APPROACHES

System	SNR [dB] -6	-3	0	3	6	9	Mean
Other systems for CHiME 2013 track 2 task							
Baseline noisy GMM [22]	70.4	63.1	58.4	51.1	45.3	41.7	55.0
NMF, noisy GMM [33]	61.9	55.6	50.9	43.5	39.1	37.4	48.1
NMF, GMM (ML)+LSTM [24]	57.4	49.0	42.5	37.4	32.6	29.7	41.4
GMM (DL+FT) [27]	54.7	45.1	36.0	28.6	24.4	21.4	35.0
Blind source extraction [43]	42.2	38.4	32.7	29.2	26.9	23.7	32.2
Bin. mask., GMM (DL+FT) [39]	44.1	35.5	28.1	21.2	17.4	14.8	26.9
DNN [44]	42.1	31.7	24.7	19.4	16.4	14.3	24.8
RDNN [44]	38.1	29.1	23.0	17.9	15.0	13.6	22.8
Examined Systems							
GMM (DL+FT)	46.4	36.2	28.5	21.6	17.9	15.7	27.7
GMM (DL+FT, enh.)	40.0	30.8	24.5	18.8	15.7	14.1	24.0
GMM (DL+FT) + LSTM	37.1	27.2	22.5	16.7	13.9	**11.8**	21.5
GMM (DL+FT, nc-enh.)+LSTM	35.2	26.3	20.7	16.2	13.4	12.0	20.6
GMM (DL+FT, enh.) + LSTM	**33.8**	**25.7**	**20.3**	**15.5**	**13.0**	11.9	**20.0**

context noise dictionary for NMF enhancement leads to a slight degradation in all tested configurations. Since the amount of required context is only small, we thus use the variant that exploits the knowledge of embedding noise in all other experiments with NMF enhancement.

Next we examine the influence of speech enhancement as a preprocessing step to the LSTM system (when used in conjunction with the GMM system). The experimental results of different setups are listed in Table IV. Here, the GMM system uses the best enhancement setup as determined in Table III (matched for the ML system and mismatched for the DL+FT system). Altogether, in this setup (GMM system sees enhanced data), no improvement can be observed by also using enhanced speech as input to the LSTM system. The best result (row 4) is achieved with the LSTM system (without NMF) in combination with the advanced GMM system (with enhancement in mismatched conditions). This is the overall best WER we obtained on the development set.

C. Test Set Results

Finally, Table V shows results on the CHiME Challenge test set. Generally, the results show the same tendencies as on the development set.

From our systems, we include the unenhanced GMM-only system (with DL and FT). Furthermore, we report results where NMF enhancement or the LSTM double-stream system, or both are applied. Here, following the results from Section VI-B, we apply enhancement in mismatched condition for the GMM, and the LSTM works with unenhanced data (row 4 in Table IV). The results for the official challenge baseline (multi-condition ML-trained HMM-GMM using MFCCs) are shown in the first row of the table (55.0% average WER). This was improved with NMF enhancement exploiting long-context speech and noise models by Hurmalainen *et al.* [33] by 13% relative. In our original contribution to the challenge [24], we used the same NMF enhancement approach as proposed in the present study, together with an earlier version of the LSTM multi-stream system, in combination with the official challenge baseline. This system reduced the WER to 41.4%. An alternative recognition system for the challenge was provided by Tachioka *et al.* in [27], which, compared to the official baseline, uses LDA, MLLT, SAT and discriminatively trained HMM-GMMs, resulting in a WER of 35.0%. This result is surpassed (8% relative) by the approach proposed by Nesta *et al.* [43]. Their system works mainly on the front-end side, exploiting blind source extraction, and using the challenge baseline recognizer. Including binary feature masking into the front-end of the system in [27] improved the result by 23% relative, which was the challenge entry with the best results [39]. In [44], a well-tuned DNN and a recurrent DNN were evaluated on the CHiME task. These systems outperform the best GMM baseline. Our systems are also based on the GMM system described in [27]. First, performing the simple beam-forming method as described in Section V-B leads to a relative improvement of 21%, down to an average WER of 27.7%. By adding NMF enhancement to this system, this result is improved by 13% relatively. The GMM-LSTM system brings a larger improvement, yielding a WER of 21.5%. Finally, when both NMF enhancement and the LSTM double-stream system are exploited, we achieve a WER of 20.6% without exploiting context in NMF, or 20.0% with context. Compared to the official challenge baseline, this is a relative improvement of 64%. The best challenge entry is beaten by 25% relative. Notably, our best system also surpasses the DNN and recurrent DNN results presented in [44].

VII. CONCLUSIONS

We have presented a system for noise-robust ASR that exploits exemplar-based speech enhancement and combines GMM acoustic modeling with phoneme predictions from a deep bidirectional LSTM RNN.

In particular, we were interested in the following questions: (I), when a state-of-the art discriminatively trained HMM-GMM system including feature transformations is used instead of the simple baseline, can the LSTM predictions still lead to an improvement? Our results (cf. Table I) revealed that the LSTM brings large improvements to both GMM systems. The other open question we wanted to address was, (II), whether speech enhancement (in our case NMF) can still improve a DNN-based recognition system. The results presented in Section VI-B show that the NMF enhancement approach was capable of improving the GMM system by 12% relative. What's more, also in the GMM-LSTM system, enhancing the GMM input improves the WER (27.3% vs. 25.1%). However, when the GMM sees enhanced features, additionally enhancing

the input to the LSTM brings no further improvement. Due to the improved memory of the LSTM network, the system is already very robust. These results are in contrast to the finding in [26], where a speech enhancement approach was able to improve a DNN ASR system. However, in those experiments, the enhancement approach had access to spatial information, while in our experiments, the enhancement and recognition systems work without this information. On the other hand, our experiments confirm the results presented in [11], where a feature enhancement method could not improve a DNN ASR system in a positive way.

Overall, the experimental results showed that the novel combination of a state-of-the-art GMM and an LSTM is highly efficient. The system achieved large improvements in WER and outperformed all entries to the 2nd CHiME Challenge (as well as comparable DNN systems) while being compliant with the challenge guidelines, leading to the best current result on this database. On the test set, the challenge baseline, a standard HMM system, had an average WER of 55.0%, whereas with our best system, a WER of 20.0% was obtained.

Future work will concentrate on finding out whether other speech enhancement approaches are able to improve the LSTM system. Furthermore, it will be interesting to investigate how the LSTM performs in the hybrid setup where it predicts HMM states instead of phonemes.

Acknowledgment

The authors would like to thank Alex Graves for helpful discussions on LSTM network training and the organizers of the CHiME Challenge for providing the data set and the HTK and Kaldi baseline ASR systems.

References

[1] T. Virtanen, R. Singh, and B. Raj, *Techniques for noise robustness in automatic speech recognition*. New York, NY, USA: Wiley, 2012.

[2] R. Martin, "Noise power spectral density estimation based on optimal smoothing and minimum statistics," *IEEE Trans. Speech Audio Process.*, vol. 9, no. 5, pp. 504–512, Jul. 2001.

[3] S. J. Rennie, J. R. Hershey, and P. A. Olsen, "Efficient model-based speech separation and denoising using non-negative subspace analysis," in *Proc. ICASSP*, Las Vegas, NV, USA, 2008, pp. 1833–1836.

[4] P. Smaragdis, "Convolutive speech bases and their application to supervised speech separation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 1, pp. 1–14, Jan. 2007.

[5] H. Hermansky, N. Morgan, A. Bayya, and P. Kohn, "RASTA-PLP speech analysis technique," in *Proc. ICASSP*, San Francisco, CA, USA, 1992, vol. 1, pp. 121–124.

[6] A. Krueger and R. Haeb-Umbach, "Model-based feature enhancement for reverberant speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 7, pp. 1692–1707, Sep. 2010.

[7] R. Haeb-Umbach and H. Ney, "Linear discriminant analysis for improved large vocabulary continuous speech recognition," in *Proc. ICASSP*, San Francisco, CA, USA, 1992, pp. 13–16.

[8] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for model and feature-space discriminative training," in *Proc. ICASSP*, Las Vegas, NV, USA, 2008, pp. 4057–4060.

[9] D. Y. Kim, C. Kwan Un, and N. S. Kim, "Speech recognition in noisy environments using first-order vector Taylor series," *Speech Commun.*, vol. 24, no. 1, pp. 39–49, 1998.

[10] 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 Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

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

[12] O. Vinyals, S. V. Ravuri, and D. Povey, "Revisiting recurrent neural networks for robust ASR," in *Proc. ICASSP*, Kyoto, Japan, 2012, pp. 4085–4088.

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

[14] 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. Piscataway, NJ, USA: IEEE Press, 2001.

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

[16] M. Wöllmer, F. Eyben, B. Schuller, Y. Sun, T. Moosmayr, and N. Nguyen-Thien, "Robust in-car spelling recognition-a tandem BLSTM-HMM approach," in *Proc. Interspeech*, Brighton, U.K., 2009, pp. 2507–2510.

[17] 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.

[18] J. A. Bilmes and C. Bartels, "Graphical model architectures for speech recognition," *IEEE Signal Process. Mag.*, vol. 22, no. 5, pp. 89–100, Sep. 2005.

[19] A. Hagen and A. Morris, "Recent advances in the multi-stream HMM/ANN hybrid approach to noise robust ASR," *Comput. Speech Lang.*, vol. 19, no. 1, pp. 3–30, 2005.

[20] Y. Sun, J. F. Gemmeke, B. Cranen, L. ten Bosch, and L. Boves, "Using a DBN to integrate Sparse Classification and GMM-based ASR," in *Proc. Interspeech*, Makuhari, Japan, 2010, pp. 2098–2101.

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

[22] 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, BC, Canada, 2013, pp. 126–130.

[23] 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," *Comput. Speech Lang., Special Issue Speech Separat. Recogn. Multisource Environ.*, vol. 27, pp. 780–797, 2013.

[24] 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, BC, Canada, 2013, pp. 25–30.

[25] J. Gemmeke, T. Virtanen, and A. Hurmalainen, "Exemplar-based sparse representations for noise robust automatic speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 7, pp. 2067–2080, Sep. 2011.

[26] M. Delcroix, Y. Kubo, T. Nakatani, and A. Nakamura, "Is speech enhancement pre-processing still relevant when using deep neural networks for acoustic modeling?," in *Proc. Interspeech*, Lyon, France, 2013, pp. 2992–2996.

[27] Y. Tachioka, S. Watanabe, and J. R. Hershey, "Effectiveness of discriminative training and feature transformation for reverberated and noisy speech," in *Proc. ICASSP*, Vancouver, BC, Canada, 2013, pp. 6935–6939.

[28] 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.

[29] G. Saon, M. Padmanabhan, R. Gopinath, and S. Chen, "Maximum likelihood discriminant feature spaces," in *Proc. ICASSP*, Istanbul, Turkey, 2000, pp. 1129–1132.

[30] M. J. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Comput. Speech Lang.*, vol. 12, no. 2, pp. 75–98, 1998.

[31] H. Kallasjoki, U. Remes, J. F. Gemmeke, T. Virtanen, and K. J. Palomäki, "Uncertainty measures for improving exemplar-based source separation," in *Proc. INTERSPEECH*, Florence, Italy, 2011, pp. 469–472.

[32] T. Sainath, B. Ramabhadran, D. Nahamoo, D. Kanevsky, D. Van Compernolle, K. Demuynck, J. Gemmeke, J. Bellegarda, and S. Sundaram, "Exemplar-based processing for speech recognition: An overview," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 98–113, Nov. 2012.

[33] A. Hurmalainen, J. F. Gemmeke, and T. Virtanen, "Compact long context spectral factorisation models for noise robust recognition of medium vocabulary speech," in *Proc. CHiME Workshop*, Vancouver, BC, Canada, 2013, pp. 13–18.
[34] F. Weninger, M. Wöllmer, and B. Schuller, "Combining Bottleneck-BLSTM and Semi-Supervised Sparse NMF for Recognition of Conversational Speech in Highly Instationary Noise," in *Proc. Interspeech*, Portland, OR, USA, 2012, pp. 302–305.
[35] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.
[36] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Netw.*, vol. 18, no. 5-6, pp. 602–610, 2005.
[37] A. Graves, "Supervised sequence labelling with recurrent neural networks," Ph.D. dissertation, Technische Univ. München, Munich, Germany, 2008.
[38] A. Mohamed, G. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 14–22, Jan. 2012.
[39] Y. Tachioka, S. Watanabe, J. Le Roux, and J. R. Hershey, "Discriminative methods for noise robust speech recognition: A CHiME challenge benchmark," in *Proc. CHiME Workshop*, Vancouver, BC, Canada, 2013, pp. 19–24.
[40] M. Wöllmer, A. Metallinou, N. Katsamanis, B. Schuller, and S. Narayanan, "Analyzing the memory of BLSTM neural networks for enhanced emotion classification in dyadic spoken interactions," in *Proc. ICASSP*, Kyoto, Japan, 2012, pp. 4157–4160.
[41] J. F. Gemmeke, A. Hurmalainen, T. Virtanen, and Y. Sun, "Toward a practical implementation of exemplar-based noise robust ASR," in *Proc. EUSIPCO*, Barcelona, Spain, 2011, pp. 1490–1494.
[42] T. Virtanen, J. Gemmeke, and B. Raj, "Active-set Newton algorithm for overcomplete non-negative representations of audio," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 11, pp. 2277–2289, Nov. 2013.
[43] F. Nesta, M. Matassoni, and R. F. Astudillo, "A flexible spatial blind source extraction framework for robust speech recognition in noisy environments," in *Proc. CHiME Workshop*, Vancouver, BC, Canada, 2013, pp. 33–38.
[44] C. Weng, D. Yu, S. Watanabe, and B.-H. Juang, "Recurrent deep neural networks for robust speech recognition," in *Proc. ICASSP*, Florence, Italy, 2014, to be published.

Jürgen T. Geiger received his diploma in electrical engineering and information technology in 2009 from TUM, where he is currently pursuing the doctoral degree at the Institute for Human-Machine Communication. His main research interest is in robust pattern recognition in audio processing. He led the winning contribution to the 2013 2nd 'CHiME' Speech Separation and Recognition Challenge (Track 1). Mr. Geiger serves as a reviewer for several high-profile international journals and conferences. He has published more than 25 peer-reviewed papers in journals and conference proceedings.

Felix Weninger received his diploma in computer science from TUM in 2009. He is currently pursuing his doctoral degree in the Machine Intelligence & Signal Processing (MISP) Group at TUM's Institute for Human-Machine Communication, focusing his research on new machine learning approaches for robust automatic speech recognition and speaker characterization. In 2013/14, he was an intern at Mitsubishi Electric Research Labs (MERL), Cambridge, MA. Mr. Weninger is a member of the IEEE and serves as a reviewer for several IEEE journals in the field. He has published more than 60 peer-reviewed papers in books, journals and conference proceedings.

Jort F. Gemmeke is a postdoctoral researcher at the KU Leuven, Belgium. He received the M.Sc degree in physics from the Universiteit van Amsterdam (UvA) in 2005. In 2011, he received the Ph.D. degree from the University of Nijmegen on the subject of noise robust ASR using missing data techniques. He is known for pioneering the field of exemplar-based noise robust ASR. His research interests are automatic speech recognition, source separation, noise robustness and acoustic modeling, in particular exemplar-based methods and methods using sparse representations.

Martin Wöllmer obtained his diploma in 2008 and his doctoral degree for his studies in affective computing, pattern recognition and speech processing in 2013, both in electrical engineering and information technology from Technische Universität München. He is currently working for the BMW Group in Munich, Germany.

Björn Schuller received his diploma in 1999, his doctoral degree for his study on Automatic Speech and Emotion Recognition in 2006, and his habilitation and private lectureship in the subject area of Signal Processing and Machine Intelligence for his work on Intelligent Audio Analysis in 2012, all in electrical engineering and information technology from TUM. He is a tenured faculty member heading the MISP Group at TUM's Institute for Human-Machine Communication since 2006 and a part-time Senior Lecturer in Machine Learning in the Department of Computing at the Imperial College London in London/UK since 2013. Dr. Schuller is president of the Association for the Advancement of Affective Computing (AAAC), elected member of the IEEE Speech and Language Processing Technical Committee, and member of the ACM, IEEE and ISCA and (co-)authored 5 books and more than 390 publications in peer reviewed books, journals, and conference proceedings.

Gerhard Rigoll obtained the Dipl.-Ing. degree in 1982 and the Dr.-Ing. degree in 1986 in the area of automatic speech recognition. He received the Dr.-Ing. habil. degree in 1991 from Stuttgart University with a thesis on speech synthesis. In 2002, he joined Technische Universität München (TUM), where he is now heading the Institute for Human-Machine Communication. His research interests are in the field of multi-modal human-machine communication, covering areas such as speech and handwriting recognition, face detection & identification, and object tracking. Dr. Rigoll is a Senior Member of the IEEE and is the author and co-author of more than 450 papers in the field of pattern recognition, covering all the previously mentioned application areas. He is Associate Editor of the EURASIP Journal on Audio, Speech and Music Processing, served as Associate Editor for the IEEE TRANSACTIONS ON AUDIO, SPEECH AND LANGUAGE PROCESSING from 2005-2008, and as member of the Overview Editorial Board of the IEEE Signal Processing Society.