Noise robust ASR in reverberated multisource environments applying convolutive NMF and Long Short-Term Memory[☆]

Martin Wöllmer[*], Felix Weninger, Jürgen Geiger, Björn Schuller, Gerhard Rigoll

Institute for Human-Machine Communication, Technische Universität München, Theresienstr. 90, 80333 München, Germany

1. Introduction

The design of robust automatic speech recognition (ASR) systems is an active area of research since it is commonly observed that ASR performance degrades in challenging acoustic conditions such as non-stationary noise and reverberation (Schuller et al., 2009). Techniques to cope with distortions can be applied both in feature extraction and speech decoding.

In addition, prior to feature extraction, speech enhancement techniques can be used to compensate the effect of noise. In the last decade, monaural source separation techniques by non-negative matrix factorization (NMF) have emerged as a promising solution that is portable across application scenarios and acoustic conditions (Helen and Virtanen, 2005; Smaragdis, 2007; Rennie et al., 2008; Raj et al., 2010; Evangelista et al., 2011). For instance, the 2006 CHiME Challenge (Cooke et al., 2010) featured an NMF-based approach for cross-talk separation that used speaker models (speech dictionaries) in a supervised NMF framework (Schmidt and Olsson, 2006). In this article, we use a

[☆] This paper has been recommended for acceptance by ‘Jon Barker’.

[*] Corresponding author. Tel.: +49 89 289 28550; fax: +49 89 289 28535.
E-mail address: woellmer@tum.de (M. Wöllmer).

convolutive extension of NMF that has delivered promising results for speech denoising (see Smaragdis, 2007), and use its capability to model spectral sequences corresponding to the words encountered in the recognition task.

In addition to speech enhancement techniques, a number of advanced feature extraction approaches have emerged as alternatives to conventional speech features such as Mel-Frequency Cepstral Coefficients (MFCC). A popular approach to enhance the front-end of recognition systems is the application of probabilistic features generated by a neural network that is trained on phoneme or phoneme state targets. Such Tandem systems unite the advantages of discriminative modeling via neural networks and generative frameworks such as Hidden Markov Models (HMM) (Hermansky et al., 2000). Recent studies show that ASR performance in challenging noisy conditions can be enhanced by utilizing neural network structures that are able to explicitly model long-range context for Tandem feature generation, leading to better results than simple feature frame stacking (Wöllmer et al., 2011). By Graves and Schmidhuber (2005), it has been shown that bidirectional neural networks that are based on the so-called Long Short-Term Memory (LSTM) technique (Hochreiter and Schmidhuber, 1997) enable better phoneme recognition rates than recurrent neural networks (RNNs) or multi-layer perceptrons (MLP).

These findings motivated research in ASR systems exploiting bidirectional LSTM (BLSTM). Initial studies concentrated on keyword spotting using BLSTM networks with a Connectionist Temporal Classification (CTC) output layer (Fernandez et al., 2007) or framewise BLSTM phoneme predictions (Wöllmer et al., 2010a,b). Recently, we introduced a multi-stream BLSTM-HMM decoder that can also be applied for continuous ASR (Wöllmer et al., 2011b).

An alternative way to generate framewise phoneme or word predictions that can be processed in an HMM-based back-end is non-negative sparse classification (NSC, see Gemmeke et al., 2011a). If the speech dictionaries are appropriately labeled—e.g., by correspondence to words, phonemes, or HMM states—the activations of their entries directly reveal content of the utterance if sparsity constraints are followed. This has been successfully exploited for *exemplar-based* techniques in speech decoding (Gemmeke et al., 2011a; Hurmalainen et al., 2011).

In this contribution, we present and compare various BLSTM- and NMF/NSC-based ASR architectures that are robust with respect to noise and reverberation. To this end, we attempt to enhance both front-end features and back-end decoding of the system by using long-range context, and exploit the source separation capabilities of NMF/NSC to complement the context modeling by BLSTM networks. In addition to Tandem BLSTM features, we evaluate CTC networks that can be used as an alternative to HMMs and can be trained on unsegmented speech data (Graves et al., 2006). Further, we show how our multi-stream BLSTM-HMM recognizer can be enhanced by employing speaker adapted BLSTM predictors. All systems are evaluated on the PASCAL CHiME corpus (Barker et al., submitted for publication) which was designed to allow researchers a comparison of their ASR systems in a noisy and reverberated multisource environment. Building on our contribution to the 2011 PASCAL CHiME Challenge (Weninger et al., 2011a), we investigate alternative BLSTM-based speech recognition architectures and improve our previous results by fully speaker adapted BLSTM networks and non-negative sparse classification.

In Section 2 we present the CHiME corpus and the challenge recognition task. Section 3 outlines our convolutive NMF approach before the principle of BLSTM and CTC is explained in Section 4. In Section 5, we briefly review the principle of non-negative sparse classification. Section 6 provides an overview over the evaluated ASR systems and experiments and results are presented in Sections 7 and 8, respectively.

2. The CHiME corpus

Our approaches for speech enhancement and ASR systems were evaluated on the official corpus provided for the 2011 PASCAL CHiME Challenge (Barker et al., submitted for publication). The challenge task is to recognize voice commands of the form *command–color–preposition–letter–digit–adverb*, e.g., “*set white by U seven again*”, spoken in a noisy living room. The vocabulary size is 51. For best comparability with the challenge results, we evaluate by the official challenge competition measure, which is keyword accuracy, i.e., the recognition rate of letters (25 spoken English letters excluding ‘W’) and digits (0–9). The challenge task is speaker dependent.

The corpus contains 24 200 utterances (34 speakers), subdivided into a training (17 000 utterances), development, and test set (3600 utterances each). These utterances have been created by convolving recordings from the Grid corpus (Cooke et al., 2006) with a binaural room impulse response (BRIR), whereby a different BRIR has been used for each set. The BRIR was measured at a position two meters directly in front of a binaural mannikin. Different BRIRs are obtained by varying the room configuration (e.g., doors open/closed, curtains drawn/undrawn). The development and test sets have been mixed with genuine binaural recordings from a domestic environment, which have been obtained over a period of several weeks in a house with two small children. On top of a quasi-stationary noise floor there are abrupt changes such as appliances being turned on/off, impact noises such as banging doors, and interfering speakers. The six signal-to-noise ratios (SNRs) employed in the challenge range from 9 dB down to −6 dB in steps of 3 dB; note that the range of SNRs has not been constructed by scaling the speech or noise amplitudes, but instead by choosing different noise segments. More details of the domestic audio corpus and the mixing process can be found in Barker et al. (submitted for publication). For the challenge, six hours of pure background noise (divided into seven subsets which were recorded on different days) were provided in addition to the noisy speech. All these data are publicly available at http://spandh.dcs.shef.ac.uk/projects/chime/PCC/datasets.html.

3. Convolutive NMF for speech enhancement

In addition to using LSTM-based ASR architectures in the back-end, we employ speech enhancement by convolutive non-negative matrix factorization as in Weninger et al. (2011a). This is to exploit two—arguably complementary—model-based approaches to coping with noise: using context information in the LSTM back-end, and retrieving a clean speech estimate in the front-end.

Our speech enhancement approach is based on the assumption that speech is corrupted by additive noise:

$$\mathbf{V} = \mathbf{V}^{(s)} + \mathbf{V}^{(n)}, \tag{1}$$

where $\mathbf{V} \in \mathbb{R}_{+}^{M \times N}$ is an observed magnitude spectrogram of noisy speech, $\mathbf{V}^{(s)}$ is the (true) spectrogram of the speech signal, and $\mathbf{V}^{(n)}$ is the (true) noise spectrogram. Furthermore, we assume that both the speech and noise spectrograms can be modeled as convolutions of base spectrograms (dictionaries) $\mathbf{X}^{(s)}(j) \in \mathbb{R}_{+}^{M \times P}$, $j = 1, \ldots, R^{(s)}$, respectively $\mathbf{X}^{(n)}(j)$, $j = 1, \ldots, R^{(n)}$, with non-negative activations $\mathbf{H}^{(s)} \in \mathbb{R}_{+}^{R^{(s)} \times N}$, $\mathbf{H}^{(n)} \in \mathbb{R}_{+}^{R^{(n)} \times N}$:

$$\mathbf{V}^{(s)}_{:,t} \approx \sum_{j=1}^{R^{(s)}} \sum_{p=1}^{\min\{P,t\}} \mathbf{H}^{(s)}_{j,t-p+1} \mathbf{X}^{(s)}_{:,p}(j), \tag{2}$$

$$\mathbf{V}^{(n)}_{:,t} \approx \sum_{j=1}^{R^{(n)}} \sum_{p=1}^{\min\{P,t\}} \mathbf{H}^{(n)}_{j,t-p+1} \mathbf{X}^{(n)}_{:,p}(j), \tag{3}$$

for $1 \le t \le N$. Let $\mathbf{X}_{:,j}$, symbolize the jth column of $\mathbf{X}$ as a column vector. Defining

$$\mathbf{W}^{(s)}(p) = [\mathbf{X}^{(s)}_{:,p+1}(1) \cdots \mathbf{X}^{(s)}_{:,p+1}(R^{(s)})], \tag{4}$$

$p = 0, \ldots, P-1$ and $\mathbf{W}^{(n)}(p)$ analogously, one obtains an NMF-alike notation of this signal model. Here, the approximation of $\mathbf{V}^{(s)}$ and $\mathbf{V}^{(n)}$ is denoted by $\mathbf{\Lambda}^{(s)}$ and $\mathbf{\Lambda}^{(n)}$, and $\overset{p\rightarrow}{\cdot}$ introduces a matrix 'shift' where the entries are shifted p spots to the right, filling with zeros from the left:

$$\mathbf{V} \approx \mathbf{\Lambda}^{(s)} + \mathbf{\Lambda}^{(n)} = \sum_{p=0}^{P-1} \mathbf{W}^{(s)}(p) \overset{p\rightarrow}{\mathbf{H}^{(s)}} + \sum_{p=0}^{P-1} \mathbf{W}^{(n)}(p) \overset{p\rightarrow}{\mathbf{H}^{(n)}} \tag{5}$$

In the remainder of this paper, we assume that both $\mathbf{W}^{(s)}(p)$ and $\mathbf{W}^{(n)}(p)$ can be estimated from training data, as shown in Section 7.2. The speech enhancement problem is thus reduced to finding non-negative coefficients (activations) $\mathbf{H}^{(s)}$ and $\mathbf{H}^{(n)}$ that match the observed spectra in $\mathbf{V}$—then, the estimated clean speech spectrogram $\widehat{\mathbf{V}}^{(s)}$ is obtained by filtering the observed spectrogram $\mathbf{V}$:

$$\widehat{\mathbf{V}}^{(s)} = \frac{\mathbf{\Lambda}^{(s)}}{\mathbf{\Lambda}^{(s)} + \mathbf{\Lambda}^{(n)}} \otimes \mathbf{V}. \tag{6}$$

where the symbol $\otimes$ corresponds to the elementwise matrix product and the fraction denotes an elementwise division. This approach to NMF speech enhancement is known as 'soft masking' or 'Wiener filtering'; in our case, it is used to reduce artifacts caused by the mismatch of NMF dictionaries and the observed spectra.

To jointly determine a solution for $\mathbf{H}^{(s)}$ and $\mathbf{H}^{(n)}$, we iteratively minimize the element-wise sum of the β-divergence d_β between the observed spectrogram $\mathbf{V}$ and the approximation $\mathbf{\Lambda} := \mathbf{\Lambda}^{(s)} + \mathbf{\Lambda}^{(n)}$:

$$d_\beta(\mathbf{V}|\mathbf{\Lambda}) = \sum_{i=1}^{N}\sum_{j=1}^{M} d_\beta(\mathbf{V}_{i,j}|\mathbf{\Lambda}_{i,j}), \tag{7}$$

starting from a (Gaussian) random solution. In NMF-based speech enhancement, using d_1 (equivalent to the generalized Kullback–Leibler divergence) is very popular (Smaragdis, 2007; Wilson et al., 2008; Raj et al., 2010), since it seems to provide a good compromise between separation quality and computational effort.

The minimization of d_1 (7) is performed by the multiplicative update algorithm for convolutive NMF proposed by Smaragdis (2007) and Wang et al. (2009), which can be very efficiently implemented using linear algebra routines employing vectorization. Note that the asymptotic complexity of this algorithm is polynomial ($O(RMNP)$), and linear in each of $R := R^{(s)} + R^{(n)}$, M, N, and P. All experiments for this paper were performed with the NMF implementations found in our open-source toolkit openBliSSART (Weninger et al., 2011b) to enforce reproducibility of our results.

4. Bidirectional LSTM context modeling

4.1. BLSTM networks

A simple and widely used technique for context-sensitive sequence labeling based on neural networks is the application of *recurrent* neural networks. RNNs are able to model a certain amount of context by using cyclic connections and can in principle map from the entire *history* of previous inputs to each output. Yet, the analysis of the error flow in conventional recurrent neural nets resulted in the finding that long-range context is inaccessible to standard RNNs since the backpropagated error either blows up or decays over time (vanishing gradient problem, see Hochreiter et al., 2001). This led to various attempts to address the problem of vanishing gradients for RNN, including non-gradient-based training (Bengio et al., 1994), time-delay networks (Schaefer et al., 2008; Lin et al., 1996; Lang et al., 1990), hierarchical sequence compression (Schmidhuber, 1992), and echo state networks (Jaeger, 2001). One of the most effective techniques is the Long Short-Term Memory architecture originally introduced by Hochreiter and Schmidhuber (1997). LSTM networks are able to store information in linear memory cells over a longer period of time. They overcome the vanishing gradient problem and can learn the optimal amount of contextual information relevant for the classification task.

LSTM networks can be interpreted as RNNs in which the hidden neurons are replaced by so-called *memory blocks*. Similar to the cyclic connections in RNNs, these memory blocks are recurrently connected. Every memory block consists of self-connected *memory cells* and three multiplicative *gate* units (input, output, and forget gates). Since these gates allow for write, read, and reset operations within a memory block, an LSTM block can be seen as (differentiable) memory chip in a digital computer. The overall effect of the gate units is that the LSTM memory cells can store and access information over long periods of time and thus avoid the vanishing gradient problem. For instance, as long as the input gate remains closed (corresponding to an input gate activation close to zero), the activation of the cell will not be overwritten by new inputs and can therefore be made available to the net much later in the sequence by opening the output gate. This allows to bridge long time lags between relevant inputs and outputs, which would not be possible with standard RNNs. Fig. 1 shows the architecture of a memory block containing one memory cell.

If α_t^{in} denotes the activation of the input gate at time t *before* the activation function f_g has been applied and β_t^{in} represents the activation *after* application of the activation function, the input gate activation (forward pass) of a certain memory block can be written as

$$\alpha_t^{\text{in}} = \sum_{i=1}^{I}\eta^{i,\text{in}}x_t^i + \sum_{h=1}^{H}\eta^{h,\text{in}}\beta_{t-1}^h + \sum_{c=1}^{C}\eta^{c,\text{in}}s_{t-1}^c \tag{8}$$

and

$$\beta_t^{\text{in}} = f_g(\alpha_t^{\text{in}}), \tag{9}$$

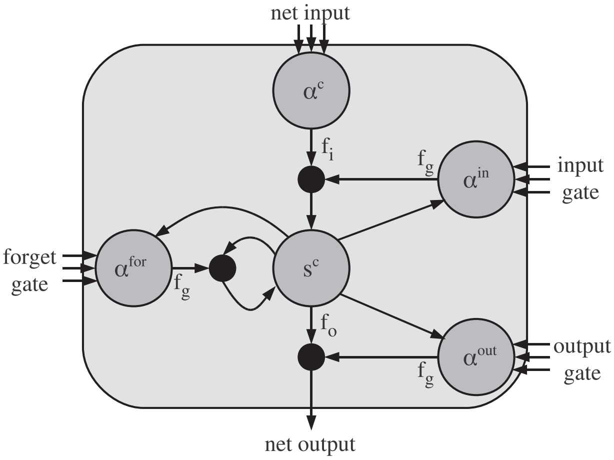

Fig. 1. LSTM memory block consisting of one memory cell: the input, output, and forget gates collect activations from inside and outside the block which control the cell through multiplicative units (depicted as small circles); input, output, and forget gate scale input, output, and internal state, respectively; f_i, f_g, and f_o denote activation functions; the recurrent connection of fixed weight 1.0 maintains the internal state.

respectively. Since Eq. (8) refers to one specific memory block in an LSTM network, all variables are scalars. The variable η^{ij} corresponds to the weight of the connection from unit i to unit j while ‘in’, ‘for’, and ‘out’ refer to input gate, forget gate, and output gate, respectively. Indices i, h, and c count the inputs x_t^i, the cell outputs from other blocks in the hidden layer, and the memory cells, while I, H, and C are the number of inputs, the number of cells in the hidden layer, and the number of memory cells in one block. Finally, s_t^c corresponds to the *state* of a cell c at time t, meaning the activation of the linear cell unit.

Similarly, the activation of the forget gates before and after applying f_g can be calculated as follows:

$$\alpha_t^{\text{for}} = \sum_{i=1}^{I} \eta^{i,\text{for}} x_t^i + \sum_{h=1}^{H} \eta^{h,\text{for}} \beta_{t-1}^h + \sum_{c=1}^{C} \eta^{c,\text{for}} s_{t-1}^c \tag{10}$$

$$\beta_t^{\text{for}} = f_g(\alpha_t^{\text{for}}). \tag{11}$$

The memory cell value α_t^{c} is a weighted sum of inputs at time t and hidden unit activations at time $t-1$:

$$\alpha_t^{\text{c}} = \sum_{i=1}^{I} \eta^{i,\text{c}} x_t^i + \sum_{h=1}^{H} \eta^{h,\text{c}} \beta_{t-1}^h. \tag{12}$$

To determine the current state of a cell c, we scale the previous state by the activation of the forget gate and the input $f_i(\alpha_t^{\text{c}})$ by the activation of the input gate:

$$s_t^{\text{c}} = \beta_t^{\text{for}} s_{t-1}^{\text{c}} + \beta_t^{\text{in}} f_i(\alpha_t^{\text{c}}). \tag{13}$$

The computation of the output gate activations follows the same principle as the calculation of the input and forget gate activations, however, this time we consider the *current* state s_t^c, rather than the state from the previous time step:

$$\alpha_t^{\text{out}} = \sum_{i=1}^{I} \eta^{i,\text{out}} x_t^i + \sum_{h=1}^{H} \eta^{h,\text{out}} \beta_{t-1}^h + \sum_{c=1}^{C} \eta^{c,\text{out}} s_t^c \tag{14}$$

$$\beta_t^{\text{out}} = f_g(\alpha_t^{\text{out}}). \tag{15}$$

Finally, the memory cell output is determined as

$$\beta_t^{\text{c}} = \beta_t^{\text{out}} f_o(s_t^{\text{c}}). \tag{16}$$

Fig. 2 provides an overview over the connections in an ‘unrolled’ LSTM network for time steps $t-1$ and t. For the sake of simplicity, this network only contains small input and output layers (two nodes each) and just one memory

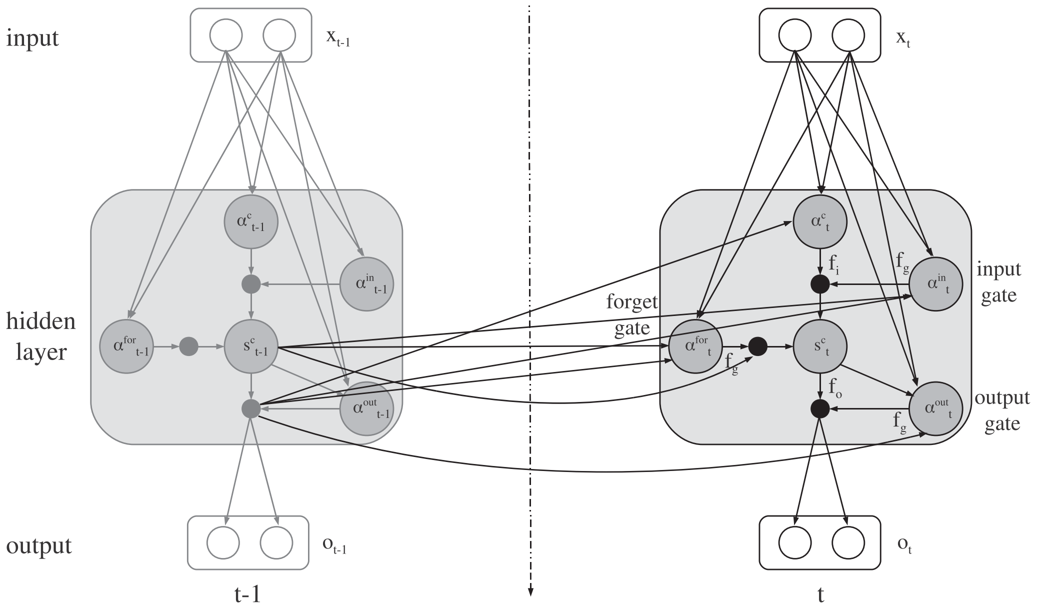

Fig. 2. Connections in an LSTM network consisting of two input nodes, one memory cell with one memory block, and two output nodes.

block with one cell. Note that the initial version of the LSTM architecture contained only input and output gates. Forget gates were added later by Gers et al. (2000) in order to allow the memory cells to reset themselves whenever the network needs to *forget* past inputs. In our experiments we exclusively consider the enhanced LSTM version including forget gates.

Another shortcoming of standard RNNs is that they have access to past but not to future context. This can be overcome by using *bidirectional* RNNs (see Schuster and Paliwal, 1997), where two separate recurrent hidden layers scan the input sequences in opposite directions. The two hidden layers are connected to the same output layer, which therefore has access to context information in both directions. The amount of context information that the network actually uses is learned during training, and does not have to be specified beforehand. In this article we use a combination of the principle of bidirectional networks and the LSTM technique (i.e., bidirectional LSTM). Of course the usage of bidirectional context implies a short look-ahead buffer, meaning that recognition cannot be performed truly on-line. However, for many speech recognition tasks it is sufficient to obtain an output, e.g., at the end of an utterance, so that both forward and backward context can be used during decoding.

In recent years, the LSTM technique has been successfully applied for a variety of pattern recognition tasks, including phoneme classification (Graves and Schmidhuber, 2005), emotion recognition (Wöllmer et al., 2010b), handwriting recognition (Graves et al., 2008), and driver distraction detection (Wöllmer et al., 2011a).

4.2. Connectionist temporal classification

One possible approach to use BLSTM networks for small vocabulary speech recognition is to train the network directly on the words contained in the vocabulary, so that the network learns a mapping from speech features to words. The Connectionist Temporal Classification objective function proposed by Graves et al. (2006) can be applied to obtain a temporal warping, i.e., to obtain a sequence of words from a (framewise) sequence of speech feature vectors. Thus, CTC allows recurrent neural networks to map *unsegmented* sequential data onto a sequence of labels. The output of a network trained with CTC typically consists of a series of spikes corresponding to words that are detected in the speech signal. These spikes are separated by periods during which the *blank* output unit is activated.

CTC allows the network to choose the location as well as the class of each label. By summing up over all sets of label locations that yield the same label sequence, CTC determines a probability distribution over possible labelings, conditioned on the input sequence.

The CTC layer as used in our experiments has as many output units as there are distinct words in the vocabulary, plus an extra *blank* unit. The activations of the outputs at each timestep are normalized and interpreted as the probability of observing the corresponding word at that point in the speech sequence. Because these probabilities are conditionally independent given the input sequence, the total probability of a given (framewise) sequence $w^f_{1:T}$ of blanks and words is

$$p(w^f_{1:T}|\mathbf{x}_{1:T}) = \prod_{t=1}^{T} o_t^{w_t^f}, \tag{17}$$

where $\mathbf{x}_{1:T}$ is a length T input sequence and o_t^k is the activation of output unit k at time t. In order to sum over all the output sequences corresponding to a particular labeling (regardless of the *location* of the labels) we define an operator $\mathcal{B}(\cdot)$ that removes first the repeated labels and then the blanks from the output sequence so that, e.g., $\mathcal{B}(AA--BBB-B) = ABB$. This implies that repeated labels can be detected, as long as there is a blank label between them in the original output sequence. If no blank label can be found between two identical labels, we assume that the repeated labels belong to the same event. The total probability of the length L labeling $w_{1:L}$, where $L \leq T$, is then

$$p(w_{1:L}|\mathbf{x}_{1:T}) = \sum_{w^f_{1:T}:\mathcal{B}(w^f_{1:T})=w_{1:L}} p(w^f_{1:T}|\mathbf{x}_{1:T}). \tag{18}$$

A naive calculation of Eq. (18) is not feasible, because the number of $w^f_{1:T}$ terms corresponding to each labeling increases exponentially with the sequence length. However, $p(w_{1:V}|\mathbf{x}_{1:T})$ can be efficiently calculated with a dynamic programing algorithm similar to the forward-backward algorithm for HMMs (see Graves et al., 2006). The CTC objective function O^{CTC} is defined as the negative log likelihood of the training set $\mathbb{S}$

$$O^{CTC} = -\sum_{(\mathbf{x}_{1:T},w_{1:L})\in\mathbb{S}} \ln\, p(w_{1:L}|\mathbf{x}_{1:T}). \tag{19}$$

An RNN with a CTC output layer can be trained with gradient descent via backpropagation through time, using the partial derivatives of O^{CTC} with respect to the output activations (for details, see Graves et al., 2006).

5. Non-negative sparse classification

As an alternative method to obtain framewise word predictions from a low-level speech feature vector sequence, we also investigate the principle of non-negative sparse classification. It is based on decomposition in the spectral domain rather than long-range context modeling of speech features; similarly to supervised NMF speech enhancement, the main idea is to use the results of spectral factorization directly for speech recognition by determining the sources which contribute to a mixed observation. To this end, the non-negative activation weights of dictionary atoms are determined by applying sparse NMF. As the identities of the atoms correspond to the phonetic content, phone or word classification can be performed based on the activation weights. In our NSC experiments, atoms represent sampled spectrogram patterns and thus are called ‘exemplars’. This is in contrast to the approach pursued for speech enhancement, where atoms are learned from training data—in fact, using the very same NSC approach for source separation has been shown to be inferior to the convolutive NMF enhancement pursued in this paper (Gemmeke et al., 2011b). Thus, while there is some methodological overlap between NSC and NMF enhancement, the parametrization of the algorithms are considerably different and further improvements are expected when combining them. Further details on the applied NSC technique can be found in previous publications by Gemmeke et al. (2011a) and Hurmalainen et al. (2011).

For NSC, we use 26 Mel-scale spectral magnitude bands as features, employing the common frame size of 25 ms and a 10 ms frame shift. We use exemplar windows spanning 20 frames and factorize each window independently as in experiments by Hurmalainen et al. (2011). Other factorization options, including weighting of features, sparsity penalty values and the number of iterations were exactly set as by Hurmalainen et al. (2011). For the sparse classification task, 5000 speaker-dependent speech exemplars and 5000 noise exemplars are extracted from the training data by random sub-sampling without overlap; the speech exemplars are balanced with respect to phonetic content (as determined by forced alignment, cf. below). This combined speech-noise basis is kept fixed during NMF iterations. After receiving the sparse activation weight vector for each window, the weights and the predetermined label sequences encoding the phonetic information of speech exemplars are used to construct a state likelihood matrix for the observation. The speech exemplars are labeled with the corresponding states of the CHiME baseline HMM recognizer (cf. the next section) by means of forced alignment. For details of this NSC setup and its standalone recognition results in a hybrid ASR system see (Hurmalainen et al., 2011). In this work, we determine the most likely word identity n_t for each frame t of the observation by summing state likelihoods corresponding to each word. The resulting sequence of word predictions is then used as a feature stream in a multi-stream decoder (see Section 6.5). Note that in our experiments, NSC is always applied on noisy speech, not NMF-enhanced speech; in a previous study (Weninger et al., 2012), we found that applying NSC to NMF-enhanced MFB features did not further improve performance, probably due to a mismatch between the noise dictionaries and the noise that is left over after speech separation.

6. Evaluated ASR systems

6.1. Baseline HMM

The baseline recognition system, as provided by the 2011 CHiME Challenge organizers, employs 51 word-level Hidden Markov Models (Barker et al., submitted for publication). The HMMs use a left-to-right model topology with no state skips. In order to model the different lengths of the words in the vocabulary, two states per phoneme are used. This results in a varying number of states per word (between 4 and 10). State emission probabilities are modeled using seven Gaussian mixture components per state with diagonal covariance matrices.

The models are trained starting with a single Gaussian and applying iterative mixture splitting and EM training. After each EM iteration, the number of mixture components is increased by splitting the component with the largest mixture weight. This is repeated until the final number of seven Gaussian mixtures is reached. For recognition, the baseline system uses a grammar which strictly follows the grammar of the Grid corpus utterances (see Section 2).

We evaluated several minor modifications of the baseline HMM system, including a larger number of Gaussian mixtures (up to 15) and the incorporation of a silence model. However, as these changes of the baseline recognizer did not result in an increased keyword recognition accuracy on the development set, we decided to employ the HMM system as provided by the CHiME Challenge organizers as baseline system.

The features used for the baseline HMM consist of standard 39-dimensional cepstral mean normalized MFCCs (12 Mel-cepstral coefficients and the logarithmic energy plus the corresponding delta and acceleration coefficients) computed from overlapping frames with a frame length of 25 ms and a frame shift of 10 ms. For better comparability with Barker et al. (submitted for publication), we only used cepstral mean normalization to generate the baseline MFCC features and did not apply more complex normalization, filtering, and enhancement methods other than the investigated NMF-based approach. Specifically, we did not integrate the ETSI advanced front-end (AFE) into our system since we found that the voice activity detection which is part of the AFE—and which is required for the AFE algorithm—tends to fail as soon as the SNR level is negative (Schuller et al., 2009).

6.2. Tandem BLSTM-HMM approach

As a first attempt to improve the baseline HMM system via feature-level BLSTM modeling, we evaluated a BLSTM front-end as extension of the standard MFCC features. Thus, we trained a BLSTM network for *framewise* word prediction (without CTC), i.e., the network inputs correspond to the 39 cepstral mean normalized MFCC features and the resulting output activations represent the posterior probabilities of the 51 words. Even though it would also be possible to train the networks on phone posteriors, we decided to use words as targets, since the baseline HMM system also applies word models rather than phoneme models. Thus, in each time frame, we obtain a vector of 51 output activations which is logarithmized and appended to the original 39-dimensional MFCC feature vector, resulting in 90 Tandem features per time step. Not including the original MFCC features in the Tandem feature vector results in lower accuracies, as shown by Wöllmer and Schuller (2011). Next, we decorrelate these features using principal component analysis (PCA) and apply only the first 40 principal components for HMM-based recognition. A flowchart of the Tandem BLSTM front-end processing NMF-enhanced speech can be seen in Fig. 3.

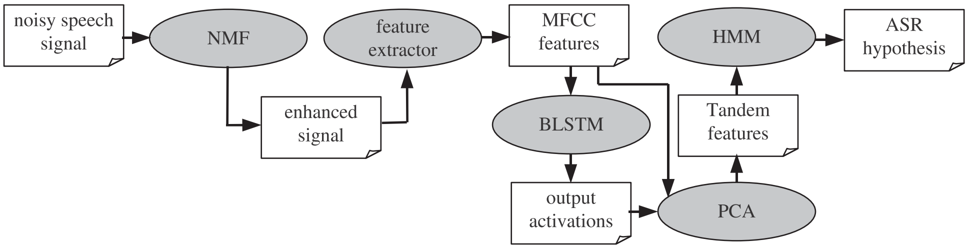

Fig. 3. Flowchart of the Tandem BLSTM-HMM recognizer processing speech enhanced via NMF.

6.3. CTC system

Using a CTC output layer, a word hypothesis can be obtained without HMM decoding (see Section 4.2). Hence, we evaluated a CTC back-end, replacing the baseline HMM system. Again, output activations represent occurrences of words. Note that purely CTC-based recognition is rather suited for small to medium vocabulary tasks, since for large vocabulary ASR the network output layer would get too large. The recognition grammar of our CTC framework is not restricted in any way, meaning that any word can be detected at any time. To determine the keyword recognition rate, we simply take the first letter and digit that are detected in an utterance. Applying our CTC recognizer, we evaluate two different front-ends: the conventional MFCC features and the Tandem BLSTM-MFCC feature extractor explained in Section 6.2.

6.4. Multi-stream BLSTM-HMM

We also test our recently introduced multi-stream BLSTM-HMM recognizer (Wöllmer et al., 2011b) as a further method to integrate LSTM modeling into speech decoding. Employing the same framewise BLSTM word predictor as outlined in Section 6.2, we generate a discrete word prediction feature b_t for each time step, corresponding to the index of the estimated word that can be obtained by determining the maximum BLSTM output activation:

$$b_t = \underset{w}{\operatorname{argmax}}(o_t^1, \ldots, o_t^w, \ldots, o_t^V). \tag{20}$$

In every time frame t the multi-stream HMM uses two independent observations: the MFCC features $\mathbf{x}_t$ and the BLSTM word prediction feature b_t. With $\mathbf{y}_t = [\mathbf{x}_t b_t]$ being the joint feature vector and the variables λ_1 and λ_2 denoting the stream weight of the MFCC stream and the BLSTM stream, respectively, the multi-stream HMM emission probability while being in a certain state s_t can be written as

$$p(\mathbf{y}_t|s_t) = \left[\sum_{m=1}^{M} c_{s_t m} \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_{s_t m}, \boldsymbol{\Sigma}_{s_t m})\right]^{\lambda_1} \times p(b_t|s_t)^{\lambda_2}. \tag{21}$$

Thus, the continuous MFCC observations are modeled via a mixture of M Gaussians per state while the BLSTM prediction is modeled using a discrete probability distribution $p(b_t|s_t)$. The index m denotes the mixture component, $c_{s_t m}$ is the weight of the m'th Gaussian associated with state s_t, and $\mathcal{N}(\cdot\,; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ represents a multivariate Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.

The main advantage of the multi-stream approach compared to the Tandem features is that the BLSTM can be integrated without time-consuming re-estimation of Gaussian mixture components. According to our experience, using the vector of logarithmized word posteriors as continuous features in the second stream results in lower ASR accuracies when compared to the multi-stream system using the discretized word prediction b_t, which is why this alternative approach was not evaluated in the following.

Using the development set, we optimized the stream weights independently for speaker independent and speaker adapted BLSTM nets, resulting in an optimum of $\lambda_1 = 1.3$ and $\lambda_2 = 0.7$ for speaker independent networks and $\lambda_1 = 1.1$ and $\lambda_2 = 0.9$ for speaker dependent networks. Fig. 4 shows a flowchart of the multi-stream BLSTM-HMM.

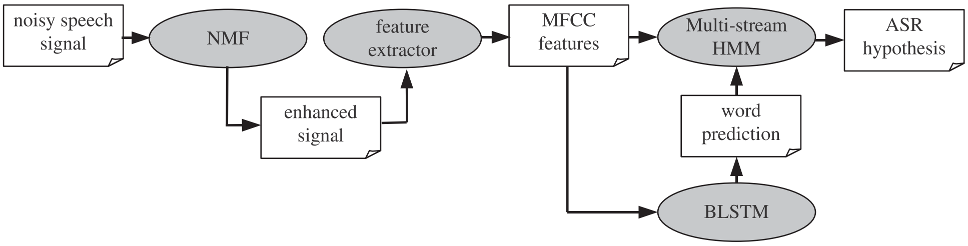

Fig. 4. Flowchart of the multi-stream BLSTM-HMM recognizer processing speech enhanced via NMF.

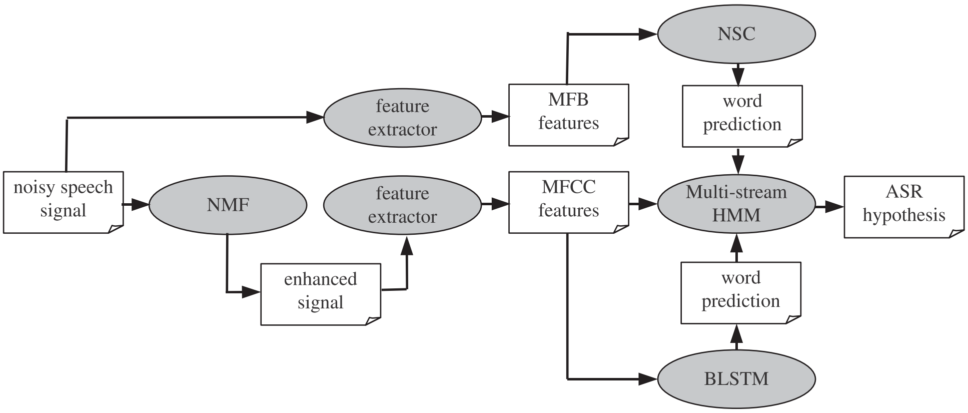

Fig. 5. Flowchart of the triple-stream recognizer exploiting word predictions obtained via BLSTM and NSC.

6.5. Triple-stream HMM exploiting BLSTM and NSC word predictions

To exploit both the BLSTM-based word prediction feature and the word prediction obtained via non-negative sparse classification (see Section 5) in addition to the MFCC feature stream, we implemented a triple-stream HMM architecture, which can be seen in Fig. 5. Similar to the multi-stream recognition architecture described in Section 6.4, the HMM uses continuous MFCC features as well as the discrete BLSTM feature b_t and the word prediction obtained by NSC (n_t) as three independent streams of observations. In contrast to the NSC-only decoder proposed in Hurmalainen et al. (2011), using NSC in a multi-stream approach along with MFCC and BLSTM predictions can be useful to exploit the properties of spectral (such as additiveness) and cepstral representation (such as a degree of speaker independence) in parallel.

The triple-stream HMM emission probability in a certain state s_t can be written as

$$p(\mathbf{y}_t|s_t) = \left[\sum_{m=1}^{M} c_{s_t m}\mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_{s_t m}, \boldsymbol{\Sigma}_{s_t m})\right]^{\lambda_1} \times p(b_t|s_t)^{\lambda_2} \times p(n_t|s_t)^{\lambda_3}. \tag{22}$$

Best results on the development set could be obtained when Mel-frequency bands (MFB) that are computed from the raw speech signal (i.e., the signal not enhanced via NMF) are used as input for non-negative sparse classification (see also Fig. 5). Stream weights were set to $\lambda_1 = \lambda_2 = \lambda_3 = 1$.

7. Experiments

7.1. Preprocessing

The binaural audio signals were down-mixed from stereo to mono by averaging channels. For NMF speech enhancement, they were transformed to the spectral domain by short-time Fourier transformation using a window size of 64 ms (corresponding to 1024 samples at a sample rate of 16 kHz) and 75% overlap, i.e., 16 ms frame shift. This kind of parametrization has been proven to deliver excellent results in speech enhancement (Smaragdis, 2007; Raj et al.,

2010) at an acceptable computational effort. We use the square root of the Hann function for windowing both in forward and backward transformation in order to reduce artifacts, as proposed by Helen and Virtanen (2005). As in Weninger et al. (2011a), the Mel filter bank for MFCC feature extraction was modified to have a cutoff frequency of 5000 Hz.

7.2. Dictionaries for NMF-based speech enhancement

As sketched in Section 3, our approach for NMF speech enhancement uses convolutive bases of both speech and noise which are learned from training data. However, in contrast to purely unsupervised learning algorithms for speech dictionaries as proposed, e.g., by Schmidt and Olsson (2006) using basic NMF and by Smaragdis (2007) using convolutive NMF, we exploit knowledge about the speech recognition task already in dictionary learning. This is partly motivated by the study by Raj et al. (2011) who found that in the context of speech enhancement for large vocabulary continuous speech recognition, incorporating phonetic information into NMF by using phoneme-dependent speech dictionaries is highly beneficial. However, in contrast to that study, which uses single spectra to model phonemes, we exploit convolutive NMF for the fact that it is very well suited to capturing spectral sequences corresponding to words (Smaragdis, 2004). Hence, convolutive NMF appears to be particularly suited to the small vocabulary CHiME recognition task.

In summary, in our approach each dictionary entry corresponds to a ‘characteristic’ spectrogram of a certain word ($R^{(s)} = 51$) that is learned from training examples. Since we further use speaker-dependent dictionaries for the separation, the characteristic spectrograms are obtained from the training set by convolutive NMF as follows. For each of the 34 speakers, we used the forced alignments, obtained by the baseline HMM-MFCC recognizer on the noise-free training set of the CHiME corpus, to extract all occurrences of each word (51 words in total). Then, for each speaker $k \in \{1, \ldots, 34\}$ and word $w \in \{1, \ldots, 51\}$, we concatenated the magnitude spectra into a matrix $\mathbf{T}^{(s,k,w)}$, which was reduced to convolutive base $\mathbf{w}^{(s,k,w)}(p)$ by a 1-component convolutive NMF,

$$\mathbf{T}^{(s,k,w)} \approx \sum_{p=0}^{P-1} \mathbf{w}^{(s,k,w)}(p) \overset{p\rightarrow}{\mathbf{h}^{(s,k,w)}}, \tag{23}$$

and formed a speaker-dependent dictionary

$$\mathbf{W}^{(s,k)}(p) = [\mathbf{w}^{(s,k,1)}(p) \cdots \mathbf{w}^{(s,k,51)}(p)]. \tag{24}$$

The parameter P was set to 13 through inspection of the word lengths in the CHiME corpus training set. This corresponds to a spectrogram of a 256 ms signal segment at 64 ms window size and 16 ms frame shift, which is enough to cover the lengths of the CHiME set of words in most cases.

In contrast to the speech, the background noise is assumed to be highly variable. Thus, to create a noise dictionary as general as possible, we sub-sampled the set of training noise (approximately 6 hours) available for the challenge, selecting 4000 random segments of 256 ms length, concatenated them into a spectrogram $\mathbf{T}^{(n)}$, and reduced them to a dictionary $\mathbf{W}^{(n)}(p)$. In analogy to the speech dictionary, it contains 51 characteristic noise spectrograms ($R^{(n)} = 51$). The sizes of the speech and noise dictionaries were chosen to be equal following previous studies on supervised NMF-based source separation (Schmidt and Olsson, 2006; Smaragdis, 2007; Gemmeke et al., 2011b).

7.3. Training and network parametrization

For increased robustness, multi-condition training (MCT) is performed by adding noisy speech to the training data. This noisy training data is obtained by mixing all 17 000 training utterances with random segments of the training noise provided in the CHiME corpus. Thus, the complete clean and noisy training database consists of 34 000 utterances. Since the training noise provided by the CHiME Challenge organizers consists of seven different background noise recordings, we also evaluated a larger MCT training set of 136 000 utterances, comprising the clean training utterances as well as seven different noisy versions of the training material, created by superposing the clean utterances with random segments of all seven noise recordings. However, since the performance gain compared to the smaller MCT set is relatively small, at the cost of an increased training time, we decided to use the smaller MCT set of 34 000 utterances for our experiments. Furthermore, we found that adding NMF-enhanced noisy speech to the MCT set did not further improve performance, which is an interesting result since it indicates that the back-end can naturally cope with the distortions introduced by NMF enhancement without further adaptation.

The BLSTM network applied for generating the Tandem features and the estimates b_t for the multi-stream systems was trained on framewise word targets obtained via HMM-based forced alignment of the clean training set. By contrast, the CTC network was trained on the unsegmented ground truth transcription of the training corpus. Similar to the network configuration used by Wöllmer et al. (2011b), the BLSTM network consisted of three hidden LSTM layers (per input direction) with a size of 78, 150, and 51 hidden units, respectively. Each LSTM memory block contained one memory cell. The remaining training configurations were the same as those used by Wöllmer et al. (2011b).

7.4. Speaker adaptation

We investigated various techniques to create speaker adapted recognition systems: first, we created speaker dependent HMMs by adapting means and variances of the speaker-independent HMMs, performing additional EM iterations using the training utterances for each speaker. This procedure is equivalent to the one applied for the baseline CHiME Challenge results. Second, we opted for mean-only MAP adaptation (with factor $\tau = 5$) as employed by Weninger et al. (2011a). Note that for all speaker adaptation methods, we exclusively used material from the *training* set.

Finally, we also adapted the BLSTM and CTC networks by performing additional training epochs using only the training utterances of the respective speaker. All network weights were initialized with the weights of the speaker independent networks and training was aborted as soon as no further improvement on the development set could be observed. Note that for experiments using multi-condition training, we use multi-condition training data also for speaker adaptation.

8. Results and discussion

Table 1 shows the keyword recognition accuracies obtained for the various system combinations on the development set of the CHiME corpus. The first row corresponds to the challenge baseline result (56.30% mean accuracy) using MFCC features and speaker adapted HMMs (Barker et al., submitted for publication). Applying multi-condition training increases the mean performance to 69.85%. A further gain is obtained by convolutive NMF as detailed in Section 3, leading to an average accuracy of 80.92% for a comparable HMM system and to 82.65% for a MAP adapted recognizer.

8.1. The effect of speaker adaptation

As expected, all speaker adaptation techniques increase the keyword recognition accuracies of the respective systems. For the baseline MFCC-HMM system, a large improvement from 73.20% to 80.92% is observed when adapting HMMs by re-training the models employing speaker-specific training material. A further 1.73% (absolute) gain is reached by MAP adaptation of the HMMs. Interestingly the performance difference between speaker-independent HMMs and re-trained speaker adapted HMMs is considerably smaller when BLSTM-modeling is applied in the front-end (83.85% vs. 84.16% for the Tandem BLSTM-HMM front-end and 84.97% vs. 86.36% for the multi-stream BLSTM-HMM). This indicates that BLSTM features are less speaker-specific than conventional MFCCs. Also for CTC back-ends, speaker adaptation boosts recognition performance (80.43% vs. 85.46% when using MFCC features and 82.95% vs. 84.57% when applying Tandem features). Finally, also framewise BLSTM word predictors tend to produce better Tandem features/word estimates when speaker-specific training material is used to adapt the networks.

8.2. MFCC features vs. tandem features

Tandem features based on bidirectional Long Short-Term Memory modeling (see Section 6.2) consistently outperform standard MFCC features: using speaker adapted networks, performance can be boosted from 82.65 to 87.69% for an HMM system and from 85.46 to 87.60% for a CTC back-end. Note, however, that the performance gain achieved via Tandem features is much smaller when applying a CTC back-end. Thus, BLSTM modeling in the front- and back-end seem to be not fully complementary.

Table 1
Development set: keyword recognition accuracies [%] for different SNR levels applying NMF, multi-condition training (MCT), MFCC, Tandem BLSTM-MFCC, or word prediction features (b_t, n_t) in combination with HMM, CTC, or multi-stream (MS) back-ends. Speaker adaptation techniques: MAP adaptation of HMMs and re-training of BLSTM, CTC, and/or HMM recognizers.

NMF	MCT	Features	Back-end	Speaker adaptation				SNR						Mean
				BLSTM	CTC	HMM	MAP	−6 dB	−3 dB	0 dB	3 dB	6 dB	9 dB	
×	×	MFCC	HMM	–	–	√	×	31.08	36.75	49.08	64.00	73.83	83.08	56.30
×	√	MFCC	HMM	–	–	√	×	47.25	55.67	66.33	76.08	84.67	89.08	69.85
√	√	MFCC	HMM	–	–	×	×	63.75	66.33	71.67	75.92	79.92	81.58	73.20
√	√	MFCC	HMM	–	–	√	×	70.33	76.08	80.08	83.17	88.08	87.75	80.92
√	√	MFCC	HMM	–	–	√	√	73.58	77.33	82.17	84.25	88.58	90.00	82.65
√	√	MFCC	CTC	–	×	–	–	71.00	73.67	79.50	82.42	87.25	88.75	80.43
√	√	MFCC	CTC	–	√	–	–	77.00	81.00	84.58	87.50	90.58	92.08	85.46
√	√	Tandem	HMM	×	–	×	×	75.75	78.05	83.42	85.73	89.58	90.58	83.85
√	√	Tandem	HMM	×	–	√	×	74.08	79.72	83.58	86.56	89.17	91.83	84.16
√	√	Tandem	HMM	×	–	√	√	77.09	80.38	84.50	87.48	91.00	92.75	85.53
√	√	Tandem	HMM	√	–	√	√	78.34	84.72	87.08	89.73	92.33	93.92	87.69
√	√	Tandem	CTC	×	×	–	–	74.08	78.42	81.92	85.17	88.42	89.67	82.95
√	√	Tandem	CTC	×	√	–	–	75.92	79.58	83.58	87.08	90.50	90.75	84.57
√	√	Tandem	CTC	√	√	–	–	79.17	84.25	87.00	89.67	92.08	93.42	87.60
√	√	MFCC, b_t	MS-HMM	×	–	×	×	77.08	80.33	84.17	88.08	89.25	90.92	84.97
√	√	MFCC, b_t	MS-HMM	×	–	√	×	78.67	81.75	85.67	88.67	90.83	92.58	86.36
√	√	MFCC, b_t	MS-HMM	×	–	√	√	81.50	83.00	86.75	90.58	92.25	93.67	87.96
√	√	MFCC, b_t	MS-HMM	√	–	√	√	83.36	86.73	90.00	91.49	94.08	95.00	90.11
√	√	MFCC, b_t, n_t	MS-HMM	√	–	√	√	86.04	89.48	92.67	94.57	96.25	96.58	**92.60**

8.3. HMM vs. CTC back-end

Replacing the HMM back-end by a CTC network as explained in Sections 4.2 and 6.3 enhances ASR performance (82.65 vs. 85.46% for speaker adapted systems). However, when applying context-sensitive Tandem features, the performance difference between HMMs and CTC networks disappears, which indicates that also HMMs can reach improved performance if long-range context is modeled on the feature level.

8.4. Methods for BLSTM-modeling

Overall, the configurations shown in Table 1 reflect three different methods to integrate BLSTM context-modeling into an ASR system: using Tandem BLSTM-MFCC features in the front-end, applying a BLSTM-based CTC back-end, and exploiting BLSTM word predictions in a multi-stream HMM framework. When comparing the keyword recognition performances of the individual methods, we see that incorporating BLSTM-modeling in a CTC back-end (85.46% accuracy) is less effective than employing Tandem features (up to 87.69% accuracy). The highest average keyword accuracy achieved with systems not performing NSC is 90.11% and can be obtained with the speaker adapted multi-stream BLSTM-HMM outlined in Section 6.4. Hence, the multi-stream architecture seems to be the most effective strategy of applying bidirectional Long Short-Term Memory for noise robust ASR.

8.5. Non-negative sparse classification

The last line of Table 1 shows the keyword recognition accuracy of the triple-stream architecture which, in addition to the BLSTM word prediction, also takes into account the word prediction n_t generated via non-negative sparse classification as described in Sections 5 and 6.5. Compared to the best BLSTM-based multi-stream system (90.11% accuracy), the triple-stream approach enables a remarkable increase in recognition performance, leading to an average accuracy of 92.60%. Thus, we can conclude that performance gains achieved via BLSTM word predictors and NSC word predictors are complementary to a certain degree.

8.6. Test set results

Results on the CHiME test set are shown in Table 2. Generally, the same trends as for the development set can be observed. Applying convolutive NMF, multi-condition training, speaker adaptation, BLSTM modeling, and NSC leads to an impressive increase of keyword recognition accuracy from 55.93 to 91.86%. Note that when evaluating the test set, the Tandem BLSTM-HMM system as well as the BLSTM-based CTC back-end can both almost reach the performance of multi-stream BLSTM-HMM decoding with an average accuracy of 89.01 and 88.00%, respectively. However, as for the development set evaluations, the most efficient way to integrate BLSTM is the multi-stream architecture (accuracy of 90.45%). Again, NSC further improves performance (significance level $p < .01$), so that our best result of 91.86% is obtained with the triple-stream model. Our approach slightly outperforms the best CHiME Challenge contribution of 91.65% average accuracy which was reported by Delcroix et al. (2011). Their system is the result of a combination of three different systems exploiting spatial, spectral, and temporal modeling of speech and noise, in addition to dynamic variance adaptation. To shed light on the significance of performance differences, we use a correlated proportions test (Dietterich, 1998) based on the assumption that the accuracy difference between a recognizer A and a baseline B with accuracies p_a and p_b is a normally distributed random variable with mean $p_a - p_b$ and variance $2p(1-p)/S$, where $p = (p_a + p_b)/2$ and S is the number of instances. For the CHiME test set utterances at a single SNR, $S = 1200$ (600 utterances containing two keywords each); consequently, for the whole CHiME test set, $S = 7200$. We use a one-tailed test, i.e., the null hypothesis (H_0) is that $p_a \leq p_b$, or informally, A is not better than the baseline B. In Fig. 6, we show how large the accuracy improvement must be to reject H_0 at either the 0.05, 0.01, or 0.001 level: the required accuracy improvement is given by the intersection of the vertical line corresponding to the baseline accuracy and the curve corresponding to the level of significance. This test allows to easily assess the significance of any difference in accuracy encountered throughout analysis; yet, results of this test should only be interpreted as a heuristic measure, since the estimates of p_a and p_b on the test set are not independent (Dietterich, 1998). Furthermore, we do not correct for repeated measurements, using ‘significance’ not in the inferential meaning but as an objective measure of differences worthwhile to be discussed.

Table 2
Test set: keyword recognition accuracies [%] for different SNR levels applying NMF, multi-condition training (MCT), MFCC, Tandem BLSTM-MFCC, or word prediction features (b_t, n_t) in combination with HMM, CTC, or multi-stream (MS) back-ends. Speaker adaptation techniques: MAP adaptation of HMMs and re-training of BLSTM, CTC, and/or HMM recognizers.

NMF	MCT	Features	Back-end	Speaker adaptation				SNR						Mean
				BLSTM	CTC	HMM	MAP	−6 dB	−3 dB	0 dB	3 dB	6 dB	9 dB	
×	×	MFCC	HMM	–	–	√	×	30.33	35.42	49.50	62.92	75.00	82.42	55.93
×	√	MFCC	HMM	–	–	√	×	47.67	56.25	67.42	76.50	82.42	88.50	69.82
√	√	MFCC	HMM	–	–	×	×	65.92	68.33	75.33	77.67	79.92	83.33	75.08
√	√	MFCC	HMM	–	–	√	×	72.08	76.50	82.08	84.25	87.17	89.17	81.88
√	√	MFCC	HMM	–	–	√	√	75.58	79.25	84.08	87.67	88.33	90.58	84.25
√	√	MFCC	CTC	–	×	–	–	70.83	76.25	80.17	84.25	86.00	88.50	81.00
√	√	MFCC	CTC	–	√	–	–	74.92	79.25	83.33	88.08	89.50	90.92	84.33
√	√	Tandem	HMM	×	–	×	×	75.67	79.22	82.08	87.81	88.17	89.92	83.81
√	√	Tandem	HMM	×	–	√	×	76.00	79.97	84.25	87.48	88.58	91.75	84.67
√	√	Tandem	HMM	×	–	√	√	77.67	80.72	84.75	88.56	90.00	92.00	85.62
√	√	Tandem	HMM	√	–	√	√	80.42	85.64	89.17	91.57	93.00	94.25	89.01
√	√	Tandem	CTC	×	×	–	–	73.33	77.67	80.83	85.83	86.58	90.25	82.42
√	√	Tandem	CTC	×	√	–	–	74.42	79.50	82.50	87.58	87.25	91.58	83.81
√	√	Tandem	CTC	√	√	–	–	80.00	84.33	87.25	90.75	91.92	93.75	88.00
√	√	MFCC, b_t	MS-HMM	×	–	×	×	76.58	81.33	83.00	88.25	89.08	91.17	84.90
√	√	MFCC, b_t	MS-HMM	×	–	√	×	79.00	82.75	86.58	89.42	89.58	92.67	86.67
√	√	MFCC, b_t	MS-HMM	×	–	√	√	80.33	83.50	86.67	90.00	90.25	92.92	87.28
√	√	MFCC, b_t	MS-HMM	√	–	√	√	82.92	87.15	90.25	93.66	93.92	94.83	90.45
√	√	MFCC, b_t, n_t	MS-HMM	√	–	√	√	84.75	88.31	92.08	93.91	95.67	96.42	**91.86**

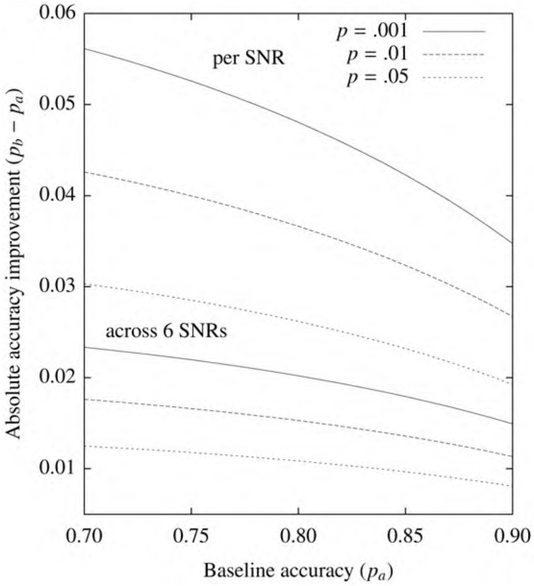

Fig. 6. Lines of significant absolute accuracy improvements for different levels of significance ($p = 0.001$, 0.01, or 0.05), for experiments on the CHiME test set. Testing on utterances at individual SNRs ($S = 1200$) or across all six SNRs ($S = 7200$).

9. Conclusion and outlook

This article presented a framework for robust speech recognition that can be applied in high levels of non-stationary background noise and reverberation. In addition to well known techniques such as speaker adaptation and multi-condition training, our system applies convolutive NMF for speech enhancement as well as the principle of Long Short-Term Memory which can efficiently exploit contextual information to enable improved recognition results in challenging conditions. We evaluated three different methods to integrate bidirectional LSTM modeling into speech decoding: first, we designed a novel Tandem front-end employing framewise BLSTM word posterior probabilities as features. Second, we created a CTC-ASR system that uses BLSTM modeling in the back-end and does not need HMMs. Third, we built a multi-stream system that decodes both MFCC features and BLSTM word predictions. All three system variants achieve remarkable performance on the CHiME Challenge task, which consists of recognizing digits and letters in a noisy and reverberated multisource environment. Best accuracy is reached by our fully speaker adapted triple-stream technique which uses non-negative sparse classification in addition to BLSTM and achieves a 4% (absolute) performance gain compared to our original challenge submission (Weninger et al., 2011a). As discussed in more detail by Weninger et al. (2012), this remarkable performance can be attributed to exploitation of complementary methods for noise-robustness in different components of the system (NMF speech enhancement, NSC, and BLSTM context modeling). Another interesting result is that CTC networks can be a promising alternative to HMM-based back-ends. Finally, we point out that the proposed system prevails over previously introduced methods (e.g., Ma et al., 2010). Our system slightly outperforms the best technique proposed in the context of the PASCAL CHiME Challenge 2011 (Delcroix et al., 2011).

Future work will strive for better integration between the system components, especially, of recognition and enhancement. This could be achieved by iterative methods exploiting decoded phonetic information in speech enhancement and vice versa, such as in the study by Raj et al. (2011). Bottleneck features (Grezl et al., 2007) might further increase the performance of our BLSTM-based Tandem front-end. Finally, future studies will include an application of the proposed system for large vocabulary tasks, which is possible if HMMs, BLSTMs, and the NSC component are trained on phonemes rather than on words. This also allows to combine CTC phoneme detectors and HMM-based decoders in order to benefit from the discriminative modeling capabilities of CTC and the advanced acoustic and language modeling technology of HMM frameworks. Furthermore, the relative performance improvement by the system components will be interesting to investigate in speaker-*independent* scenarios, especially since different speech representations (spectral and cepstral) are used in the front-end and the back-end.

Acknowledgements

The research leading to these results has received funding from the Federal Republic of Germany through the German Research Foundation (DFG) under grant nos. SCHU 2508/4-1 and SCHU 2508/2-1. This work was further partially supported 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) program. The authors would like to thank Jort Gemmeke, Antti Hurmalainen, and Tuomas Virtanen for providing source code for non-negative sparse classification.

References

Barker, J.P., Vincent, E., Ma, N., Christensen, H., Green, P.D. The PASCAL CHiME speech separation and recognition challenge. Computer Speech and Language, submitted for publication.

Bengio, Y., Simard, P., Frasconi, P., 1994. Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks 5 (2), 157–166.

Cooke, M., Barker, J., Cunningham, S., Shao, X., 2006. An audio-visual corpus for speech perception and automatic speech recognition. The Journal of the Acoustical Society of America 120 (5), 2421–2424.

Cooke, M., Hershey, J.R., Rennie, S.J., 2010. Monaural speech separation and recognition challenge. Computer Speech and Language 24, 1–15.

Delcroix, M., Kinoshita, K., Nakatani, T., Araki, S., Ogawa, A., Hori, T., Watanabe, S., Fujimoto, M., Yoshioka, T., Oba, T., Kubo, Y., Souden, M., Hahm, S.J., Nakamura, A., 2011. Speech recognition in the presence of highly non-stationary noise based on spatial, spectral and temporal speech/noise modeling combined with dynamic variance adaptation. In: Proc. of Machine Listening in Multisource Environments (CHiME 2011), Satellite Workshop of Interspeech 2011, Florence, Italy, pp. 12–17.

Dietterich, T.G., 1998. Approximate statistical tests for comparing supervised classification learning algorithms. Neural Computation 10, 1895–1923.

Evangelista, G., Marchand, S., Plumbley, M., Vincent, E., 2011. Sound source separation. In: Zölzer, U. (Ed.), DAFX—Digital Audio Effects. , 2nd edition. Wiley.

Fernandez, S., Graves, A., Schmidhuber, J., 2007. An application of recurrent neural networks to discriminative keyword spotting. In: Proc. of ICANN, Porto, Portugal, pp. 220–229.

Gemmeke, J., Virtanen, T., Hurmalainen, A., 2011a. Exemplar-based sparse representations for noise robust automatic speech recognition. IEEE Transactions on Audio, Speech, and Language Processing 19 (7), 2067–2080.

Gemmeke, J.F., Virtanen, T., Hurmalainen, A., 2011b. Exemplar-based speech enhancement and its application to noise-robust automatic speech recognition. In: Proc. of CHiME Workshop, Florence, Italy, pp. 53–57.

Gers, F., Schmidhuber, J., Cummins, F., 2000. Learning to forget: continual prediction with LSTM. Neural Computation 12 (10), 2451–2471.

Graves, A., Fernandez, S., Gomez, F., Schmidhuber, J., 2006. Connectionist temporal classification: labelling unsegmented data with recurrent neural networks. In: Proc. of ICML, Pittsburgh, USA, pp. 369–376.

Graves, A., Fernandez, S., Liwicki, M., Bunke, H., Schmidhuber, J., 2008. Unconstrained online handwriting recognition with recurrent neural networks. Advances in Neural Information Processing Systems 20, 1–8.

Graves, A., Schmidhuber, J., 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Networks 18 (5–6), 602–610.

Grezl, F., Karafiat, M., Stanislav, K., Cernocky, J., 2007. Probabilistic and bottle-neck features for LVCSR of meetings. In: Proc. of ICASSP, pp. 757–760.

Helen, M., Virtanen, T., 2005. Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine. In: Proc. of EUSIPCO, Antalya, Turkey.

Hermansky, H., Ellis, D.P.W., Sharma, S., 2000. Tandem connectionist feature extraction for conventional HMM systems. In: Proc. of ICASSP, Istanbul, Turkey, pp. 1635–1638.

Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J., 2001. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In: Kremer, S.C., Kolen, J.F. (Eds.), A Field Guide to Dynamical Recurrent Neural Networks. IEEE Press, pp. 1–15.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Computation 9 (8), 1735–1780.

Hurmalainen, A., Mahkonen, K., Gemmeke, J.F., Virtanen, T., 2011. Exemplar-based recognition of speech in highly variable noise. In: Proc. of Machine Listening in Multisource Environments (CHiME 2011), Satellite Workshop of Interspeech 2011, Florence, Italy, pp. 1–5.

Jaeger, H., 2001. The echo state approach to analyzing and training recurrent neural networks. Tech. Rep. German National Research Center for Information Technology, Bremen (Tech. Rep. No. 148).

Lang, K.J., Waibel, A.H., Hinton, G.E., 1990. A time-delay neural network architecture for isolated word recognition. Neural Networks 3 (1), 23–43.

Lin, T., Horne, B.G., Tino, P., Giles, C.L., 1996. Learning long-term dependencies in NARX recurrent neural networks. IEEE Transactions on Neural Networks 7 (6), 1329–1338.

Ma, N., Barker, J., Christensen, H., Green, P., 2010. Distant microphone speech recognition in a noisy indoor environment: combining soft missing data and speech fragment decoding. In: Proc. of ISCA Workshop on Statistical and Perceptual Audition (SAPA), Makuhari, Japan.

Raj, B., Singh, R., Virtanen, T., 2011. Phoneme-dependent NMF for speech enhancement in monaural mixtures. In: Proc. of Interspeech, ISCA, Florence, Italy, pp. 1217–1220.

Raj, B., Virtanen, T., Chaudhuri, S., Singh, R., 2010. Non-negative matrix factorization based compensation of music for automatic speech recognition. In: Proc. of Interspeech, Makuhari, Japan, pp. 717–720.

Rennie, S.J., Hershey, J.R., Olsen, P.A., 2008. Efficient model-based speech separation and denoising using non-negative subspace analysis. In: Proc. of ICASSP, Las Vegas, NV, USA, pp. 1833–1836.

Schaefer, A.M., Udluft, S., Zimmermann, H.G., 2008. Learning long-term dependencies with recurrent neural networks. Neurocomputing 71 (13–15), 2481–2488.

Schmidhuber, J., 1992. Learning complex extended sequences using the principle of history compression. Neural Computing 4 (2), 234–242.

Schmidt, M.N., Olsson, R.K., 2006. Single-channel speech separation using sparse non-negative matrix factorization. In: Proc. of Interspeech, Pittsburgh, PA, USA.

Schuller, B., Wöllmer, M., Moosmayr, T., Rigoll, G., 2009. Recognition of noisy speech: a comparative survey of robust model architecture and feature enhancement. Journal on Audio, Speech, and Music Processing (ID 942617).

Schuster, M., Paliwal, K.K., 1997. Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing 45, 2673–2681.

Smaragdis, P., 2004. Discovering auditory objects through non-negativity constraints. In: Proc. of SAPA, Jeju, Korea.

Smaragdis, P., 2007. Convolutive speech bases and their application to supervised speech separation. IEEE Transactions on Audio, Speech and Language Processing 15 (1), 1–14.

Wang, W., Cichocki, A., Chambers, J.A., 2009 July. A multiplicative algorithm for convolutive non-negative matrix factorization based on squared Euclidean distance. IEEE Transactions on Signal Processing 57 (7), 2858–2864.

Weninger, F., Geiger, J., Wöllmer, M., Schuller, B., Rigoll, G., 2011a. The Munich 2011 CHiME Challenge Contribution: NMF-BLSTM Speech Enhancement and Recognition for Reverberated Multisource Environments. In: Proc. of Machine Listening in Multisource Environments (CHiME 2011), Satellite Workshop of Interspeech 2011, Florence, Italy, pp. 24–29.

Weninger, F., Lehmann, A., Schuller, B., 2011b. openBliSSART: design and evaluation of a research toolkit for blind source separation in audio recognition tasks. In: Proc. of ICASSP, Prague, Czech Republic, pp. 1625–1628.

Weninger, F., Wöllmer, M., Geiger, J., Schuller, B., Gemmeke, J., Hurmalainen, A., Virtanen, T., Rigoll, G., 2012. Non-Negative Matrix Factorization for Highly Noise-Robust ASR: to Enhance or to Recognize? In: Proc. of ICASSP, Kyoto, Japan, pp. 4681–4684.

Wilson, K.W., Raj, B., Smaragdis, P., Divakaran, A., 2008. Speech denoising using nonnegative matrix factorization with priors. In: Proc. of ICASSP, Las Vegas, NV, USA, pp. 4029–4032.

Wöllmer, M., Blaschke, C., Schindl, T., Schuller, B., Färber, B., Mayer, S., Trefflich, B., 2011a. On-line driver distraction detection using long short-term memory. IEEE Transactions on Intelligent Transportation Systems 12 (2), 574–582.

Wöllmer, M., Eyben, F., Graves, A., Schuller, B., Rigoll, G., 2010a. Bidirectional LSTM networks for context-sensitive keyword detection in a cognitive virtual agent framework. Cognitive Computation 2 (3), 180–190.

Wöllmer, M., Eyben, F., Schuller, B., Rigoll, G., 2011b. A multi-stream ASR framework for BLSTM modeling of conversational speech. In: Proc. of ICASSP, Prague, Czech Republic, pp. 4860–4863.

Wöllmer, M., Schuller, B., 2011. Enhancing spontaneous speech recognition with BLSTM features. In: Proc. of NOLISP, Las Palmas de Gran Canaria, Spain, pp. 17–24.

Wöllmer, M., Schuller, B., Eyben, F., Rigoll, G., 2010b. Combining long short-term memory and dynamic Bayesian networks for incremental emotion-sensitive artificial listening. IEEE Journal of Selected Topics in Signal Processing 4 (5), 867–881.

Wöllmer, M., Schuller, B., Rigoll, G., 2011. Feature frame stacking in RNN-based Tandem ASR systems—learned vs. predefined context. In: Proc. of Interspeech, Florence, Italy, pp. 1233–1236.