Feature Enhancement by Deep LSTM Networks for ASR in Reverberant Multisource Environments

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

[a] *Institute for Human-Machine Communication, Technische Universität München, 80290 Munich, Germany*
[b] *BMW Group, 80788 Munich, Germany*
[c] *Imperial College London, Department of Computing, London, U.K.*

Abstract

This article investigates speech feature enhancement based on deep bidirectional recurrent neural networks. The Long Short-Term Memory (LSTM) architecture is used to exploit a self-learnt amount of temporal context in learning the correspondences of noisy and reverberant with undistorted speech features. The resulting networks are applied to feature enhancement in the context of the 2013 2nd Computational Hearing in Multisource Environments (CHiME) Challenge Track 2 task, which consists of the Wall Street Journal (WSJ-0) corpus distorted by highly non-stationary, convolutive noise. In extensive test runs, different feature front-ends, network training targets, and network topologies are evaluated in terms of frame-wise regression error and speech recognition performance. Furthermore, we consider gradually refined speech recognition back-ends from baseline 'out-of-the-box' clean models to discriminatively trained multi-condition models adapted to the enhanced features. In the result, deep bidirectional LSTM networks processing log Mel filterbank outputs deliver best results with clean models, reaching down to 42 % word error rate (WER) at signal-to-noise ratios ranging from -6 to 9 dB (multi-condition CHiME Challenge baseline: 55 % WER). Discriminative training of the back-end using LSTM enhanced features is shown to further decrease WER to 22 %. To our knowledge, this is the best result reported for the 2nd CHiME Challenge WSJ-0 task yet.

Keywords: automatic speech recognition, feature enhancement, deep neural networks, Long Short-Term Memory

1. Introduction

Decoding of large vocabulary speech in unfavorable acoustic conditions, especially in hands-free scenarios involving interfering noise sources and room reverberation, is still a major challenge for today's automatic speech recognition (ASR) systems despite decades of research on this topic. Robustness of ASR systems can be addressed at different stages of the recognition process (Schuller et al., 2009), and successful systems usually employ a combination of them (Barker et al., 2013). Popular techniques comprise front-end speech enhancement, such as by microphone array processing (Maas et al., 2011; Nesta et al., 2013) or monaural speech de-noising

*corresponding author: weninger@tum.de, Tel.: +49 89 289-28562, Fax.: +49 89 289-28535

techniques (Rennie et al., 2008; Raj et al., 2010), as well as improvements in the back-end by model adaptation (Gales and Wang, 2011) or improved ASR architectures taking into account additional sources of information, such as neural networks (Hinton et al., 2012; Seltzer et al., 2013; Geiger et al., 2013). 'In between' one can also address noise-robust features – a popular expert crafted feature extraction scheme is RASTA-PLP (Hermansky et al., 1992) – or feature enhancement, defining a mapping from noisy to noise free speech features. An example for a data-based, non-parametric technique for feature enhancement is histogram equalization (de la Torre et al., 2005; Wöllmer et al., 2011a).

Furthermore, feature enhancement by recurrent neural networks has been considered (Parveen and Green, 2004; Maas et al., 2013). In particular, bidirectional Long Short-Term Memory (BLSTM) recurrent neural networks (RNNs) have been employed by Wöllmer et al. (2013) for feature enhancement in highly non-stationary noise, by mapping noisy cepstral features to clean speech cepstral features, and have been shown to outperform traditional RNNs on this task. In (Weninger et al., 2013), we have successfully applied the BLSTM methodology to both ASR tasks (small and medium vocabulary) of the 2013 2nd CHiME Speech Separation and Recognition Challenge (Vincent et al., 2013), which features highly non-stationary convolutive noise recorded from a real home environment over a period of several weeks. There, our BLSTM approach outperformed a similar approach using conventional RNNs on a small vocabulary task (Maas et al., 2013). In this article we proceed to a larger scale evaluation of BLSTM-RNNs and other types of neural networks – including feedforward neural networks – in a medium vocabulary task.

With respect to our earlier study (Weninger et al., 2013), this article presents several improvements of network topology and training, resulting in further performance gains. Furthermore, a goal of our present study is to clarify which parts of the performance gain can be attributed to refined ASR back-ends and which to better feature enhancement. In particular, we consider feature mappings from noisy and reverberated to close-talk features with deep network topologies, as well as feature enhancement in the logarithmic Mel frequency domain instead of the cepstral domain. We also investigate whether measures of the network regression performance are correlated to ASR performance, which involves much more complicated likelihood functions than typically used in network training. We also take into account the effect of using multi-condition training with reverberated and noisy speech, feature transformations, and discriminative back-end training separately. All these points have not been addressed in our earlier work (Weninger et al., 2013).

In the following, we will first outline our feature enhancement methodology before describing the experimental setup including a brief outline of the CHiME Challenge data and presenting the results.

2. Feature Enhancement

2.1. Deep LSTM Recurrent Neural Networks

In this article, we use deep LSTM recurrent neural networks (RNNs) for speech feature enhancement. By that, we combine several ideas that have been successfully applied to speech recognition tasks: using multiple hidden layers for increasingly higher level representations of the input features (Hinton et al., 2012; Graves et al., 2013), exploiting temporal context by using recurrent neural networks with an internal state that is preserved over time by using the LSTM architecture (Gers et al., 2000; Graves, 2008), and supervised learning of non-linear mappings from noisy and reverberant to clean features (Maas et al., 2013; Weninger et al., 2013).

Let us denote the noisy input features in time frame t by $\mathbf{x}_t$ and the corresponding clean features by $\mathbf{s}_t$. We use deep LSTM-RNNs with N layers to generate an estimate of the clean speech features $\hat{\mathbf{s}}_t$ by the following iterative procedure:

$$\begin{aligned} \mathbf{h}_t^{(0)} &:= \mathbf{x}_t, && (1)\\ \mathbf{h}_t^{(n)} &:= \mathcal{L}_t^{(n)}\left(\mathbf{h}_t^{(n-1)}, \mathbf{h}_{t-1}^{(n)}\right), && (2)\\ \hat{\mathbf{s}}_t &:= \mathbf{W}^{(N),(N+1)}\mathbf{h}_t^{(N)} + \mathbf{b}^{(N+1)} && (3) \end{aligned}$$

for $n = 1, \ldots, N$ and $t = 1, \ldots, T$, where T is the number of frames in the utterance. $\mathbf{h}_t^{(n)}$ denotes the hidden feature representation of time frame t at level n. In the above and in the ongoing, $\mathbf{W}^{(n),(n+1)}$ denotes the feed-forward connection weights from layer n to the next layer ($n = 0$: input layer, $n = N$: output layer), while $\mathbf{W}^{(n),(n)}$, $n > 0$, contains the 'self-loop' weights implementing the recurrent structure; $\mathbf{b}$ denotes bias vectors.

From Eqn. 1, 2, and 3, it is obvious that the enhanced speech frame $\hat{\mathbf{s}}_t$ depends on the previous inputs and also the previous enhanced frames $\hat{\mathbf{s}}_{t-1}, \hat{\mathbf{s}}_{t-2}, \ldots$. This way, recurrent neural networks are able to model speech feature dynamics both in the input and output, rather than doing frame by frame enhancement. In contrast to other studies using recurrent neural networks for speech de-noising (Maas et al., 2013), our networks employ the LSTM activation function $\mathcal{L}_t^{(n)}$ instead of the typically used simple sigmoid-like functions. The crucial point is to augment the activation function of each cell with a state variable c_t that is preserved by means of a recurrent connection with weight 1. This enables the network to store inputs over longer periods of time; for example, noise frames without speech can be valuable of enhancing noisy speech frames in the future. It also resolves the 'vanishing gradient problem' where the influence of inputs on the output would decrease exponentially over time in conventional RNNs, making them difficult to train using gradient descent (Bengio et al., 1994). The hidden layer activations correspond to the states of the cells scaled by the activations of the 'output gates',

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

where $\otimes$ denotes element-wise multiplication and tanh is applied element-wise. For $\mathbf{c}_t^{(n)}$, the following definition holds:

$$\mathbf{c}_t^{(n)} = \mathbf{f}_t^{(n)} \otimes \mathbf{c}_{t-1}^{(n)} + \mathbf{i}_t^{(n)} \otimes \tanh\left(\mathbf{W}^{(n-1),(n)}\mathbf{h}_t^{(n-1)} + \mathbf{W}^{(n),(n)}\mathbf{h}_{t-1}^{(n)} + \mathbf{b}_c^{(n)}\right). \quad (4)$$

There, $\mathbf{f}_t^{(n)}$ is the activation of the 'forget gate' that can scale the state variable and probably reset it to zero. Furthermore, $\mathbf{i}_t^{(n)}$ is the activation of the input gate that regulates the 'influx' from the feedforward and recurrent connections. Similarly to (4), the activations of the output gates o_t, input gates i_t and forget gates f_t are non-linear functions of weighted combinations of $\mathbf{h}_t^{(n-1)}$ (feedforward connections) and $\mathbf{h}_{t-1}^{(n)}$ (recurrent connections). In particular, instead of multiplying the hidden layer activations from the previous time step with a static weight as in a traditional RNN, the network 'learns when to forget' (Gers et al., 2000). Details can be found in (Graves, 2008; Graves et al., 2013). It has been shown in the context of speech recognition that using the LSTM activation function provides a self-learnt amount of temporal context to the network, which seems to be superior to relying on a manually defined amount of 'stacked' input feature frames (Wöllmer et al., 2011b).

The parameters $\mathbf{W}$ and $\mathbf{b}$ are learned by backpropagation through time from noisy and clean training data (cf. Section 3.3). The sum of the squared deviations between $\hat{\mathbf{s}}_t$ and the original clean speech $\mathbf{s}_t$ (sum of squared errors, SSE) is used as error function,

$$d = \sum_{t,f} (s_{t,f} - \hat{s}_{t,f})^2. \tag{5}$$

In case that $\hat{\mathbf{s}}_t$ and $\mathbf{s}_t$ are log spectra, this function is related to the log spectral distance (Gray and Markel, 1976).

2.2. Bidirectional Extension

So far, the automaton structure given by (1, 2, 3) can exploit acoustic context from previous frames. For automatic speech recognition, where whole utterances are decoded, future context can be used as well. This results in the concept of bidirectional networks. Each layer of a bidirectional network consists of two independent layers, one of which applies (2, 3) in the order $t = 1, \ldots, T$ as above (forward layer) and the other in the reverse order, i.e., replacing $t - 1$ by $t + 1$ for the recurrent connections and iterating over $t = T, \ldots, 1$ (backward layer).

For each time step t, the activations of the n-th forward ($\rightarrow$) and backward ($\leftarrow$) layer are collected in a single vector

$$\mathbf{h}_t^{(n)} = \left[\overrightarrow{\mathbf{h}}_t^{(n)}; \overleftarrow{\mathbf{h}}_t^{(n)} \right]. \tag{6}$$

Both the forward and backward layers in the next level $(n + 1)$ 'see' this entire vector as input. Thus, conceptually, in a deep BLSTM network one processes the sequence forward, then backward, collects the activations and uses them as input for a forward and backward pass on the sequence on the next level, etc. Alternatively to (6), one can consider 'subsampling layers' (Graves, 2008) performing the operation

$$\mathbf{h}_t^{(n)} = \tanh\left(\mathbf{W}^{\mathrm{sub},(n)} \left[\overrightarrow{\mathbf{h}}_t^{(n)}; \overleftarrow{\mathbf{h}}_t^{(n)} \right] \right), \tag{7}$$

with trainable low-rank weight matrices $\mathbf{W}^{\mathrm{sub},(n)}$, for $n = 1, \ldots, N - 1$. We found this very useful for information reduction between the layers, reducing training time without decreasing performance, in contrast to simply using less hidden units.

3. Experimental Setup

3.1. The 2nd CHiME Challenge Corpus

In this article, we perform evaluations on the medium vocabulary (5 k) task of the 2013 2nd CHiME Challenge (Vincent et al., 2013). It consists of reverberated and noisy utterances corresponding to artificially degraded versions of the the speaker independent development and evaluation test sets of the Wall Street Journal corpus of read speech (WSJ-0). It is split into disjoint sets with 84, 10, and 8 training, development, and test speakers, each comprising different prompts (si_tr_s, si_dt_05 and si_et_05). The monophonic original utterances have been convolved with stereophonic room impulse responses measured in a domestic environment, and overlaid with realistic, stereophonic noise recorded in the same environment at signal-to-noise ratios (SNRs) from -6 to 9 dB, in steps of 3 dB. Instead of artificially scaling speech and noise to

resemble various SNRs, segments matching a specific SNR are selected from the noise recordings. Thus, noise types differ among SNRs and range from household appliances to music and to interfering speakers. The full set of utterances is used at all SNRs in each of the development and test sets. Thus, there are $6 \times 409 = 2\,454$ development, and $6 \times 330 = 1\,980$ test utterances. A noisy training set is provided in addition, which comprises randomly selected, disjoint subsets of WSJ-0 training utterances at each SNR. Thus, the number of training utterances in the noisy training set is the same as in the original WSJ-0 corpus (7 138). The training and development sets are also provided in a noise-free, but reverberated version to allow for evaluation of denoising algorithms. The total length of the training, development, and test set is 14.5, 4.5, and 4 hours. While the Challenge data is stereophonic, in our study we only consider simple beamforming and subsequent monaural processing (cf. below). The 2nd CHiME Challenge corpus is made publicly available for WSJ-0 licensees[1].

3.2. Feature Enhancement Front-End

Our contribution to the 2nd CHiME Challenge itself (Weninger et al., 2013), and a related contribution using standard RNNs (Maas et al., 2013) considered only Mel frequency cepstral coefficients (MFCCs) as input and output of the feature enhancement networks. Using MFCCs is mainly an ad-hoc solution motivated by their use in the speech recognition back-end; in particular, HMMs with diagonal covariance Gaussian mixtures.

However, recent studies on deep neural network based speech recognition (Hinton et al., 2012; Graves et al., 2013) directly use logarithmic Mel filterbank outputs (Log-FB). The rationale behind using Log-FB is to let the network derive a suited higher-level feature extraction strategy by itself. Furthermore, we also consider Log-FB as training targets. Since Log-FB are correlated with each other, this resembles multi-task regularization of the network and is thus expected to help generalization. 26 Log-FB covering the frequency range from 20–8 000 Hz are used, as is often done in ASR. We add delta coefficients both to the input and output; using them as targets is similar in spirit to the proposal by Seltzer and Droppo (2013) to use multi-frame information as training targets in neural network based speech recognition, which again serves to improve generalization. As additional feature in input and output, we use root-mean-square (RMS) energy with deltas. For the MFCC features, we also add acceleration coefficients (second order deltas), and we perform cepstral mean normalization (CMN) to (partially) compensate channel effects. Thus, in the MFCC case, the network input and output exactly correspond to the ASR front-end used in the HTK CHiME baseline (Vincent et al., 2013). In the Log-FB case, the outputs can be converted to MFCCs by simply applying a Discrete Cosine Transformation (DCT) (Young et al., 2006), cf. below. Log-FB features are investigated with and without log spectral subtraction, which is the Log-FB domain equivalent of CMN (Gelbart and Morgan, 2001). For transparency, feature extraction is done using HTK, using the `MFCC_E_D_A_Z`, `FBANK_E_D` and `FBANK_E_D_Z` types of features with the default parameters (Young et al., 2006).

Prior to feature extraction, the stereophonic signals are down-mixed to monophonic audio by averaging channels, corresponding to simple delay-and-sum beam-forming. This is useful for the CHiME Challenge track 2 data where the speaker is positioned at a frontal position with respect to the microphone, and is hence exploited in the baseline system by Vincent et al. (2013).

All features are globally mean and variance normalized. To this end, we compute the global means and variances of the noise-free and the noisy training set feature vectors and perform mean

[1] http://spandh.dcs.shef.ac.uk/chime_challenge/ – last retrieved January 2014

and variance normalization of the network training targets and the network inputs accordingly. This normalization was found to be very important for performance; in particular, it ensures that features with large variance due to noise do not 'mask' important information in features with lower variance such as delta coefficients.

3.3. Network Training

Feature enhancement BLSTM networks are trained on the task to map the features of the noisy training set of the above mentioned corpus to a noise-free training set. In a first set of experiments, we consider de-noising only, i.e., learning mappings of noisy to clean features within the same acoustic environment. As a consequence, the output features will still be reverberated, and they will be used for decoding with a model adapted to the reverberated training data. This corresponds to our contribution to the 2nd CHiME Challenge (Weninger et al., 2013). In this article, we additionally consider learning mappings from noisy and reverberated to 'fully clean' (noise-free, close-talk microphone speech) features, i.e., the network also learns feature-space de-reverberation. There, we also consider deep learning with pre-training, where the first layers are trained to de-noising and subsequent layer(s) are trained to perform de-reverberation. While the CHiME WSJ-0 corpus also contains noise context for each utterance, we use only the 'isolated' utterances, i.e., the end-pointed speech segments.

We train the networks through on-line gradient descent with a learning rate of 10^{-5} and a momentum of 0.9. Prior to training, all weights are randomly initialized with Gaussian random numbers (mean 0, standard deviation 0.1). The on-line gradient descent algorithm applies weight changes after processing each utterance, using a random order of utterances in each training epoch to alleviate overfitting. Using on-line learning was found to drastically speed up convergence and increase generalization compared to batch learning. Zero mean Gaussian noise with standard deviation 0.1 is added to the input activations in the training phase, and an early stopping strategy is used in order to further help generalization. The latter is implemented as follows: We evaluate the overall SSE (5) on the development set after every fifth epoch. We abort training as soon as no improvement of the SSE on the development set has been observed during 30 epochs. The network that achieved the best SSE on the development set (across all six SNRs) is chosen as the final network.

Most of the applied BLSTM networks have three hidden layers consisting of $2M$, 128, and $2M$ LSTM cells as described above, where M is the input and output feature dimension (39 for MFCC, 54 for Log-FB). Each memory block contains one memory cell. This topology was empirically determined on a similar speech feature enhancement task (Wöllmer et al., 2013). In case that noisy features are mapped to clean features, we also consider networks with four hidden layers incorporating $2M$, 128, $2M$, and $2M$ LSTM cells. The rationale behind this is that mapping to clean features is a more complex task than just removing noise, which also involves de-reverberation. Besides training the four hidden layers without additional constraints, we also aim at enforcing structure by pre-training of the first three hidden layers. In particular, we add a fourth hidden layer to the three-layer network which has been trained to map noisy and reverberated to noise-free reverberated features, and then run additional training epochs using the same inputs, but clean features as targets. For the sake of consistency, the training parameters are set based on our previous experience with RNN-based enhancement of conversational speech in noise (Wöllmer et al., 2013). Our LSTM training software is publicly available[2].

[2]https://sourceforge.net/p/currennt – last retrieved January 2014

3.4. Baseline Networks

To verify the effectiveness of BLSTM networks for feature enhancement, we also consider simpler network architectures: bidirectional RNNs (BRNNs) and feedforward neural networks (FNN). Bidirectional RNNs are obtained by replacing $\mathcal{L}_t^{(n)}$ in Eqn. 2 by the hyperbolic tangent function $\tanh(\mathbf{W}^{(n-1),(n)}\mathbf{h}_t^{(n-1)} + \mathbf{W}^{(n),(n)}\mathbf{h}_{t-1}^{(n)})$, and in the case of FNN, $\tanh(\mathbf{W}^{(n-1),(n)}\mathbf{h}_t^{(n-1)})$. Since the latter does not take into account context which is vital for speech processing tasks, in the case of FNN we replace $\mathbf{x}_t$ by $[\mathbf{x}_{t-\mathcal{T}}; \cdots ; \mathbf{x}_{t+\mathcal{T}}]$ in Eqn. 1 where $\mathcal{T}$ is a fixed parameter representing the context length, i.e., features are stacked into a column 'super'vector. We use $\mathcal{T} = 4$, i.e., nine frame context windows. In analogy to RNNs, FNNs are trained on the task to provide a clean speech estimate $\hat{\mathbf{s}}_t$ of $\mathbf{x}_t$, which is the center frame of the context window.

As FNN topologies, we investigate both 'symmetric' hidden layers (3×256 units) as well as a structure that reduces information layer by layer (486, 256, and 108 hidden units), matching the size of the first hiddden layer to the input layer and the size of the third layer to two times the size of the output layer. BRNNs have the same size as BLSTM-RNNs (108, 128, 108 hidden units). Since BLSTM-RNNs have many more parameters than FNNs or BRNNs of the same hidden layer size, we also investigate a smaller BLSTM net (81, 96, and 81 hidden units) whose number of parameters compares to the simpler architectures. For a fair comparison, both BRNNs and FNNs were trained using the same stochastic gradient descent algorithm as BLSTM-RNNs, using random initialization. We tuned the learning rate for FNNs and BRNNs on the development set and found that best performance with FNN was obtained with 10^{-7}, as opposed to 10^{-5} for the BLSTM-RNNs, requiring more training epochs until convergence. BRNNs required setting the learning rate as low as 10^{-8} in order for training to converge.

3.5. Obtaining ASR Features

As detailed above, the first step of ASR feature extraction is presenting the frame-wise noisy features (MFCC or Log-FB) $\mathbf{x}$ to the trained network and computing the denoised features $\hat{\mathbf{s}}$ as the output activations. In principle, cepstral mean normalized MFCC features with deltas output by a network can be used 'as is' in the speech recognizer. However, due to the normalization of the training targets, $\hat{\mathbf{s}}$ will be (approximately) mean and variance normalized, which does not match the features used to train the baseline models. Thus, to be able to use the enhanced features in a 'plug-and-play' fashion, i.e., without any recognizer modification, the global mean and variance normalization is reverted after obtaining the enhanced MFCC features, to ensure compatibility with the means and variances of the trained recognition models. More specifically, each enhanced feature vector is multiplied element-wise with the corresponding variances of the noise-free training set, and the mean feature vector of the noise-free training set is added. For the Log-FB features, deltas output by the network are thrown away, the MVN is reverted as above, and cepstral mean normalized MFCC features with delta and acceleration coefficients are computed from the Log-FB features output by the network.

4. Speech Recognition Back-Ends

In the following, we now describe the speech recognition back-ends we use for evaluating our feature enhancement procedure.

4.1. Baseline models

We evaluate the performance of the enhanced features using the baseline models provided by the Challenge organizers, as well as re-trained models using enhanced features. The baseline is implemented using HTK (Young et al., 2006) based on the WSJ-0 'recipe' by Vertanen (2006). From these models, a 'reverberated' baseline model is generated by an Expectation Maximization (EM) Maximum Likelihood (ML) algorithm on the reverberated training set. Four EM-ML iterations are used. The 'noisy' baseline model is created by four additional EM-ML iterations using the training set with convolutive noise. From these 'noisy' models, we derive 're-trained' models simply by repeating the multi-condition training step using features that have been processed by our enhancement networks. This is done to investigate to which extent distortions by enhancement can be compensated by model re-training. Furthermore, it is expected that feature de-noising and de-reverberation results in lower feature variance, requiring model adaptation. In contrast, using the baseline models without modification serves to estimate the 'compatibility' of enhanced features with their clean counterparts used to train the ASR models. From an application point of view, it corresponds to a 'plug-and-play' configuration – in other words, a scenario where the recognizer back-end is a 'black box' and only the feature extraction front-end is known.

4.2. Discriminatively trained models with feature transformations

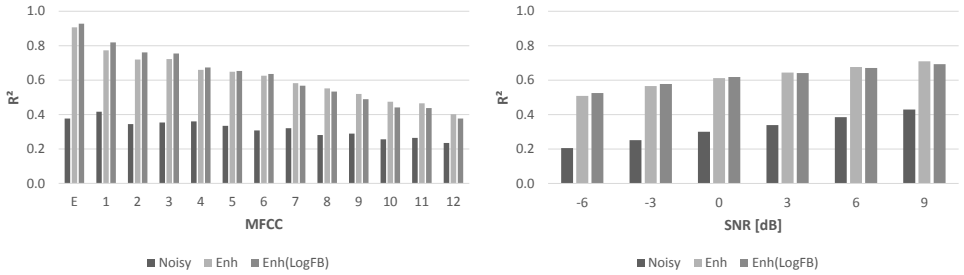
The training procedure used to generate the CHiME baseline models does not use many state-of-the-art ASR techniques, such as feature transformations and discriminative training. Thus, it is of crucial interest to investigate whether the performance of state-of-the-art ASR, such as the back-end used by Tachioka et al. (2013) for their (winning) contribution to the CHiME Challenge track 2, can also be improved by our feature enhancement technique. This system is implemented with the Kaldi speech recognition toolkit (Povey et al., 2011). The 'recipe' for training the back-end is publicly available[3]. Discriminative training is performed using boosted Maximum Mutual Information (MMI) as proposed by Povey et al. (2008). The MMI principle aims at maximising the posterior probabilities of the correct utterances, given the trained models. Boosted MMI (bMMI) introduces a weight, strenghtening the influence of hypotheses with a higher error. For bMMI, the objective function is

$$\mathcal{F}_{bMMI}(\lambda) = \sum_{r=1}^{R} \log \frac{p_\lambda(\mathcal{X}_r|\mathcal{M}_{s_r})^\kappa p_L(s_r)}{\sum_s p_\lambda(\mathcal{X}_r|\mathcal{M}_s)^\kappa p_L(s) e^{-bA(s,s_r)}}, \tag{8}$$

where $r = 1 \ldots R$ are the training utterances and $\mathcal{X}_r$ 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 phoneme accuracy of sentence s given the reference s_r.

Furthermore, techniques for feature transformation are employed. Feature transformation can improve the class separation and address the speaker variability in the training data. Linear discriminant analysis (LDA) is applied on stacked MFCCs and reduces the resulting high-dimensional feature vector to a smaller dimension. The necessary classes are obtained by aligning the tri-phone HMM states. By that, robustness to noise and reverberation can be addressed,

[3] http://spandh.dcs.shef.ac.uk/chime_challenge/WSJ0public/CHiME2012-WSJ0-Kaldi_0.03.tar.gz – last retrieved January 2014



(a) R^2 by coefficient across SNRs

(b) R^2 by SNR across MFCC 1–12 and energy

Figure 1: Evaluation of BLSTM feature de-noising: R^2 of enhanced and noisy MFCC features (1–12) and RMS energy (E) with noise-free MFCC features on the CHiME 2013 track 2 development set. De-Noised (MFCC): BLSTM output = enhanced MFCC; De-Noised (Log-FB): MFCCs generated from enhanced Log-FB features output by BLSTM.

assuming that these distortions occur in regular temporal patterns which can be expressed as feature dimensions not related to phonetic information, and hence be discarded. 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. Several transformations for decreasing the correlations between features have been proposed. We use maximum likelihood linear transform (MLLT), as described in (Saon et al., 2000). Additionally, large variations among speakers degrade the performance of the acoustic models. To address this problem, speaker adaptive training (SAT) (Anastasakos et al., 1997) is applied: Before the ML training procedure, feature-space maximum likelihood linear regression (f-MLLR), which is the same as constrained MLLR (Gales, 1998), is applied to estimate a speaker-dependent transform for each speaker. The estimated transform is then used during model re-estimation in training. During decoding, speaker identities are assumed to be known. First, a tight-beam decoding is performed on all test utterances of a single speaker to obtain a first pass transcription, which is used to re-estimate the SAT transform, before doing a final decoding.

Parameterization and training of acoustic models follows (Tachioka et al., 2013) 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. Another set of ML training iterations is then performed after applying the described feature transformations, using the noisy training data. Here, first, 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 tri-phone 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 bMMI with a boosting factor of $b = 0.1$.

Table 1: *ASR evaluation of BLSTM feature de-noising (MFCC, Log-FB). Word error rates (% WER) on CHiME 2013 track 2 development set using baseline HMM-MFCC recognizer trained by EM-ML on reverberated noise-free training set (si_tr_s). SSub: (log) spectral subtraction.*

WER [%]	**SNR** [dB]						**Mean**
FE Domain	-6	-3	0	3	6	9	
—	86.25	82.79	76.08	71.35	63.04	55.87	72.56
MFCC	69.57	62.23	53.83	48.51	43.18	37.15	52.41
Log-FB	**62.59**	55.84	**47.80**	43.82	**38.25**	**34.68**	**47.16**
Log-FB + SSub	63.57	**55.38**	48.02	**43.76**	38.75	35.48	47.49

5. Results and Discussion

5.1. Regression Performance

Before turning to task-based ASR evaluation, let us first investigate the feature enhancement performance in terms of regression error. We compute the determination coefficient R^2 (squared Pearson correlation coefficient) of the noise free features with (i) the unprocessed noisy MFCC features, (ii) the MFCC features output by the MFCC enhancement network, and (iii) the MFCC features computed from the output of the Log-FB enhancement network. We did not consider the correlation of Log-FB outputs with Log-FB 'ground truth' because we are mostly interested in comparing the two types of enhancement in the context of ASR using MFCC features. From the results displayed in Figure 1, it can be seen that BLSTM feature enhancement always improves over the noisy baseline. Furthermore, lower order MFCCs are predicted with slightly higher precision by the Log-FB enhancement network while the MFCC enhancement network is better at predicting higher order MFCCs. This is somewhat expected since the error function averages over frequency bands in the first case and over MFCCs in the second case – thus low quefrencies are given more weight in the error calculation for the Log-FB enhancement network. However, lower order MFCCs seem to be easier to enhance than higher order MFCCs regardless of the actual type of features used in the network. Especially for high MFCCs at higher SNRs, we observe a drop in performance by Log-FB instead of direct MFCC enhancement (e.g., MFCC 12 at 9 dB SNR, Log-FB: $R^2 = .46$, MFCC: $R^2 = .51$). Conversely, e.g., enhancement of the MFCC 1 at -6 dB SNR works considerably better when using Log-FB as features in the enhancement network (Log-FB: $R^2 = .73$, MFCC $R^2 = .67$). Overall, these results are quite promising since it is expected that higher performance on the lower order MFCCs achieved by Log-FB domain enhancement would result in ASR performance gains. This hypothesis will be verified below.

5.2. ASR Performance

We begin our ASR evaluation of BLSTM enhanced features by considering BLSTM de-noising, i.e., learning mappings between noisy and noise-free features within the same acoustic environment. As acoustic models, we use the 'reverberated' CHiME baseline models (Vincent et al., 2013). Evaluation is done on the CHiME 2013 track 2 development set (test set results will be given below for selected systems). The resulting word error rates (WER) are shown in Table 1. It can be seen that by enhancing the MFCCs directly, one obtains an improvement of 20 % absolute (28 % relative) in terms of WER. Using Log-FB outputs as net input and target, WER is further decreased by 5 % absolute (10 % relative), reaching 47.16 % average WER across the

Table 2: *ASR evaluation of alternative network topologies in Log-FB domain enhancement. FNNs using nine frames of input context to enhance center frame. # Wts: number of weights in network. Word error rates (% WER) on CHiME 2013 track 2 development set using baseline HMM-MFCC recognizer trained by EM-ML on reverberated noise-free training set (si_tr_s).*

%WER			**SNR** [dB]						**Mean**
Network	Layers	# Wts	-6	-3	0	3	6	9	
BLSTM	81-96-81	305 k	63.36	55.22	48.71	44.28	38.05	34.39	**47.34**
BRNN	108-128-108	159 k	76.44	69.68	60.90	58.49	52.32	47.62	60.91
FNN	256-256-256	270 k	74.78	68.27	59.79	54.91	49.54	43.60	58.48
FNN	384-384-384	503 k	76.26	69.68	60.96	56.99	50.07	45.79	59.96
FNN	486-256-108	395 k	76.37	68.86	60.88	55.93	49.82	46.87	59.79

six SNRs. Using log spectral subtraction (SSub) on the filterbank outputs cannot further improve results. Thus, it seems that the mapping from noisy to clean features can best be learnt in the 'raw' log spectral domain.

Regarding the performance of BLSTM in comparison to simpler network architectures, i.e., bidirectional RNN and feedforward networks with input frame stacking, we find that BLSTM significantly outperforms both BRNN and FNN (Table 2). This corroborates our earlier results with neural network based feature enhancement (Wöllmer et al., 2013). Comparing the number of parameters of the networks, it can be seen that the superiority of BLSTM is not simply due to increasing model complexity in terms of weights. In particular, the BLSTM network with 81, 96, and 81 units per layer performs almost equally to the larger network considered above, while FNNs with the same number of parameters perform significantly worse (58.48 % WER with the FNN with 3×256 units having 270 k weights, vs. 47.34 % with the BLSTM having 305 k weights). Further increasing the FNN size to 384 units per layer, or adjusting the hidden layer size to the size of the adjacent input and output layers (486-256-108 topology) does not improve performance. Generally, the fact that larger networks do not improve performance could be attributed to the limited amount of training data in the CHiME Challenge. Furthermore, we observe that BRNNs perform slightly worse than FNNs with stacked inputs, pointing at the difficulty of training conventional RNNs through standard gradient descent. The fact that BLSTM modeling outperforms feature frame stacking is in accordance with the results reported by Wöllmer et al. (2011b) for neural network based phoneme recognition.

Next, in Table 3, we consider the performance of BLSTM de-noised and de-reverberated features in the close-talk recognizer. The baseline WER of this recognizer applied to the CHiME development set is very high (89.43 % on average and 82.07 % even at 9 dB SNR). However, a drastic drop in WER occurs when applying feature enhancement in the MFCC domain (50.79 % WER, using the same network topology as above). Again, when using Log-FB outputs as enhancement domain, we obtain further improvement down to 46.97 % WER (using the same network topology as for de-noising). When simply using a fourth layer, results are much worse (51.52 % WER), pointing at overfitting due to the increased number of parameters. When we use the above-mentioned deep training technique for mapping to clean features, we obtain 47.76 % WER, which is, however, below the result with simple training of a three-layer network. Switching to the zero mean log spectral domain, direct training of three- or four-layer networks does not reach the performance obtained with deep training. In the result, the lowest average WER we

Table 3: *ASR evaluation of BLSTM feature de-noising and de-reverberation. Word error rates (% WER) on CHiME 2013 track 2 development set using baseline HMM-MFCC recognizer trained by EM-ML on close-talk microphone WSJ-0 training set (si_tr_s). SSub: (log) spectral subtraction. 3+1 layers: 4 layer network with pre-training of 3 layers (see text).*

WER [%]	Layers	**SNR** [dB]						**Mean**
FE Domain		-6	-3	0	3	6	9	
Baseline (no enhancement)								
—	—	94.08	92.97	91.51	89.92	86.03	82.07	89.43
With BLSTM feature enhancement								
MFCC	3	70.10	61.16	52.34	47.66	38.97	34.51	50.79
Log-FB	3	65.34	57.58	48.18	41.78	36.5	32.44	46.97
Log-FB	4	69.97	62.49	52.71	47.28	41.23	35.41	51.52
Log-FB	3+1	66.53	59.27	48.6	43.08	36.49	32.57	47.76
Log-FB + SSub	3	65.92	58.53	48.34	42.19	36.65	31.36	47.17
Log-FB + SSub	4	65.48	57.47	47.62	42.49	35.97	31.70	46.79
Log-FB + SSub	3+1	**65.07**	**56.85**	**47.03**	**41.51**	**35.56**	**30.90**	**46.15**

attain with the unmodified close-talk recognizer is at 46.15 %, which is a 48 % relative reduction with respect to using unenhanced features. In comparison, a four-layer network achieves 46.79 % average WER and a three-layer network 47.17 % WER. These rates are significantly worse (true average WER differences $\geq .45$ and $\geq .65$ with 95 % confidence, according to a one-tailed t-test, treating WER per SNR as independent observations). Comparing the results to those obtained with the reverberated ASR models and BLSTM de-noising without de-reverberation, we find that the latter works better at lower SNRs and performs worse at higher SNRs. This can be attributed to higher variances of the reverberated ASR models.

In the following, let us further investigate the relation between back-end refinement and front-end enhancement. The most obvious back-end adaptation is to consider multi-condition training using noisy data, as is done in the CHiME 'noisy' baseline acoustic models. As front-end enhancement, we investigate Log-FB de-noising (yielding best results with the reverberated models) and Log-FB de-noising and de-reverberation with log spectral subtraction (best results with the clean models). Results are shown in Table 4.

Without any front-end enhancement, the CHiME multi-condition baseline yields an average WER of 58.27 % on the development set, improving by over 40 % absolute with respect to the clean models. With BLSTM de-noising, an additional improvement of 8 % absolute WER is observed. If we re-train the multi-condition models using the BLSTM de-noised training set, average WER is decreased to 43.38 %. The gain by re-training is especially visible at higher SNRs. When using BLSTM de-noising and de-reverberation, we obtain additional improvements in the re-trained multi-condition models at higher SNRs ($\geq$ 3 dB), at the expense of reduced accuracy at lower SNRs. This is in line with the observations made above without noisy training.

The system proposed by Tachioka et al. (2013) exploiting LDA, MLLT and SAT with fMLLR adaptation achieves better results without front-end enhancement than the best CHiME baseline system with front-end enhancement (40.00 % average WER)[4]. However, using BLSTM

[4]Note that this result is much better (6 % absolute WER difference) than the corresponding result reported by Tachioka

Table 4: *ASR evaluation of BLSTM enhanced features in multi-condition trained models: EM-ML trained CHiME Challenge baseline models and discriminatively trained models using feature transformations (see text). Multi-condition training using unenhanced (= no re-training) and enhanced (= re-training) noisy and reverberated CHiME training set. Feature enhancement (FE) type: de-noising with Log-FB front-end (Table 3) or de-noising + de-reverberation with Log-FB + SSub front-end (Table 3). Evaluation on CHiME 2013 track 2 development set.*

WER [%]	**Retraining**	**SNR** [dB]						**Mean**
FE type		-6	-3	0	3	6	9	
EM-ML trained recognizer (Vincent et al., 2013)								
—	—	73.17	67.43	59.89	55.71	49.07	44.34	58.27
de-noising	✗	62.68	56.89	51.36	47.90	43.02	40.94	50.47
de-noising	✓	**57.74**	**51.14**	**43.85**	39.10	35.54	32.88	**43.38**
+de-rev.	✗	65.49	60.22	54.79	50.10	47.87	43.92	53.73
+de-rev.	✓	62.34	53.39	45.17	**38.93**	**34.17**	**29.63**	43.94
EM-ML trained recognizer with feature transformations (Tachioka et al., 2013)								
—	—	59.63	49.97	40.60	34.72	29.56	25.52	40.00
de-noising	✗	**46.35**	**37.94**	**31.20**	27.45	**23.60**	**21.12**	**31.28**
de-noising	✓	49.45	41.08	33.18	29.31	25.14	22.05	33.37
+de-rev.	✗	48.77	39.21	33.04	**27.26**	24.52	21.26	32.34
+de-rev.	✓	55.60	46.79	38.69	31.60	27.57	22.78	37.17
Boosted MMI trained recognizer with feature transformations (Tachioka et al., 2013)								
—	—	56.47	47.12	38.47	31.86	27.50	23.32	37.46
de-noising	✗	47.91	40.30	33.09	28.22	25.11	22.70	32.89
de-noising	✓	**43.71**	**35.12**	**27.66**	24.90	21.55	18.56	**28.58**
+de-rev.	✗	57.86	50.48	44.36	39.81	36.63	33.49	43.77
+de-rev.	✓	47.31	37.65	30.15	**24.30**	**20.83**	**18.00**	29.71

de-noising, we gain another 8.7 % absolute accuracy improvement (31.28 % WER) 'on top'. Interestingly, we find results to be best in a 'plug-and-play' setup where the feature transformations are estimated on noisy data instead of enhanced data – thus, there seems to be a larger mismatch in the enhanced features than in the noisy features across training and development set. This could be due to the networks being trained speaker-independently – in the future, we could investigate enhancement on the features after applying the SAT transformation.

Finally, we observe that BLSTM feature enhancement is also complementary to discriminative training using boosted MMI. Boosted MMI and feature transformations without front-end enhancement yield 37.46 % WER, while the best combination (BLSTM de-noising, feature transformations, boosted MMI training using enhanced noisy data) results in 28.58 % average WER on the development set. Notably, when using boosted MMI training using unenhanced data and evaluating using de-noised and de-reverberated data, results are vastly degraded (43.77 % WER) while reasonable results are obtained with re-training (29.71 % WER) – this probably indicates in a large mismatch of the phoneme errors on unenhanced and enhanced data, leading to overfitting.

Since fMLLR adaptation, as used by Tachioka et al. (2013), requires the utterances of each

et al. (2013), because for a fair comparison we use beam-forming as in the CHiME baseline.

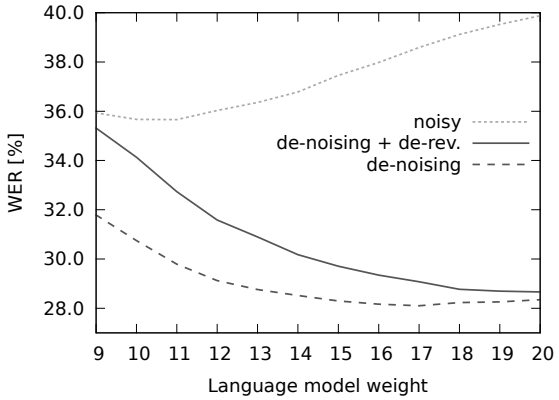


Figure 2: *Influence of language model weight on WER on CHiME 2013 track 2 development set for noisy features, BLSTM de-noising and de-reverberation.*

speaker to be processed at once, it is not suitable for real-time applications such as dialog systems; it is thus of interest to also consider results without adaptation (and hence without SAT). In this case, best performances (not shown in Table 4) are obtained using the de-noising (not de-reverberation) front-end, with recognizer re-training, leading to 35.87 % (instead of 33.37 %) average WER in the EM-ML and 30.82 % (instead of 28.58 %) WER in the discriminatively trained recognizer (without de-noising and without SAT: 46.07 %, 45.13 %).

For the results reported so far (Table 1, 3, and 4), a constant language model weight (μ) of 15 has been used for a fair comparison of results. However, we found that since 'cleaner' features yielded generally higher acoustic likelihoods, the language model weight should be increased accordingly. We determined an optimal weight $\mu^* \in \{9, 10, 11, \ldots, 20\}$ on the development set. Results are displayed in Figure 2. It can clearly be seen that the 'cleaner' the features, the higher the language model weight has to be for optimal performance. In the boosted MMI system using feature transformations, $\mu^* = 11$ yields 34.18 % WER without BLSTM feature enhancement; 28.10 % WER are obtained at $\mu^* = 17$ with BLSTM de-noising; for BLSTM de-noising and de-reverberation, $\mu^* = 20$, resulting in 28.66 % WER. For comparison, let us note that the best FNN in the best back-end (LDA-MLLT, SAT, fMLLR adaptation, boosted MMI) achieved 33.22 % WER, which is significantly (more than 4 % absolute) better than the noisy baseline but clearly below the BLSTM result.

We now proceed to evaluate selected ASR systems (combinations of back-ends and BLSTM front-ends) on the official CHiME Challenge track 2 test set, and compare to other state-of-the-art approaches. Results are shown in Table 5. The best system without back-end modification (using close-talk acoustic models) yields 42.06 % average WER across SNRs from -6 to 9 dB. This is much better than the result using noise compensation only in the back-end by multi-condition training, in the same HMM framework (55.01 %, Vincent et al. (2013)). It also outperforms a state-of-the-art approach for feature enhancement in the linear Mel frequency domain using non-negative matrix factorization (NMF) (Geiger et al., 2013), which gives 48.07 % WER. Combining BLSTM feature enhancement and multi-condition training results in 39.24 % WER, which is a noticeable improvement but also indicates the limits of the basic HMM recognizer framework. Still, this result is better than our previous result with multi-stream HMM fusion of

Table 5: *Final CHiME 2013 track 2 test set evaluation of ASR systems with BLSTM feature enhancement and comparison to related approaches.*

WER [%]						
SNR [dB]						**Mean**
-6	-3	0	3	6	9	
Systems using BLSTM feature enhancement						
BLSTM de-noising + de-reverberation / base WSJ-0						
61.55	50.64	43.84	37.04	31.94	27.33	42.06
BLSTM de-noising / CHiME multi-condition baseline						
53.18	44.97	40.65	34.32	32.39	29.93	39.24
BLSTM de-noising / feat. transf. + MMI						
35.55	27.11	22.40	17.45	16.14	14.29	**22.16**
BLSTM de-noising + de-rev. / feat. transf. + MMI						
37.79	28.71	23.37	18.70	15.41	12.70	22.78
Other systems for CHiME 2013 track 2 task						
CHiME multi-condition baseline (Vincent et al., 2013)						
70.43	63.09	58.42	51.06	45.32	41.73	55.01
NMF / CHiME multi-condition baseline (Geiger et al., 2013)						
61.85	55.58	50.94	43.51	39.14	37.40	48.07
BLSTM-HMM double-stream recognizer (Geiger et al., 2013)						
58.57	50.07	43.94	37.06	32.67	28.25	41.76
Binary masking / feat. transf. + MMI (Tachioka et al., 2013)						
44.12	35.46	28.12	21.20	17.43	14.83	26.86
FNN feature enhancement / feat. transf. + MMI						
42.11	33.08	26.17	21.43	18.08	16.05	26.15

multi-condition EM-ML trained MFCC-GMMs and a BLSTM phone recognizer (Geiger et al. (2013), 41.76 % WER). In this work, a deep BLSTM was used as a secondary acoustic model providing frame-wise phoneme probabilities, instead of performing front-end enhancement. Using the BLSTM front-end, but changing the back-end to a state-of-the-art system exploiting feature transformations and discriminative training (Tachioka et al., 2013), 22.16 and 22.78 % WER are obtained in combination with BLSTM de-noising and de-noising / de-reverberation. This is, to the best of our knowledge, the best recorded score on the CHiME 2 track 2 test set at the time of this writing, and a 17 % relative improvement over our previous best result in the challenge (26.73 % WER, cf. (Weninger et al., 2013)). The BLSTM result also outperforms FNN feature enhancement by 4 % absolute; in turn, FNN enhancement in the front-end seems to perform slightly better than binary masking (Tachioka et al., 2013).

6. Conclusion and Outlook

We have demonstrated the efficacy of data-based feature enhancement using deep recurrent neural networks for ASR in non-stationary convolutive noise. Enhancement has yielded significant improvements for every single ASR system investigated in this study. Reasonable results have been achieved even with unmodified close-talk acoustic models, which otherwise fail at

decoding the CHiME utterances. Best results on the 2013 2nd CHiME Challenge track 2 task[5] have been achieved by combining enhancement with feature transformations and discriminative HMM training. An average WER of 22.16 % is measured, whereas simple multi-condition HMM training yields an average WER of 55.01 %. The improvements by the proposed BLSTM feature enhancement method are all the more noticeable since it does not directly exploit phonetic information. Still, our method has been shown to be complementary with approaches that do, such as using LDA and MMI in back-end recognizer training. While other neural network approaches such as FNN enhancement with feature frame stacking also provide complementary gains to these back-end improvements, BLSTM enhancement has delivered most promising results on the CHiME task. In future research, it will be interesting to investigate how the use of more training data (such as noisy speech from arbitrary sources), also in generative pre-training, affects the performance of FNN, RNN, and BLSTM-RNN. For example, Mohamed et al. (2012) report phoneme error reductions in the order of 1 % absolute (5 % relative) on the TIMIT database by pre-training.

An advantage of the proposed method over data-based monaural Mel or Fourier domain feature enhancement by NMF (Weninger et al., 2012; Hurmalainen et al., 2011) is that the complexity of the model does not depend on the amount of training data, and that most of the computational complexity involved is shifted to a training phase, while evaluation can be done very efficiently – in contrast to typical NMF approaches involving little to no model pre-training but considerable effort in model evaluation. Despite the temporal dependencies in recurrent neural networks, they can be trained efficiently on graphics processing units (GPUs), as can be verified by the interested reader, by downloading our open-source CUDA RecurREnt Neural Network Toolkit (CURRENNT, cf. above). CURRENNT is delivered with a subset of the CHiME 2013 feature enhancement task as use case.

In contrast to other enhancement techniques such as factorial models (Rennie et al., 2008; Weninger et al., 2012), our approach learns frame-by-frame correspondences between distorted and clean training features. Hence, the most straightforward approach to generate training data is to algorithmically apply distortions to clean data, as done in the CHiME Challenges and previous evaluations such as the AURORA-4 database. Still, realistic training data is not trivial to obtain (it could be done, e.g., by loudspeaker playback and recording in various settings involving real noise and reverberation). A more promising approach might be to use semi-supervised learning, initialized by large amounts of systematically generated training data using combinations of speech and noise corpora, and continuing using real noisy and reverberated speech for which no 'clean' counterpart exists.

One important issue in deep learning methods, as compared to 'blind' de-noising and dereverberation approaches, is generalization to unseen test scenarios. In the future, this might be improved by extended multi-task regularization, i.e., including additional training targets such as noise magnitudes or phonetic information. A related approach would be to use deep learning to add a phoneme classification layer on top of the feature enhancement layers, in order to further improve performance of BLSTM phoneme predictors (cf. Geiger et al. (2013)) in challenging settings. Including noise context, i.e., training on noisy streams instead of end-pointed but corrupted speech data, might also help generalization – we already have evidence that LSTM networks are very well suited to voice activity detection in noise (Eyben et al., 2013).

[5]Note that our result could not have been an official competition result in the Challenge, because learning a mapping between noisy and clean features was not allowed as per the Challenge guidelines (Vincent et al., 2013).

Acknowledgments

The research leading to these results has received funding from the Federal Republic of Germany through the German Research Foundation (DFG) under grant no. SCHU 2508/4-1. 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) programme.

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. Zixing Zhang assisted in preparation of the experimental data.

References

Anastasakos, T., McDonough, J., Makhoul, J., 1997. Speaker adaptive training: A maximum likelihood approach to speaker normalization. In: Proc. of ICASSP. IEEE, pp. 1043–1046.

Barker, J. P., Vincent, E., Ma, N., Christensen, H., Green, P. D., 2013. The PASCAL CHiME speech separation and recognition challenge. Computer Speech and Language 27 (3), 621–633.

Bengio, Y., Simard, P., Frasconi, P., 1994. Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks 5 (2), 157–66.

de la Torre, A., Peinado, A. M., Segura, J. C., Perez-Cordoba, J. L., Benitez, M. C., Rubio, A. J., 2005. Histogram equalization of speech representation for robust speech recognition. IEEE Transactions on Speech and Audio Processing 13 (3), 355–366.

Eyben, F., Weninger, F., Squartini, S., Schuller, B., May 2013. Real-life Voice Activity Detection with LSTM Recurrent Neural Networks and an Application to Hollywood Movies. In: Proceedings 38th IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2013. IEEE, Vancouver, Canada, pp. 483–487.

Gales, M. J., 1998. Maximum likelihood linear transformations for HMM-based speech recognition. Computer speech & language 12 (2), 75–98.

Gales, M. J. F., Wang, Y. Q., 2011. Model-based approaches to handling additive noise in reverberant environments. In: Proc. IEEE Workshop on Hands-free Speech Communication and Microphone Arrays. Edinburgh, UK, pp. 121 – 126.

Geiger, J. T., Weninger, F., Hurmalainen, A., Gemmeke, J. F., Wöllmer, M., Schuller, B., Rigoll, G., Virtanen, T., June 2013. The TUM+TUT+KUL Approach to the CHiME Challenge 2013: Multi-Stream ASR Exploiting BLSTM Networks and Sparse NMF. In: Proceedings The 2nd CHiME Workshop on Machine Listening in Multisource Environments held in conjunction with ICASSP 2013. IEEE, Vancouver, Canada, pp. 25–30.

Gelbart, D., Morgan, N., 2001. Evaluating long-term spectral subtraction for reverberant ASR. In: Proc. of ASRU. IEEE, Madonna di Campiglio, Italy, pp. 103–106.

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

Graves, A., 2008. Supervised sequence labelling with recurrent neural networks. Ph.D. thesis, Technische Universität München.

Graves, A., Mohamed, A., Hinton, G., May 2013. Speech recognition with deep recurrent neural networks. In: Proc. of ICASSP. IEEE, Vancouver, Canada, pp. 6645–6649.

Gray, A., Markel, J., 1976. Distance measures for speech processing. IEEE Transactions on Acoustics, Speech and Signal Processing 24 (5), 380–391.

Hermansky, H., Morgan, N., Bayya, A., Kohn, P., 1992. RASTA-PLP speech analysis technique. In: Proceedings of International Conference on Acoustics, Speech, and Signal Processing. Vol. 1. pp. 121–124.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., Kingsbury, B., 2012. Deep neural networks for acoustic modeling in speech recognition. IEEE Signal Processing Magazine 29 (6), 82–97.

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.

Maas, A. L., O'Neil, T. M., Hannun, A. Y., Ng, A. Y., June 2013. Recurrent neural network feature enhancement: The 2nd CHiME challenge. In: Proceedings The 2nd CHiME Workshop on Machine Listening in Multisource Environments held in conjunction with ICASSP 2013. IEEE, Vancouver, Canada, pp. 79–80.

Maas, R., Schwarz, A., Zheng, Y., Reindl, K., Meier, S., Sehr, A., Kellermann, W., 2011. A two-channel acoustic front-end for robust automatic speech recognition in noisy and reverberant environments. In: Proc. of CHiME. pp. 41–46.

Mohamed, A., Hinton, G., Penn, G., 2012. Understanding how deep belief networks perform acoustic modelling. In: Proc. of ICASSP. Kyoto, Japan, pp. 4273–4276.

Nesta, F., Matassoni, M., Astudillo, R. F., 2013. A flexible spatial blind source extraction framework for robust speech recognition in noisy environments. In: Proc. of CHiME. Vancouver, Canada, pp. 33–38.

Parveen, S., Green, P., 2004. Speech enhancement with missing data techniques using recurrent neural networks. In: Proc. of ICASSP. Montreal, Canada.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlícek, P., Qian, Y., Schwarz, P., et al., 2011. The Kaldi speech recognition toolkit. In: Proc. of ASRU.

Povey, D., Kanevsky, D., Kingsbury, B., Ramabhadran, B., Saon, G., Visweswariah, K., 2008. Boosted MMI for model and feature-space discriminative training. In: Proc. of ICASSP. IEEE, pp. 4057–4060.

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.

Saon, G., Padmanabhan, M., Gopinath, R., Chen, S., 2000. Maximum likelihood discriminant feature spaces. In: Proc. of ICASSP. Vol. 2. IEEE, pp. 1129–1132.

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

Seltzer, M. L., Droppo, J., 2013. Multi-task learning in deep neural networks for improved phoneme recognition. In: Proc. of ICASSP. IEEE, Vancouver, Canada, pp. 6965–6969.

Seltzer, M. L., Yu, D., Wang, Y., 2013. An investigation of deep neural networks for noise robust speech recognition. In: Proc. of ICASSP. Vancouver, Canada, pp. 7398–7402.

Tachioka, Y., Watanabe, S., Hershey, J. R., 2013. Effectiveness of discriminative training and feature transformation for reverberated and noisy speech. In: Proc. of ICASSP. Vancouver, Canada, pp. 6935–6939.

Vertanen, K., 2006. Baseline WSJ acoustic models for HTK and Sphinx: Training recipes and recognition experiments. Tech. rep., Cavendish Laboratory, University of Cambridge.

Vincent, E., Barker, J., Watanabe, S., Le Roux, J., Nesta, F., Matassoni, M., 2013. The second 'CHiME' speech separation and recognition challenge: Datasets, tasks and baselines. In: Proc. of ICASSP. Vancouver, Canada, pp. 126–130.

Weninger, F., Geiger, J., Wöllmer, M., Schuller, B., Rigoll, G., June 2013. The Munich Feature Enhancement Approach to the 2013 CHiME Challenge Using BLSTM Recurrent Neural Networks. In: Proceedings The 2nd CHiME Workshop on Machine Listening in Multisource Environments held in conjunction with ICASSP 2013. IEEE, Vancouver, Canada, pp. 86–90.

Weninger, F., Wöllmer, M., Geiger, J., Schuller, B., Gemmeke, J. F., 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. IEEE, Kyoto, Japan, pp. 4681–4684.

Wöllmer, M., Marchi, E., Squartini, S., Schuller, B., 2011a. Multi-stream LSTM-HMM decoding and histogram equalization for noise robust keyword spotting. Cognitive Neurodynamics 5 (3), 253–264.

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

Wöllmer, M., Zhang, Z., Weninger, F., Schuller, B., Rigoll, G., 2013. Feature enhancement by bidirectional LSTM networks for conversational speech recognition in highly non-stationary noise. In: Proc. of ICASSP. Vancouver, Canada, pp. 6822–6826.

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P., 2006. The HTK book (v3.4). Cambridge University Press.