PROBABILISTIC ASR FEATURE EXTRACTION APPLYING CONTEXT-SENSITIVE CONNECTIONIST TEMPORAL CLASSIFICATION NETWORKS

Martin Wöllmer[1], *Björn Schuller*[2], *Gerhard Rigoll*[2]

[1]BMW Group, Munich, Germany
[2]Institute for Human-Machine Communication, Technische Universität München, Germany
martin.woellmer@bmw.de

ABSTRACT

This paper proposes a novel automatic speech recognition (ASR) front-end that unites the principles of bidirectional Long Short-Term Memory (BLSTM), Connectionist Temporal Classification (CTC), and Bottleneck (BN) feature generation. BLSTM networks are known to produce better probabilistic ASR features than conventional multilayer perceptrons since they are able to exploit a self-learned amount of temporal context for phoneme estimation. Combining BLSTM networks with a CTC output layer implies the advantage that the network can be trained on unsegmented data so that the quality of phoneme prediction does not rely on potentially error-prone forced alignment segmentations of the training set. In challenging ASR scenarios involving highly spontaneous, disfluent, and noisy speech, our BN-CTC front-end leads to remarkable word accuracy improvements and prevails over a series of previously introduced BLSTM-based ASR systems.

Index Terms— Automatic Speech Recognition, Connectionist Temporal Classification, Tandem Features, Long Short-Term Memory

1. INTRODUCTION

Aiming to improve the robustness of automatic speech recognition (ASR) systems, more and more researchers are focusing on so-called tandem ASR front-ends in which neural networks are applied to produce probabilistic features serving as input for Hidden Markov Models (HMMs). In most cases, these neural networks are feed-forward multilayer perceptrons (MLP), meaning that they are composed of multiple hidden layers. MLPs used within tandem front-ends are usually trained to map form MFCC or PLP features to framewise phoneme or phoneme state labels so that the output activations of the trained network can be used as speech features, after being logarithmized and decorrelated. Such probabilistic features typically lead to better speech recognition performance than conventional cepstral features [1–3].

Alternatively to MLP output activations, also activations within a hidden layer of the network were shown to be suited as speech features. Using hidden layer activations of an MLP-based phoneme predictor as HMM input has the advantage that by choosing the size of the corresponding hidden layer, the dimensionality of the probabilistic feature vector can be defined, so that subsequent dimensionality reduction may be omitted. Compared to the other hidden layers, the hidden layer whose activations are used as features tends to be small, so that these features are usually referred to as 'bottleneck' (BN) features [4–6]. Bottleneck features are known to increase word accuracies, especially for extremely difficult ASR tasks such as recognizing noisy, conversational speech containing non-linguistic vocalizations, disfluencies, and emotional coloring [4, 7–9].

Recently, it was shown that in order to capture co-articulation effects and higher level context in human speech, the application of bidirectional Long Short-Term Memory (BLSTM) networks [10–12] leads to better results than simple feature frame stacking as it is usually done within MLP front-ends [13]. By replacing the hidden neurons with so-called 'memory blocks', BLSTM networks are able to learn the amount of temporal context that has to be considered for the respective sequence labeling task and can model context over longer time spans than standard recurrent neural networks (RNNs). First promising results in phoneme-based keyword spotting via BLSTM networks [14] motivated further research in the areas of continuous ASR [15], noise robust speech recognition [16], language modeling [17], and Bottleneck-BLSTM front-ends [18].

Since the networks within probabilistic front-ends need to be trained on framewise targets, a forced alignment of the transcriptions in the training set has to be done prior to neural network training in order to determine the segment boundaries. Hence, potential errors in the forced alignment limit the quality of the phoneme predictor that is to be trained. In [19], a solution to this problem is proposed: If the networks are equipped with a so-called Connectionist Temporal Classification (CTC) output layer, the segment boundaries do not have to be known during training as the network is allowed to choose the location of each label. This technique has led to excellent results in phoneme recognition [19], handwriting recognition [20], and keyword spotting [21].

The aim of this paper is to combine the CTC idea with the Bottleneck-BLSTM front-end we proposed in [18], so that the resulting ASR feature extractor profits from contextual knowledge by BLSTM modeling, robust feature generation via bottleneck networks, and flexible phoneme boundary modeling via CTC. To enable a comparison between the proposed CTC features and earlier attempts to exploit BLSTM for continuous ASR, we conduct experiments on the 'COnversational Speech In Noisy Environments' (COSINE) corpus [22] and on the Buckeye corpus [23]. Both databases contain instationary noise sources as well as spontaneous and colloquial speaking styles that typically lead to high error rates.

The structure of this paper is as follows: First, we review the principle of Connectionist Temporal Classification in Section 2. Next, we describe the concept of Bottleneck-BLSTM networks in Section 3 and our method of CTC feature extraction in Section 4. Experiments are shown in Section 5 before we draw conclusions in Section 6 and provide references to related work in Section 7.

This research has been supported by the German Research Foundation (DFG) through grant no. SCHU 2508/4.

2. CONNECTIONIST TEMPORAL CLASSIFICATION

A major problem with the standard objective functions for RNNs is that they require individual targets for each point in the data sequence, which in turn requires the boundaries between segments with different labels (e. g., the phoneme boundaries in speech) to be pre-determined. The Connectionist Temporal Classification output layer [19] solves this problem by allowing 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.

A CTC layer has as many output units as there are distinct labels for a task, plus an extra *blank* unit for no label. The activations of the outputs at each timestep are normalized and interpreted as the probability of observing the corresponding label (or no label) at that point in the sequence. These probabilities are conditionally independent given the input sequence. Thus, with $x_{1:T}$ being a length T feature vector input sequence and o_t^q representing the activation of output unit q at time t, the total probability of a given (framewise) sequence $z_{1:T}$ of blanks and labels is

$$p(z_{1:T}|x_{1:T}) = \prod_{t=1}^{T} o_t^{z_t}. \quad (1)$$

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., $\mathcal{B}(AA--BBB-B) = ABB$. The total probability of the resulting length V labeling $l_{1:V}$, where $V \leq T$, then is

$$p(l_{1:V}|x_{1:T}) = \sum_{z_{1:T}:\mathcal{B}(z_{1:T})=l_{1:V}} p(z_{1:T}|x_{1:T}). \quad (2)$$

A naive calculation of Equation (2) is unfeasible, because the number of $z_{1:T}$ terms corresponding to each labeling increases exponentially with the sequence length. However, $p(l_{1:V}|x_{1:T})$ can be efficiently calculated with a dynamic programming algorithm similar to the forward-backward algorithm for HMMs (see [19]).

An RNN with a CTC output layer can be trained with gradient descent by backpropagating through time the partial derivatives of the objective function with respect to the output activations. When a new input sequence is presented to a network trained with CTC, the output activations (corresponding to the label probabilities) tend to form single frame *spikes* separated by long intervals where the blank label is emitted. The location of the spikes corresponds to the portion of the input sequence where the label is detected.

3. BOTTLENECK-BLSTM NETWORKS

In tandem ASR systems, the output activations of neural networks trained on framewise phoneme or phoneme state targets are used as probabilistic features, alternatively to (or in combination with) standard MFCC features. For enhanced probabilistic feature generation, standard multilayer perceptrons (MLPs) can be replaced by bidirectional LSTM networks [12] which allow to access and model long-range temporal context information via so-called *memory blocks* substituting the conventional neurons in the network's hidden layers. Generally, bidirectional networks consist of two sets of hidden layers, one for forward and one for backward processing. This enables the incorporation of past and future context and captures for example co-articulation effects in human speech (for more details, see [12]). In [24], an ASR front-end using logarithmized BLSTM output activations in combination with MFCC features was proposed.

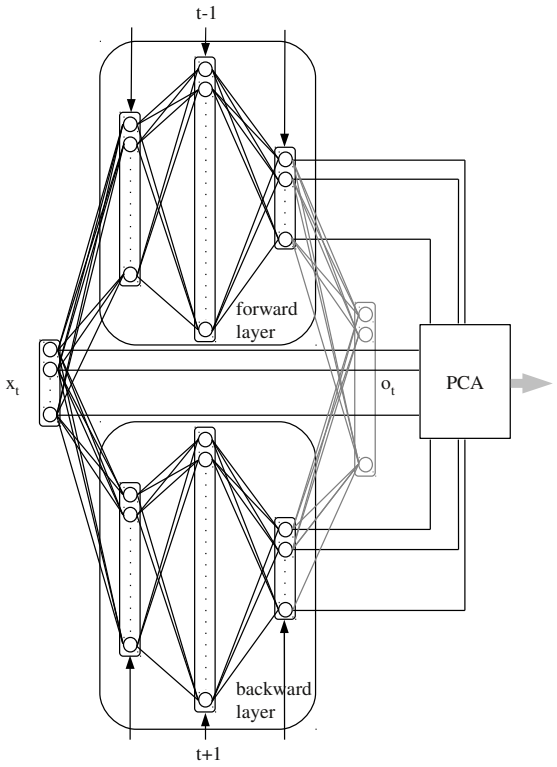

Fig. 1: Architecture of the Bottleneck-BLSTM front-end.

Combining BLSTM based feature generation with the 'bottleneck' idea proposed in [4] was shown to lead to lower error rates in spontaneous speech recognition [18]. The bottleneck principle allows to generate tandem feature vectors of arbitrary size by using the activations of a narrow hidden (bottleneck) layer as features – rather than the logarithmized output activations corresponding to the estimated phoneme or phoneme state posteriors.

Figure 1 illustrates the detailed structure of the Bottleneck-BLSTM front-end considered in our experiments. Since we focus on *bidirectional* processing, we have two bottleneck layers: one within the network processing the speech sequence in forward direction and one within the network for backward processing. The MFCC feature vectors x_t serve as input for a BN-BLSTM network that is trained on framewise phoneme targets. During feature extraction, the activations of the output layer are ignored; only the activations of the forward and backward bottleneck layer are processed (i. e., the memory block outputs of the bottleneck layers). Together with the original MFCC features, the forward and backward bottleneck layer activations are concatenated to one large feature vector which is then decorrelated and dimensionality reduced by Principal Component Analysis (PCA). In Figure 1, the connections between the bottleneck layers and the output layer are depicted in grey, indicating that the activations of the output layer (o_t) are only used during network training and not during BN-BLSTM feature extraction.

4. CTC FEATURE GENERATION

Alternatively to BLSTM features as described in Section 3, this study considers the extraction of CTC features, i. e., features derived from the activations of a network using the BLSTM principle in combination with Connectionist Temporal Classification (see Section 2). This implies the advantage that the neural network applied within the probabilistic feature extractor does not have to be trained on framewise phoneme targets but on the unsegmented phoneme label sequences $l_{1:V}$. Hence, CTC feature extraction does not rely on error-prone forced alignments which in turn can affect the phoneme modeling accuracy of the neural network in a negative way if the segments are inaccurate.

The input layer of our CTC network is of size 39, corresponding to the 39 cepstral mean and variance normalized MFCC features (including deltas and double deltas) that are extracted from the speech signal every 10 ms using a window size of 25 ms. Both the forward and the backward branch of the underlying BLSTM architecture consist of three hidden layers with 78, 128, and 80 memory blocks, respectively. This is equivalent to the configuration used for BLSTM feature generation in [18]. The CTC output layer comprises 42 output nodes which corresponds to 41 phoneme targets and one blank label (see Section 2). During CTC network training we use a learning rate of 10^{-4} and a momentum of 0.9. Zero mean Gaussian noise with standard deviation 0.6 is added to the input activations in the training phase in order to improve generalization. Prior to training, all weights are randomly initialized in the range from -0.1 to 0.1. In the training phase, we evaluate the overall label error rate on a development set after every fifth epoch. Training is aborted as soon as no improvement on the development set can be observed during the last 50 epochs, and the network that achieved the best label error rate on the development set is chosen as the final network.

To gaussianize the framewise output activations o_t of the trained CTC feature extractor, we take the logarithm of each CTC output. These logarithmized CTC outputs are then appended to the original 39-dimensional MFCC feature vector, resulting in an extended 81-dimensional tandem feature vector. To reduce the dimensionality of the feature vector and to decorrelate its components, PCA is applied. The resulting features will be called 'CTC features' in the following.

As an alternative to these CTC features, we also consider the bottleneck technique within our CTC feature generation framework. Thus, similar to the BN-BLSTM feature extractor described in Section 3, we collect activations of a hidden layer within our CTC network. Analogous to [18], we chose the third hidden layer (size 80) as bottleneck layer. After appending the original MFCC features, we end up with 199 feature vector components (39 MFCC features, 80 bottleneck features from the forward branch, and 80 bottleneck features from the backward branch). Again, we apply PCA to reduce the feature vector dimensionality. In the following, these features will be referred to as 'BN-CTC features'.

5. EXPERIMENTS

5.1. Databases

To enable comparisons between the proposed CTC feature extractor and previously introduced concepts for BLSTM modeling of spontaneous speech, we use the 'COnversational Speech In Noisy Environments' (COSINE) corpus [22] which has also been used in [15], [13], and [18]. The COSINE corpus contains multi-party conversations recorded in real world environments. The recordings were captured on a wearable recording system so that the speakers were able to walk around during recording. Since the participants were asked to speak about anything they liked and to walk to various noisy locations, the corpus consists of natural, spontaneous, and highly disfluent speaking styles partly masked by indoor and outdoor noise sources such as crowds, vehicles, and wind. The recordings were captured using multiple microphones simultaneously, however, to match most application scenarios, we exclusively used speech recorded by a close-talking microphone (Sennheiser ME-3). Details on the speech recognition task and on the speaker-independent division into training, development, and test partition can be found in [15] and [9].

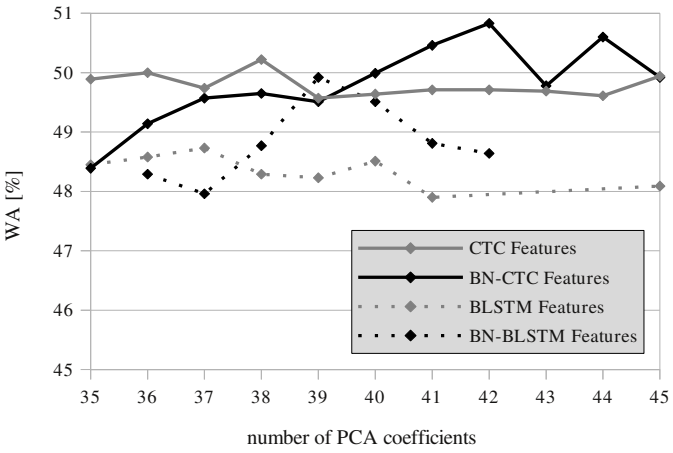

Fig. 2: Word accuracy (WA) on the COSINE test set for different numbers of PCA coefficients: CTC, Bottleneck-CTC, BLSTM, and Bottleneck-BLSTM features.

In conformance with [18], the best system configurations are also evaluated on the Buckeye corpus [23]. It contains recordings of interviews with 40 subjects, who were told that they were in a linguistic study on how people express their opinions. The corpus was originally intended to study phonetic variation among speakers, and has been used for a variety of phonetic studies as well as for ASR experiments [25]. Similar to the COSINE database, the contained speech is highly spontaneous. Further details on the corpus can be found in [9].

5.2. Experimental Setup

In what follows, we compare the CTC and BN-CTC features proposed in Section 4 to the (BN-)BLSTM features introduced in [18] and to alternative approaches for BLSTM-based phoneme modeling using a discretized BLSTM feature (see [13] and [15]). The HMM system applied for processing our probabilistic features is identical to the back-end used to determine the baseline HMM results in [18]: Each phoneme is represented by three emitting states (left-to-right HMMs) with 16 Gaussian mixtures. The initial monophone HMMs were mapped to tied-state cross-word triphone models with shared state transition probabilities. Two Baum-Welch iterations were performed for re-estimation of the triphone models. Finally, the number of mixture components of the triphone models was increased to 16 in four successive rounds of mixture doubling and re-estimation (four iterations in every round). Both, acoustic models and a back-off bigram language model were trained on the training set of the COSINE corpus.

Table 1: Word accuracies on the COSINE and Buckeye test set for CTC, Bottleneck-CTC, and other BLSTM-based front-ends introduced in [18], [13], and [15].

	WA [%]	
features	**COSINE**	**Buckeye**
Bottleneck-CTC features	**50.83**	57.27
CTC features	50.22	**58.46**
Bottleneck-BLSTM features [18]	49.92	58.21
BLSTM features [18]	48.23	57.80
discrete BLSTM (multi-stream) [13]	48.01	56.61
discrete BLSTM [15]	45.04	55.91
MFCC	43.36	50.97

5.3. Results

For our initial experiments, we focus on the COSINE corpus and investigate how the number of PCA coefficients used as final feature vector affects the word accuracy (WA) on the COSINE test set. Thus, we train and evaluate ASR systems based on feature vectors containing 35 to 45 principle components, i. e., the principal components corresponding to the 35 to 45 largest eigenvalues. In Figure 2, the results obtained by CTC, Bottleneck-CTC, BLSTM, and Bottleneck-BLSTM features are shown. The BLSTM and BN-BLSTM results are taken from [9]. When we compare CTC and BLSTM features, we find that CTC features lead to better word accuracies than BLSTM features with a performance gain of up to 2 % absolute. The influence of the number of PCA coefficients on the recognition performance is rather low for both front-ends. The best WA for CTC features (50.22 %) is reached for a front-end using 38 principal components. When considering bottleneck features, we see that the dependency on the number of PCA coefficients is more pronounced: For BN-BLSTM features, there is a clear WA maximum for 39-dimensional feature vectors (WA of 49.92 %) while the BN-CTC front-end performs best if 42 principal components are used, leading to the overall best WA of 50.83 % on the COSINE test set.

Table 1 compares the best results obtained by BN-CTC and CTC features with the word accuracies reported in [18] for (Bottleneck-) BLSTM features. Additionally, the performance of an ASR system processing MFCC features and a discretized maximum-likelihood BLSTM phoneme prediction feature in a single (see [15]) or in multiple feature streams (see [13]) is shown. Starting from a simple MFCC-HMM system as described in Section 5.2 (WA of 43.36 %), the word accuracy on the COSINE test set can be increased to 45.04 % and 48.01 % by modeling an additional BLSTM-based phoneme prediction feature using a single-stream and a multi-stream HMM, respectively. Applying tandem BLSTM and Bottleneck-BLSTM features (see Section 3), a further performance gain can be observed, leading to a WA of 48.23 % and 49.92 % (39 principal components, see [18]). Best word accuracies on the COSINE corpus are reached with the CTC and BN-CTC front-end proposed in this paper: As shown in Figure 2, WA can be increased to up to 50.22 % and 50.83 %, respectively, by employing Connectionist Temporal Classification within the BLSTM network for probabilistic feature extraction. For the COSINE experiment, we can see that since the training of the CTC network does not require any knowledge about the phoneme boundaries, phonemes are modeled more accurately within the front-end, which in turn implies a better word accuracy of the overall ASR system. Especially for challenging corpora and recognition tasks as considered in the COSINE experiment, we can assume that spontaneous and disfluent speaking styles as well as background noise lead to errors in the forced alignment transcriptions needed for BLSTM network training. This of course limits the accuracy of the phoneme estimates generated by the BLSTM network and thus negatively affects the quality of the generated BLSTM features. The CTC principle allows us to train our front-end on unsegmented speech and enables the generation of enhanced probabilistic features.

Next, we train and evaluate all considered ASR systems on the Buckeye corpus. The Buckeye corpus also contains spontaneous speaking styles but is less noisy than the COSINE database, which leads to a higher baseline HMM word accuracy of 50.97 %. Note that for the Buckeye experiment, we use the same system configuration as for the COSINE experiment, without any further optimizations, i. e., 38 principal components are used in the CTC front-end and 42 principal components are used within the BN-CTC feature extractor. As shown in Table 1, the word accuracy on the Buckeye test set can be increased to up to 58.46 % when applying CTC features. Compared to the COSINE experiment, the performance difference between BLSTM and CTC features is less pronounced. This smaller performance gap can be attributed to the fact that the forced alignments of the Buckeye training set are more accurate as the utterances contain less noise than the utterances in the COSINE corpus. Thus, we can conclude that CTC features tend to be best suited for extremely challenging ASR scenarios that involve non-stationary interfering noise sources and conversational speaking styles that are hard to recognize and typically lead to erroneous forced alignments.

6. CONCLUSION

We showed how context-sensitive ASR tandem feature generation can be enhanced via Connectionist Temporal Classification. Building on our Bottleneck-BLSTM front-end as proposed in [18], we investigated how a CTC output layer incorporated into a BLSTM network for phoneme prediction affects the word accuracy of the resulting ASR system. By using CTC, we are able to train our network on unsegmented data and thus do not rely on the quality of forced alignment segmentations that would be needed if a standard output layer was applied. We found that especially for very noisy and spontaneous speech, CTC-based probabilistic feature extraction prevails over comparable BLSTM features. Generally, it seems to be advantageous to use CTC and Bottleneck-CTC features, whenever an ASR system shall be trained on noise corrupted spontaneous speech in order to match the conditions that are expected during testing. In such cases, errors in the forced alignments of the training set tend to undermine the full potential of BLSTM-based front-ends, as the network is partly trained on incorrect framewise phoneme targets. In future work, we will further investigate the relation between errors in the forced alignment and BLSTM phoneme prediction quality. Furthermore, we plan to combine our CTC front-end with methods for speech enhancement such as non-negative matrix factorization [26].

7. RELATION TO PRIOR WORK

Prior work on BLSTM-based continuous speech recognition includes [15], [13], [18], and [8]. Connectionist Temporal Classification was first introduced in [19]. First studies on speech processing via CTC comprise keyword spotting based on whole-word models [27] and phoneme models [21]. This paper uses the experimental framework of [18] and shows how CTC can be incorporated into Bottleneck-BLSTM front-ends for enhanced recognition accuracies in challenging ASR tasks.

8. REFERENCES

[1] H. Hermansky, D. P. W. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proc. of ICASSP*, Istanbul, Turkey, 2000, pp. 1635–1638.

[2] D. P. W. Ellis, R. Singh, and S. Sivadas, "Tandem acoustic modeling in large-vocabulary recognition," in *Proc. of ICASSP*, Salt Lake City, UT, USA, 2001, pp. 517–520.

[3] Q. Zhu, B. Chen, N. Morgan, and A. Stolcke, "Tandem connectionist feature extraction for conversational speech recognition," in *Machine Learning for Multimodal Interaction*, pp. 223–231. Springer, 2005.

[4] F. Grezl, M. Karafiat, K. Stanislav, and J. Cernocky, "Probabilistic and bottle-neck features for LVCSR of meetings," in *Proc. of ICASSP*, Honolulu, Hawaii, 2007, pp. 757–760.

[5] D. Yu and M. L. Seltzer, "Improved bottleneck features using pretrained deep neural networks," in *Proc. of Interspeech*, Florence, Italy, 2011, pp. 237–240.

[6] C. Plahl, R. Schlüter, and H. Ney, "Hierarchical bottle neck features for LVCSR," in *Proc. of Interspeech*, Makuhari, Japan, 2010, pp. 1197–1200.

[7] F. Grezl and P. Fousek, "Optimizing bottle-neck features for LVCSR," in *Proc. of ICASSP*, Las Vegas, NV, 2008, pp. 4729–4732.

[8] F. Weninger, M. Wöllmer, and B. Schuller, "Combining Bottleneck-BLSTM and Semi-Supervised Sparse NMF for Recognition of Conversational Speech in Highly Instationary Noise," in *Proc. of Interspeech*, Portland, Oregon, USA, 2012.

[9] M. Wöllmer and B. Schuller, "Probabilistic speech feature extraction with context-sensitive bottleneck neural networks," *Neurocomputing*, 2012.

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

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

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

[13] M. Wöllmer, B. Schuller, and G. Rigoll, "Feature frame stacking in RNN-based Tandem ASR systems - learned vs. predefined context," in *Proc. of Interspeech*, Florence, Italy, 2011, pp. 1233–1236.

[14] M. Wöllmer, F. Eyben, J. Keshet, A. Graves, B. Schuller, and G. Rigoll, "Robust discriminative keyword spotting for emotionally colored spontaneous speech using bidirectional LSTM networks," in *Proc. of ICASSP*, Taipei, Taiwan, 2009, pp. 3949–3952.

[15] M. Wöllmer, F. Eyben, B. Schuller, and G. Rigoll, "Recognition of spontaneous conversational speech using long short-term memory phoneme predictions," in *Proc. of Interspeech*, Makuhari, Japan, 2010, pp. 1946–1949.

[16] F. Weninger, J. Geiger, M. Wöllmer, B. Schuller, and G. Rigoll, "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, 2011, pp. 24–29.

[17] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM Neural Networks for Language Modeling," in *Proc. of Interspeech*, Portland, Oregon, USA, 2012.

[18] M. Wöllmer, B. Schuller, and G. Rigoll, "A novel Bottleneck-BLSTM front-end for feature-level context modeling in conversational speech recognition," in *Proc. of ASRU*, Waikoloa, Big Island, Hawaii, 2011, pp. 36–41.

[19] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented data with recurrent neural networks," in *Proc. of ICML*, Pittsburgh, USA, 2006, pp. 369–376.

[20] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 855–868, 2009.

[21] M. Wöllmer, F. Eyben, B. Schuller, and G. Rigoll, "Spoken term detection with connectionist temporal classification - a novel hybrid CTC-DBN decoder," in *Proc. of ICASSP*, Dallas, Texas, 2010, pp. 5274–5277.

[22] A. Stupakov, E. Hanusa, D. Vijaywargi, D. Fox, and J. Bilmes, "The design and collection of COSINE, a multi-microphone in situ speech corpus recorded in noisy environments," *Computer Speech and Language*, vol. 26, no. 1, pp. 52–66, 2011.

[23] M. A. Pitt, L. Dilley, K. Johnson, S. Kiesling, W. Raymond, E. Hume, and E. Fosler-Lussier, *Buckeye Corpus of Conversational Speech (2nd release)*, Department of Psychology, Ohio State University (Distributor), Columbus, OH, USA, 2007, [www.buckeyecorpus.osu.edu].

[24] M. Wöllmer and B. Schuller, "Enhancing spontaneous speech recognition with BLSTM features," in *Proc. of NOLISP*, Las Palmas de Gran Canaria, Spain, 2011, pp. 17–24.

[25] F. Weninger, B. Schuller, M. Wöllmer, and G. Rigoll, "Localization of non-linguistic events in spontaneous speech by non-negative matrix factorization and Long Short-Term Memory," in *Proc. of ICASSP*, Prague, Czech Republic, 2011, pp. 5840–5843.

[26] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. of NIPS*, Vancouver, Canada, 2001, pp. 556–562.

[27] S. Fernandez, A. Graves, and J. Schmidhuber, "An application of recurrent neural networks to discriminative keyword spotting," in *Proc. of ICANN*, Porto, Portugal, 2007, pp. 220–229.