A Novel Bottleneck-BLSTM Front-End for Feature-Level Context Modeling in Conversational Speech Recognition

Martin Wöllmer, Björn Schuller, Gerhard Rigoll

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

[woellmer|schuller|rigoll]@tum.de

***Abstract*—We present a novel automatic speech recognition (ASR) front-end that unites Long Short-Term Memory context modeling, bidirectional speech processing, and bottleneck (BN) networks for enhanced Tandem speech feature generation. Bidirectional Long Short-Term Memory (BLSTM) networks were shown to be well suited for phoneme recognition and probabilistic feature extraction since they efficiently incorporate a flexible amount of long-range temporal context, leading to better ASR results than conventional recurrent networks or multi-layer perceptrons. Combining BLSTM modeling and bottleneck feature generation allows us to produce feature vectors of arbitrary size, independent of the network training targets. Experiments on the COSINE and the Buckeye corpora containing spontaneous, conversational speech show that the proposed BN-BLSTM front-end leads to better ASR accuracies than previously proposed BLSTM-based Tandem and multi-stream systems.**

I. Introduction

The accuracy of systems for automatic speech recognition (ASR) heavily depends on the quality of the features extracted from the speech signal. Thus, during the last decades, a variety of methods were proposed to enhance commonly used Mel-Frequency Cepstral Coefficient (MFCC) or Perceptual Linear Prediction (PLP) features, especially in noisy conditions. A popular technique that has become state-of-the-art in modern ASR systems, is to apply a neural network to generate phoneme or phoneme state posteriors which in turn can be used as 'Tandem' features [1].

While first experiments on Tandem ASR systems concentrated on using the logarithmized and decorrelated activations of the *output* layer of recurrent neural networks (RNN) or multi-layer perceptrons (MLP) as probabilistic features, recent studies report performance gains when extracting the activations of a narrow *hidden* layer within the network as so-called 'bottleneck' (BN) features [2]. This implies the advantage that the size of the feature space can be chosen by defining the size of the network's bottleneck layer which makes the dimension of the feature vectors independent of the number of network training targets. The linear outputs of the bottleneck layer are usually well decorrelated and do not have to be logarithmized.

Since human speech is highly context-sensitive, both, the ASR front- and back-end need to account for contextual information in order to produce acceptable recognition results. Standard recognizer back-ends consider context by applying triphones, using language models, and via the Markov assumption in Hidden Markov Models (HMM) or general Graphical Models [3]. Feature-level context is usually modeled by appending derivatives of low-level features and by presenting a number of successive stacked feature frames to the neural network for Tandem feature extraction. Furthermore, the extraction of long-term features is an active area of research [4]. In Tandem systems, context can also be modeled *within* the neural network, e. g., by using recurrent connections. Studies on phoneme recognition [5] reveal that a very effective way to exploit long-range context for ASR is to apply the Long Short-Term Memory (LSTM) architecture originally introduced in [6] and extended to bidirectional LSTM (BLSTM) in [5]. LSTM overcomes the *vanishing gradient problem* of conventional RNNs and models a self-learned amount of context via memory blocks in the hidden layer.

After first successes in using BLSTM for speech-based recognition tasks such as phoneme recognition [5] and keyword spotting [7], the first system incorporating BLSTM for continuous ASR was presented in [8] and refined in [9] and [10]. In this paper, we show how bidirectional LSTM networks can be combined with the bottleneck principle to design a robust and efficient ASR front-end for context-sensitive feature extraction. We propose a novel BN-BLSTM system and evaluate it on the COSINE and the Buckeye database, which contain disfluent, spontaneous, conversational, and partly noisy speech recorded during natural conversations.

In Section II we outline the theoretical background of LSTM networks and explain previous attempts to use LSTM for continuous speech recognition. Section III continues with the description of our BN-BLSTM front-end. The applied databases are briefly introduced in Section IV before we present our experiments and results in Section V.

II. LSTM Modeling for ASR

A. Long Short-Term Memory

The analysis of the error flow in conventional recurrent neural nets resulted in the finding that long-range context is inaccessible to standard RNNs since the backpropagated error either blows up or decays over time (vanishing gradient problem [11]). Thus, only context sizes in the order of 10 frames can be captured via conventional RNNs [5]. One of the most effective techniques to overcome the vanishing gradient problem is the Long Short-Term Memory architecture [6], which is able to store information in linear memory cells over a longer period of time and can learn the optimal amount of contextual information relevant for the classification task. An LSTM hidden layer is composed of multiple recurrently connected subnets which will be referred to as *memory blocks* in the following. Every memory block consists of self-connected *memory cells* and three multiplicative *gate* units (input, output, and forget gates). Since these gates allow for write, read, and reset operations within a memory block, an LSTM block can be interpreted as (differentiable) memory chip in a digital computer.

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

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

and

$$\beta_t^{\text{in}} = f_g(\alpha_t^{\text{in}}), \quad (2)$$

respectively. The variable η^{ij} corresponds to the weight of the connection from unit i to unit j while 'in', 'for', and 'out' refer to input gate, forget gate, and output gate, respectively (see equations 3 and 7). Indices i, h, and c count the inputs x_t^i, the cell outputs from other blocks in the hidden layer, and the memory cells, while I, H, and C are the number of inputs, the number of cells in the hidden layer, and the number of memory cells in one block. Finally, s_t^c corresponds to the *state* of a cell c at time t, meaning the activation of the linear cell unit.

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

$$\alpha_t^{\text{for}} = \sum_{i=1}^{I} \eta^{i,\text{for}} x_t^i + \sum_{h=1}^{H} \eta^{h,\text{for}} \beta_{t-1}^h + \sum_{c=1}^{C} \eta^{c,\text{for}} s_{t-1}^c \quad (3)$$

$$\beta_t^{\text{for}} = f_g(\alpha_t^{\text{for}}). \quad (4)$$

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

$$\alpha_t^{\text{c}} = \sum_{i=1}^{I} \eta^{i,\text{c}} x_t^i + \sum_{h=1}^{H} \eta^{h,\text{c}} \beta_{t-1}^h. \quad (5)$$

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

$$s_t^{\text{c}} = \beta_t^{\text{for}} s_{t-1}^{\text{c}} + \beta_t^{\text{in}} f_i(\alpha_t^{\text{c}}). \quad (6)$$

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

$$\alpha_t^{\text{out}} = \sum_{i=1}^{I} \eta^{i,\text{out}} x_t^i + \sum_{h=1}^{H} \eta^{h,\text{out}} \beta_{t-1}^h + \sum_{c=1}^{C} \eta^{c,\text{out}} s_t^c \quad (7)$$

$$\beta_t^{\text{out}} = f_g(\alpha_t^{\text{out}}). \quad (8)$$

Finally, the memory cell output is determined as

$$\beta_t^{\text{c}} = \beta_t^{\text{out}} f_o(s_t^{\text{c}}). \quad (9)$$

The overall effect of the gate units is that the LSTM memory cells can store and access information over long periods of time and thus avoid the vanishing gradient problem.

B. Bidirectional LSTM

A further shortcoming of standard RNNs is that they have access to past but not to future context. This can be overcome by using *bidirectional* RNNs [12], where two separate recurrent hidden layers scan the input sequences in opposite directions. The two hidden layers are connected to the same output layer, which therefore has access to context information in both directions. For our Bottleneck-BLSTM front-end, we use a combination of the principle of bidirectional networks and the LSTM technique (i. e., bidirectional LSTM). Of course the usage of bidirectional context implies a short look-ahead buffer, meaning that recognition cannot be performed truly on-line. However, for many speech recognition tasks it is sufficient to obtain an output, e. g., at the end of an utterance, so that both, forward and backward context can be used during decoding.

C. Previous Approaches

Previous approaches towards continuous speech recognition exploiting BLSTM context-modeling concentrated on appending a discrete BLSTM feature to the (continuous) acoustic feature vector. This additional feature b_t encodes the frame-wise phoneme prediction generated via a BLSTM network, i. e., it corresponds to the index of the most active output activation which in turn corresponds to a certain phoneme at a given time step (see [8] for formulas). Applying the resulting extended feature vector $y_t = [x_t; b_t]$, that contains MFCC features x_t and the BLSTM phoneme estimate b_t, was shown to boost recognition performance of keyword detectors [7] and continuous ASR systems [8]. Further performance gains could be obtained by employing a multi-stream HMM to model x_t and b_t as two independent data streams [9] which allows to introduce different stream weights for low-level acoustic features and BLSTM phoneme predictions. Modeling long-range feature-level context via bidirectional Long Short-Term Memory could outperform simple feature frame stacking, as it is done in conventional Tandem ASR systems [10]. Thus, the application of BLSTM appears to be a promising method to generate enhanced Tandem features for speech recognition.

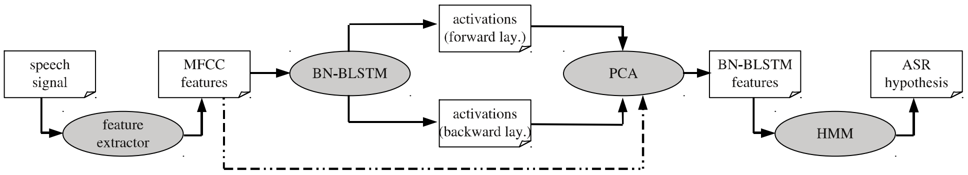

Fig. 1. Bottleneck-BLSTM front-end incorporated into an HMM-based ASR system.

III. Bottleneck-BLSTM Front-End

The Bottleneck-BLSTM feature extractor investigated in this paper can be seen as a combination of bidirectional LSTM modeling for improved context-sensitive Tandem feature generation and bottleneck front-ends. The bottleneck principle allows to generate Tandem feature vectors of arbitrary size by using the activations of the hidden (bottleneck) layer as features – rather than the logarithmized output activations corresponding to the estimated phoneme or phoneme state posteriors. 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. Figure 1 shows the system flowchart of our ASR system based on BN-BLSTM features. 39 cepstral mean and variance normalized MFCC features (including deltas and double deltas) are extracted from the speech signal every 10 ms using a window size of 25 ms. These features 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 by Principal Component Analysis (PCA). In our experiments, we evaluated feature vectors with and without the original MFCC features, which is indicated by the dashed line in Figure 1. Finally, the decorrelated BN-BLSTM features are used as input for an HMM system computing the ASR word hypothesis.

Figure 2 illustrates the detailed structure of the applied Bottleneck-BLSTM front-end. The input activations of the network correspond to the normalized MFCC features. Three hidden LSTM layers are used per input direction. As will be shown in Section V, best performance could be obtained when using a hidden layer of size 78 (two times the number of MFCC features) as first hidden LSTM layer, a second hidden layer of size 128, and a comparably narrow third hidden layer, representing the bottleneck (size 20 to 80). 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. To obtain the final decorrelated feature vectors, PCA is applied on the joint feature vectors consisting of forward and backward bottleneck layer activations and MFCC features x_t.

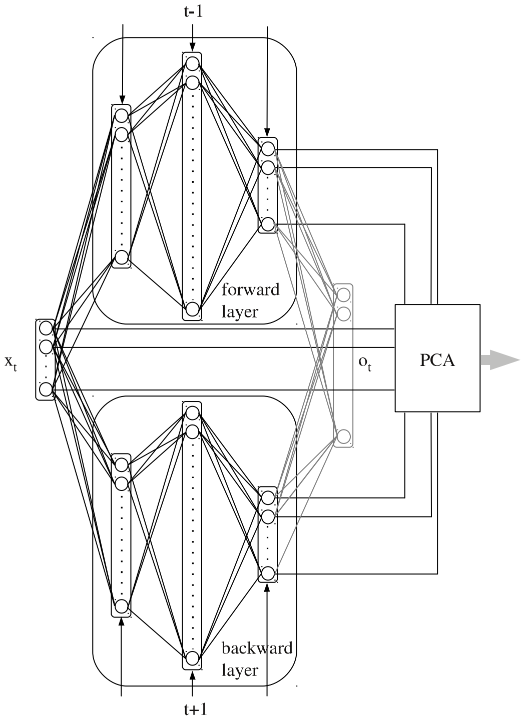

Fig. 2. Architecture of the Bottleneck-BLSTM front-end.

To compare the performance of BN-BLSTM features to probabilistic features obtained from the *output* activations of a BLSTM network (i. e., a conventional Tandem structure based on BLSTM networks) we also implemented a Tandem BLSTM system as shown in Figure 3. Since the output layer uses a softmax activation function, the BLSTM features are approximately gaussianized by conversion to the logarithmic domain before they are decorrelated via PCA. Again, Tandem feature vectors with and without appended MFCCs are evaluated. However, for the sake of better comparability, all front-ends used the same number of principal components as final feature vectors for the HMM system.

IV. Databases

In order to enable comparisons between the proposed BN-BLSTM system and previously introduced concepts for BLSTM modeling of spontaneous speech, we used the 'COnversational Speech In Noisy Environments' (COSINE) corpus

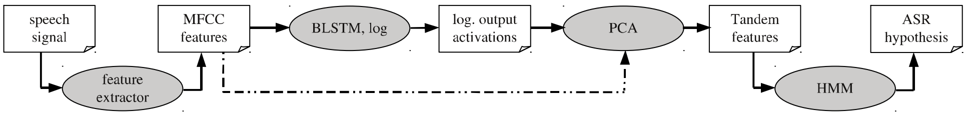

Fig. 3. Tandem BLSTM front-end incorporated into an HMM-based ASR system.

[13] which has also been used in [8], [9], and [10]. In addition, experiments with the best configurations were repeated applying the Buckeye corpus [14] to verify whether the main findings also hold for other spontaneous speech scenarios.

A. COSINE

The COSINE corpus [13] is a relatively new database containing 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).

We used all ten transcribed sessions, containing 11.40 hours of pairwise conversations and group discussions. All 37 speakers are fluent, but not necessarily native English speakers. For our experiments, we used the recommended test set (sessions 3 and 10) which comprises 1.81 hours of speech. Sessions 1 and 8 were used as validation set (2.72 h of speech) and the remaining six sessions made up the training set. The vocabulary size is 4.8 k, whereas the out-of-vocabulary (OOV) rate in the test set is 3.4 %.

B. Buckeye

The Buckeye corpus 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 [15]. Similar to the COSINE database, the contained speech is highly spontaneous. The 255 recording sessions, each of which is approximately 10 min long, were subdivided into turns by cutting whenever the subject's speech was interrupted by the interviewer, or once a silence segment of more than 0.5 s length occurred. We used the same speaker independent training, validation, and test sets as defined in [15]. The lengths of the three sets are 20.7 h, 2.4 h, and 2.6 h, respectively, and the vocabulary size is 9.1 k.

V. Experiments and Results

A. Network Training and Evaluation

For Tandem and bottleneck feature generation, we trained different recurrent and LSTM networks on framewise phoneme targets obtained via HMM-based forced alignment of the COSINE training set. We evaluated four different network types: conventional recurrent neural networks, bidirectional neural networks (BRNN), unidirectional LSTM networks, and bidirectional LSTM networks. All networks consisted of three hidden layers (per input direction) and each LSTM memory block contained one memory cell.

The networks were trained on the standard (CMU) set of 39 different English phonemes with additional targets for *silence* and *short pause*. Training was aborted as soon as no improvement on the validation set (sessions 1 and 8) could be observed for at least 50 epochs, and we chose the network that achieved the best framewise phoneme error rate on the validation set.

In conformance with [10], the three hidden layers for Tandem feature generation had a size of 78, 128, and 80, respectively. For bottleneck feature extraction, we evaluated a number of alternative network topologies that are listed in the second column of Table I, e.g., we investigated the effect of decreasing the size of the bottleneck layer from 80 to 20.

Prior to using the Tandem and bottleneck features for continuous ASR, we evaluated the framewise phoneme recognition accuracy of the underlying neural network architectures. As can be seen in the third column of Table I, the differences between the phoneme recognition rates of the various BLSTM networks are relatively small (around 70 % recognition rate for all BLSTM topologies). However, we can see that bidirectional LSTM networks perform notably better than unidirectional LSTM nets and that LSTM architectures outperform conventional RNNs.

B. Tandem Feature Extraction

Applying the trained networks, Tandem and bottleneck features were extracted according to the ASR system flowcharts depicted in Figures 1 and 3, respectively. The second column of Table I shows the sizes of five layers building the (B)LSTM and (B)RNN networks (one input, three hidden, and one output layer). The layers whose activations are used as features are indicated as numbers in boldface. For the first three BN-BLSTM configurations we employed networks containing a bottleneck layer as second hidden layer (sizes 20, 40, and 80), whereas for the remaining bottleneck experiments we

TABLE I

COSINE test set: framewise phoneme recognition rates and word accuracies for different network topologies and recognition systems processing continuous (cont.) or combined continuous-discrete (cont./disc.) Tandem features. Layer sizes in boldface indicate the layer whose activations are used as features.

		phon. accuracy	word accuracy [%]	
model architecture	**network topology**	**(framewise) [%]**	**w/o MFCC**	**w/ MFCC**
BN-BLSTM (Tandem, cont.)	39-128-**20**-128-41	69.11	43.79	44.05
BN-BLSTM (Tandem, cont.)	39-128-**40**-128-41	69.21	43.34	43.63
BN-BLSTM (Tandem, cont.)	39-128-**80**-128-41	70.12	44.95	45.86
BN-BLSTM (Tandem, cont.)	39-78-128-**20**-41	69.54	43.00	47.09
BN-BLSTM (Tandem, cont.)	39-78-128-**40**-41	69.75	43.73	49.17
BN-BLSTM (Tandem, cont.)	39-78-128-**80**-41	69.96	44.35	**49.92**
BN-LSTM (Tandem, cont.)	39-78-128-**80**-41	61.79	41.16	45.94
BN-BRNN (Tandem, cont.)	39-78-128-**80**-41	56.93	30.37	41.39
BN-RNN (Tandem, cont.)	39-78-128-**80**-41	48.88	27.01	40.74
BLSTM (Tandem, cont.)	39-78-128-80-**41**	69.96	44.41	48.23
LSTM (Tandem, cont.)	39-78-128-80-**41**	61.79	41.37	46.68
BRNN (Tandem, cont.)	39-78-128-80-**41**	56.93	30.86	40.67
RNN (Tandem, cont.)	39-78-128-80-**41**	48.88	27.97	40.14
multi-stream BLSTM-HMM (cont./disc.) [10]	39-78-128-80-**41**	69.96	-	48.01
multi-stream BLSTM-HMM (cont./disc.) [9]	39-78-128-80-**41**	66.41	-	46.50
BLSTM (Tandem, cont./disc.) [8]	39-78-128-80-**41**	66.41	-	45.04
triphone HMM	-	56.91	-	43.36

used activations of third hidden layer, focusing on the 78-128-XX hidden layer topology that was proven to give good results for LSTM-based phoneme recognition [8], [9], [10] (see also Figure 2). We found that best ASR performance can be obtained when taking only the first 39 principal components as final feature vectors. Thus, the results shown in Table I are all based on feature vectors of size 39 (except for the results taken from [8], [9], and [10], which are obtained using 39+1 features, see Section II-C).

C. Tandem ASR

The HMM system applied for processing the Tandem and BN-BLSTM features generated according to Figures 1 and 3 was identical to the back-end used to determine the baseline HMM results in [9]: 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.

D. Results

The last two columns of Table I show the word accuracies obtained for the various BLSTM-based bottleneck and Tandem systems trained and evaluated on the COSINE corpus. For each configuration better results are reached when the original MFCC features are appended to the probabilistic feature vector prior to PCA. Increasing the size of the forward and backward bottleneck BLSTM layer from 20 to 80 raises word accuracies from 44.05 to 45.86 % for the 128-XX-128 hidden layer topology and from 47.09 to 49.92 % for the 78-128-XX topology. When applying bidirectional processing, front-ends using bottleneck activations from the third hidden layer outperform Tandem systems processing the logarithmized output activations. For both front-end types RNN architectures cannot compete with LSTM architectures, which shows the importance of long-range context modeling in challenging spontaneous and disfluent speech scenarios. Best performance is reached when applying a BN-BLSTM network consisting of a comparably large third hidden layer with 80 memory blocks, representing the bottleneck (49.92 %). This system prevails over a comparable BLSTM system using continuous output activations as features (48.23 %), as well as over the best multi-stream BLSTM-HMM technique [10] applying combined continuous-discrete modeling of MFCC features and BLSTM phoneme predictions (48.01 %). For comparison, the last two rows of Table I show the performance of the continuous-discrete BLSTM Tandem system introduced in [8] (45.04 %) and the word accuracy of a baseline HMM processing only MFCC features (43.36 %).

To collect further evidence for the obtainable ASR performance gains when applying the proposed Bottleneck-BLSTM front-end, we repeated our experiments, training and evaluating the most promising network configurations on the Buckeye corpus (see Section IV-B). Since the transcriptions of the Buckeye corpus also contain the events *laughter*, *noise*, *vocal noise*, and *garbage speech*, the size of the network output layers was increased by four from 41 to 45. Thus, we also increased the size of the third hidden layer from 80 to 90 to have roughly twice as many memory blocks as phoneme targets in the last hidden layer. As shown in Table II, the baseline HMM achieves a word accuracy of 50.97 % which is comparable to the result reported in [15] (49.99 %). Accuracies for the Buckeye experiment are notably higher than for the COSINE task since the Buckeye corpus contains speech which is less disfluent and noisy than in the COSINE database. Performance can be boosted to up to 58.21 % when applying

TABLE II

BUCKEYE TEST SET: FRAMEWISE PHONEME RECOGNITION RATES AND WORD ACCURACIES FOR DIFFERENT NETWORK TOPOLOGIES AND RECOGNITION SYSTEMS PROCESSING CONTINUOUS (CONT.) OR COMBINED CONTINUOUS-DISCRETE (CONT./DISC.) TANDEM FEATURES. LAYER SIZES IN BOLDFACE INDICATE THE LAYER WHOSE ACTIVATIONS ARE USED AS FEATURES.

		phon. accuracy	**word accuracy [%]**	
model architecture	**network topology**	**(framewise) [%]**	**w/o MFCC**	**w/ MFCC**
BN-BLSTM (Tandem, cont.)	39-78-128-**90**-45	69.89	53.93	**58.21**
BN-LSTM (Tandem, cont.)	39-78-128-**90**-45	61.52	48.12	52.53
BN-BRNN (Tandem, cont.)	39-78-128-**90**-45	53.40	38.50	49.28
BN-RNN (Tandem, cont.)	39-78-128-**90**-45	47.05	35.43	48.78
BLSTM (Tandem, cont.)	39-78-128-90-**45**	69.89	55.12	57.80
LSTM (Tandem, cont.)	39-78-128-90-**45**	61.52	48.95	53.86
BRNN (Tandem, cont.)	39-78-128-90-**45**	53.40	42.11	48.64
RNN (Tandem, cont.)	39-78-128-90-**45**	47.05	39.59	48.21
multi-stream BLSTM-HMM (cont./disc.) [10]	39-78-128-90-**45**	69.89	-	56.61
BLSTM (Tandem, cont./disc.) [8]	39-78-128-90-**45**	69.89	-	55.91
triphone HMM	-	53.20	-	50.97

our BN-BLSTM feature extraction. General trends are similar to the COSINE experiment: Including MFCC features prior to PCA increases word accuracy and the Bottleneck-BLSTM principle prevails over the BLSTM multi-stream approach employed in [10].

VI. Conclusion

We proposed a novel context-sensitive feature extraction scheme employing the principle of bidirectional Long Short-Term Memory as well as the idea of bottleneck ASR front-ends. Replacing conventional MLP or RNN front-ends with BLSTM networks allows us to exploit a self-learned amount of feature-level context for accurate phoneme predictions in challenging ASR scenarios. Fusing this concept with the bottleneck technique enables the generation of a well decorrelated and compact feature space that carries information complementary to the original MFCC features. The experiments presented in this paper focused on the recognition of spontaneous, conversational, and partly disfluent, emotional, or noisy speech which usually leads to very poor ASR performance. Our BN-BLSTM technique is able to increase word accuracies from 43.36 to 49.92 % and from 50.97 to 58.21 % for the COSINE and the Buckeye task, respectively, and outperforms previous attempts to use BLSTM for continuous speech recognition as presented in a series of recent publications [8], [9], [10].

Future research should include the incorporation of delta-BN-BLSTM features and hierarchical network structures, as well as the combination of multi-stream HMMs and BN-BLSTM features. Furthermore, we plan to develop an on-line version of the proposed ASR front-end as an extension of the multi-stream ASR framework that is part of our real-time speech processing toolkit openSMILE [16].

Acknowledgment

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.

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] F. Grezl, M. Karafiat, K. Stanislav, and J. Cernocky, "Probabilistic and bottle-neck features for LVCSR of meetings," in *Proc. of ICASSP*, 2007.

[3] M. Wöllmer, F. Eyben, B. Schuller, and G. Rigoll, "Robust vocabulary independent keyword spotting with graphical models," in *Proc. of ASRU*, Merano, Italy, 2009, pp. 349–353.

[4] S. Thomas, S. Ganapathy, and H. Hermansky, "Phoneme recognition using spectral envelope and modulation frequency features," in *Proc. of ICASSP*, Taipei,Taiwan, 2009, pp. 4453–4456.

[5] 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.

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

[7] M. Wöllmer, F. Eyben, A. Graves, B. Schuller, and G. Rigoll, "Bidirectional LSTM networks for context-sensitive keyword detection in a cognitive virtual agent framework," *Cognitive Computation*, vol. 2, no. 3, pp. 180–190, 2010.

[8] 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.

[9] ——, "A multi-stream ASR framework for BLSTM modeling of conversational speech," in *Proc. of ICASSP*, Prague, Czech Republic, 2011, pp. 4860–4863.

[10] 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.

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

[12] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, pp. 2673–2681, 1997.

[13] 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.

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

[15] 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.

[16] F. Eyben, M. Wöllmer, and B. Schuller, "openSMILE - the Munich versatile and fast open-source audio feature extractor," in *Proc. of ACM Multimedia*, Firenze, Italy, 2010, pp. 1459–1462.