BLSTM-Driven Stream Fusion for Automatic Speech Recognition: Novel Methods and a Multi-Size Window Fusion Example

Timo Lohrenz, Tim Fingscheidt

Technische Universität Braunschweig
Institute for Communications Technology
Schleinitzstr. 22, 38106 Braunschweig, Germany

t.lohrenz@tu-bs.de, t.fingscheidt@tu-bs.de

Abstract

Optimal fusion of streams for ASR is a nontrivial problem. Recently, so-called posterior-in-posterior-out (PIPO-)BLSTMs have been proposed that serve as state sequence enhancers and have highly attractive training properties. In this work, we adopt the PIPO-BLSTMs and employ them in the context of stream fusion for ASR. Our contributions are the following: First, we show the positive effect of a PIPO-BLSTM as state sequence enhancer for various stream fusion approaches. Second, we confirm the advantageous context-free (CF) training property of the PIPO-BLSTM for all investigated fusion approaches. Third, we show with a fusion example of two streams, stemming from different short-time Fourier transform window lengths, that all investigated fusion approaches take profit. Finally, the turbo fusion approach turns out to be best, employing a CF-type PIPO-BLSTM with a novel iterative augmentation in training.

Index Terms: Speech recognition, Multi-size windows, Multistream-HMM, Turbo fusion, Recurrent neural networks

1. Introduction

The last decade introduced a vast variety of neural network-based methods, reducing error rates of automatic speech recognition (ASR) systems significantly. For acoustic modeling, the first successfully trained deep neural networks [1] triggered the rediscovery of convolutional neural networks [2, 3], time-delay neural networks [4, 5] and recurrent neural networks [6, 7]. All of these architectures use different strategies to incorporate temporal context into acoustic modeling or into ASR in general, which is of great importance for speech recognition, since relevant information of a spoken phoneme is distributed over a temporal span of up to half a second around a central time frame [8]. For recurrent long-short term memory (LSTM) networks that are able to use somewhat unlimited temporal context through their recurrence, it has been stated in [9] that the common use of temporal input context as spliced features is not beneficial. Moreover, recently it has been shown that BLSTMs can be effectively combined with models that indeed use large temporal context, however, a modularly trained posterior-in-posterior-out (PIPO)-BLSTM with context-free (CF) BLSTM *training* gave best results [10]. Due to the state posterior representation both at the input and output of a PIPO-BLSTM, it can then be advantageously combined with large context feature extractors in inference. This is an interesting novel property of PIPO-BLSTMs which we make use of in this work.

Optimal fusion of streams for ASR is a problem unsolved. For a jointly trained system, the common way is to simply combine different feature types at the acoustic model's input by stacking (as for example in [11]). Modular fusion approaches use posterior combinations as for example the multi-stream HMM (MSHMM) approach [12, 13], where posterior outputs of several acoustic models are simply subject to stream exponents and multiplied before decoding. The turbo fusion method [14, 15, 16] uses an iterative exchange of probabilistic information between systems to improve recognition. Further methods combine systems at output-level based on confusion network combination [17] or word hypothesis output [18].

The effectiveness of information fusion increases with the complementarity of the fused information sources. One prominent fusion task yielding robustness in noisy conditions is audiovisual speech recognition [19, 20, 21], suitable for applications that provide additional visual sensors. Fusion in *single-channel* scenarios is conducted using different feature types (e.g., magnitude and phase features [16, 22], filterbank and fMLLR features [23], and several others...) or combining a variety of multiple acoustic models [17, 24]. Another rather unexplored source of complementarity might arise from different temporal and spectral resolutions in feature extraction. ASR with short-time Fourier transform (STFT)-based features usually applies only a single window size and frame shift and is thereby quite limited. An early approach to overcome this drawback of the STFT is the use of wavelet functions [25], while a lot of recent research is focusing on the use of the raw speech signal directly as input for recognition to circumvent this drawback completely [26]. Concerning common practice in ASR, a window size of 25 ms with a frame shift of 10 ms are used, while a wider range of 15 ms to 35 ms is recommended in [27]. Recent research in [28] also hints that especially short phonemes might ineffectively be captured by the commonly used window size.

In this paper, for the first time the recently proposed modular PIPO-BLSTMs are employed as state stream enhancers in a stream fusion setup. As an example application, we investigate whether combining different window sizes improves recognition on phone level on the TIMIT task and compare several fusion methods for this particular multi-size window fusion scenario. In this paper, we focus on a comparison of fusion methods and do not strive for a new benchmark on TIMIT, which to our knowledge is reported using Li-GRU acoustic models with several effective training techniques in [23]. We conduct experiments using the context-free (CF) training strategy of PIPO-BLSTMs proposed in [10] to gain insight if it also provides benefits to fusion tasks. Finally, for turbo fusion, we introduce a new training method to PIPO-BLSTMs using augmentation with iteratively created data.

The paper is structured as follows: In Section 2, we briefly review information fusion strategies and introduce the new PIPO-BLSTM-based turbo fusion method. Section 3 describes the setup of the fusion experiments on the TIMIT phone recognition task, while corresponding results are reported and discussed in Section 4. The paper concludes in Section 5.

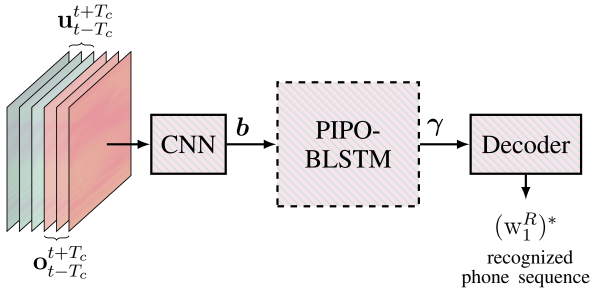

Figure 1: *Processing of the **feature combination fusion** FComb(-PIPO). Fusion is performed by combining first and second window size features (shown as red and blue layers) as a set of input channels for the convolutional input layer. Time indices t are omitted.*

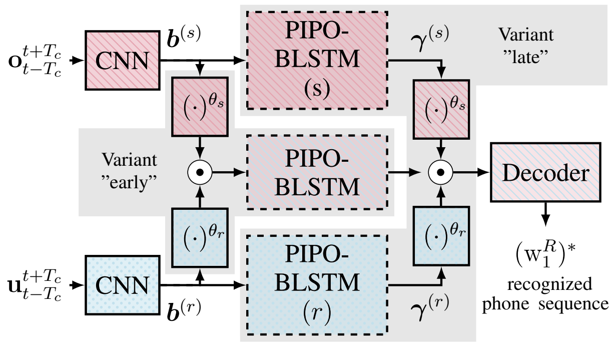

Figure 2: *Processing of the **multi-stream HMM fusion** MSHMM-PIPOe (early) or MSHMM-PIPOℓ (late).*

2. Information Fusion Approaches

2.1. Fusion by Feature Combination (FComb)

The most common and intuitive fusion method is the simple *feature combination*, usually performed by concatenating feature vectors to a joint feature representation $\mathbf{y}_t = [\mathbf{o}_t^\mathsf{T}, \mathbf{u}_t^\mathsf{T}]^\mathsf{T}$, with $(\ )^\mathsf{T}$ being the transpose. Here, two input feature streams $\mathbf{o}$ and $\mathbf{u}$, emerging from different DFT window sizes, are spliced with $T_c = 4$ frames to each side (resulting in $\mathbf{o}_{t-T_c}^{t+T_c}$ and $\mathbf{u}_{t-T_c}^{t+T_c}$). As depicted in Figure 1, for the convolutional neural network (CNN, see Section 3.3 for details) we use both feature representations together as a combined set of input channels (3 channels each: static, Δ, and $\Delta\Delta$ coefficients) at the convolutional input layer (which is possible due to the equal feature dimension d in this multi-size window fusion scenario). After the CNN, a PIPO-BLSTM (see Section 3.4 for details) is employed as a state sequence enhancer (method FComb-PIPO); in case the PIPO-BLSTM is omitted, we call the approach simply FComb. The output posteriors $\boldsymbol{\gamma}$ and (or $\boldsymbol{b}$ for the FComb approach) are then transformed into the recognized phone sequences $(\mathrm{w}_1^R)^*$ of length R by a weighted finite state transducer (WFST)-based decoder from the Kaldi toolkit [29] employing HMM topology constraints, and (for phoneme recognition) a simple phone-based language model. No hyperparameter is required, but as soon as feature representation $\mathbf{o}$ or $\mathbf{u}$ changes, CNN (and PIPO-BLSTM) need to be retrained.

2.2. Fusion by Multi-Stream HMM (MSHMM)

A second simple modular fusion method for systems with equal HMM state spaces and a synchronous frame shift is the *multi-stream HMM* (MSHMM) approach, where both input streams are now separately analyzed by two convolutional neural networks, yielding two streams of output posterior vectors $\boldsymbol{b}^{(s)}$ and $\boldsymbol{b}^{(r)}$ (indices (s) and (r) identify entities belonging to one of the streams). As shown in Figure 2, both streams of posteriors are combined with an element-wise multiplication $\odot$ after exponential weights θ_s and θ_r are applied to the individual streams [12, 13]. For the two investigated variants, either a single PIPO-BLSTM is employed after the actual fusion multiplication (early fusion variant, dubbed MSHMM-PIPO*e*) or two individual PIPO-BLSTMs (s) and (r) are used before the final fusion (ℓate fusion variant, dubbed as MSHMM-PIPOℓ. In case no PIPO-BLSTM is employed at all, we call the approach simply MSHMM. The fused posteriors are normalized per frame before decoding. The two posterior stream weights are fusion hyperparameters for all MSHMM-based methods.

2.3. Turbo Fusion

The original turbo fusion approach for speech recognition—comprehensively introduced in [14]—employs two component recognizers based on a modified forward-backward algorithm. Here we replace both component recognizers with two posterior-in-posterior-out (PIPO-)BLSTMs [10], which are well-suited for the *iterative* exchange of such posterior probabilities $\boldsymbol{\gamma}^{(s)}$ and $\boldsymbol{\gamma}^{(r)}$.

The turbo fusion approach (Turbo-PIPO), depicted in Figure 3 works as follows: Both input streams are separately analyzed by two CNNs (as for the MSHMM approaches). Considering the first PIPO-BLSTM indexed by (s), the CNN outputs $\boldsymbol{b}^{(s)}$ are combined with an additional *a priori* probability $\mathbf{g}^{(s)}$ by a simple element-wise multiplication $\boldsymbol{b}^{(s)} \odot \mathbf{g}^{(s)}$, before being normalized per frame and fed into the input layer of the PIPO-BLSTM (s). Starting with the first iteration, $\mathbf{g}^{(s)}$ is an all-one vector $\mathbf{1}$, as depicted by the switch in Figure 3, while for all following iterations the *a priori* probabilities $\mathbf{g}^{(s)}$ and $\mathbf{g}^{(r)}$ emerge from the opposite PIPO-BLSTM through the iterative loop, illustrated as green connections. After each iteration z (which we define as one call of one of the PIPO-BLSTMs), output posteriors $\boldsymbol{\gamma}^{(s)}$ and $\boldsymbol{\gamma}^{(r)}$ are subject to decoding, yielding phone sequences $(\mathrm{w}_1^{R_s})^*$ and $(\mathrm{w}_1^{R_r})^*$.

In between both PIPO-BLSTMs, two limiters employ a simple yet effective mechanism to control the amount of information in the exchanged posterior vectors as proposed in [15]. Upper and lower limits are applied to the logarithmic values of the exchanged posterior probs $\boldsymbol{\gamma}$ to weaken the impact of peaky posterior distributions and allow a less biased "discussion" (exchange of information) between both PIPO-BLSTMs. The opening of the limiters is linearly increased over iterations z towards a final dynamic range in the $z_{\max}$-th iteration which is controlled with one fusion hyperparameter for each limiter.

3. Experimental Setup

3.1. Database

To capture effects of different window sizes on phone level and to evaluate performance without the blurring influence of sophisticated language models, recognition experiments in this work are conducted on the well-known TIMIT database [30]. For training of all acoustic models, we use the 462 speaker training set with the SA-tagged dialect records being removed. Performance is reported for the standard core test set comprising 192 sentences of 24 speakers. For cross-validation during CNN and PIPO-BLSTM training, we use a separate 50 speaker development set, disjoint from the core test set. We use the complete 61 distinct TIMIT phonemes yielding $N = 183$ HMM

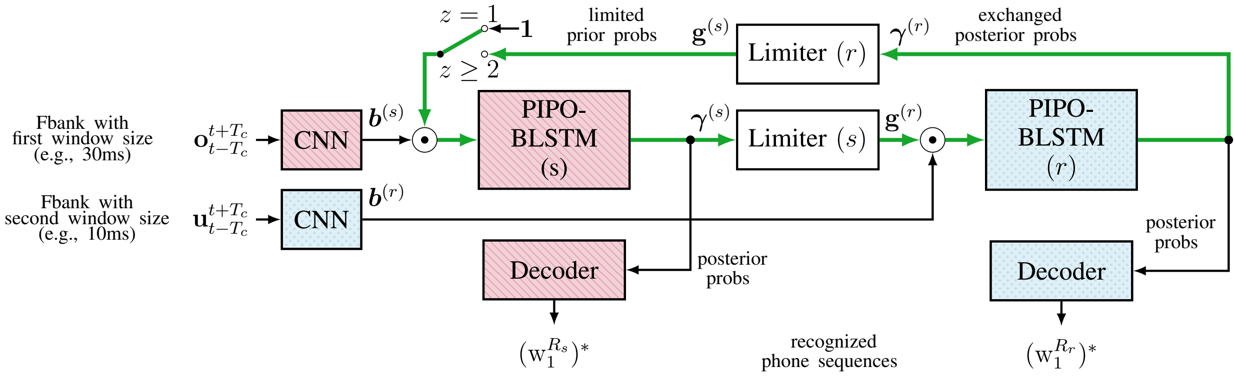

Figure 3: *Processing of the iterative* ***turbo fusion*** *with parallel streams of CNN-based acoustic posteriors $\boldsymbol{b}^{(s)}$ and $\boldsymbol{b}^{(r)}$. For the* Turbo-PIPO(IA-CF) *approaches, the output posteriors from both non-optional PIPO-BLSTMs (s) and (r) are iteratively exchanged. While the first PIPO-BLSTM (s) is active in iterations $z=1,3,5,\ldots$, and the other PIPO-BLSTM (r) is active in iterations $z=2,4,\ldots$, both process feature sequences of individual window sizes. For evaluation of each iterations phone error rate, the (identical) standard decoders are fed with the respective posterior $\boldsymbol{\gamma}^{(s)}$ or $\boldsymbol{\gamma}^{(r)}$ of that active PIPO-BLSTM. The block colors correspond to the respective DFT window sizes (e.g., red-lined pattern for the 30 ms window and blue-dotted for the 10 ms window), while for the* Baseline-PIPO(-CF) *approaches only blocks with the same color are active for one particular window size.*

states with 3 states per phoneme. For scoring, the decoded phoneme sequences $(\mathrm{w}_1^R)^*$ are merged into the smaller phone set comprising 39 phonemes, according to [31]. Based thereon, we measure recognition performance in terms of phone error rate given as $\mathrm{PER} = \left(1 - \frac{N-D-I-S}{N}\right)$ with N, D, I, S being the amount of labeled words, deletions, insertions, and substitutions, respectively. To assure comparability with common TIMIT results, we used identical settings as for example in [1, 32, 3, 33, 34, 10].

3.2. Input Streams: Multi-Size Windows

We investigate several combinations of smaller-than-standard window sizes ($\leq$ 25 ms) with larger window sizes ($\geq$ 25 ms) of the Hamming window used in the discrete Fourier transform (DFT) that analyzes the original raw speech sampled at 16 kHz. To enable a synchronous fusion of different window sizes, we used a constant frame shift of 10 ms for all window sizes, even though a variation of the frame shift might also reveal complementarity in the temporal resolution. The emerging different amounts of DFT coefficients are processed by a standard mel filterbank resulting in 40 static feature coefficients for *all* window sizes. In addition, logarithmic energy was appended as well as first- and second-order derivatives (that are treated as separate input channels for the CNN acoustic models) yielding a total amount of $d = 123$ acoustic feature coefficients $\mathbf{o}_t$ per time frame t. All input feature coefficients are normalized to zero mean and unit variance on the training set.

3.3. Acoustic Models: Convolutional Neural Networks

The employed CNNs—extracting posteriors $\boldsymbol{b}^{(s)}$ and $\boldsymbol{b}^{(r)}$ from the input features for both streams—employ limited weight sharing [35] in the convolutional input layer, dividing the 9-frame spliced input context into three blocks in the temporal direction and into seven sections along the spatial domain (please refer to [10] for a detailed illustration of this CNN). As in [36], we use a hierarchical structure, where the three input blocks are first processed seperately in the lower CNN part and are subsequently merged in an upper part with a bottleneck layer of 400 nodes, followed by 3 fully-connected layers of 1024 nodes and a standard softmax output layer, where posteriors $\boldsymbol{b}$ emerge. We employ dropout as well as batch normalization to all layers. In total, all CNNs comprise a total of 8.48 M parameters (except for the FComb approach where CNNs have 8.59 M parameters due to the larger input layer).

3.4. State Sequence Enhancement: PIPO-BLSTMs

The topology of our PIPO-BLSTMs is a simple stack of three bidirectional hidden LSTM layers with input and output layers having the same dimension of $N = 183$ context-independent phoneme HMM states. All hidden bidirectional layers employ 350 units for each direction. The layer outputs in forward and backward directions are concatenated, yielding an output of 700 units that is passed on to the subsequent layer. No peephole connections are used and we apply a dropout probability of 0.45 only to the outputs between each LSTM layer, except for the last. In total, each PIPO-BLSTM consists of 7.51M parameters.

Instead of being trained with acoustic features, PIPO-BLSTMs are trained with state posteriors, that emerge from *any* acoustic model (in our case the previously described CNNs). Due to the posterior input layer it is possible to use the PIPO-BLSTM in a modular fashion with *any* other model that has been trained in the same posterior domain as the PIPO-BLSTM. In the detailed investigation in [10] it has been shown that indeed PIPO-BLSTMs are most effective when the posteriors in training emerge from CNNs *without input context* ($T_c = 0$), and are inferred with posteriors $\boldsymbol{b}$ stemming from CNNs that indeed use *large temporal input context* (in our experiments $T_c = 4$). Approaches with PIPO-BLSTMs that include this context-free training strategy are tagged with the suffix -PIPO-CF.

Due to the iterative call of the PIPO-BLSTMs in the turbo fusion method, we can augment the training data by using the respective PIPO-BLSTM's input from all $z_{\max} = 5$ iterations on the training dataset, utilizing the fusion hyperparameters found for the Turbo-PIPO-CF approach. This *iterative augmentation* can exclusively be used for the Turbo-PIPOIA-CF approach, where we use the retrained PIPO-BLSTMs (s) and (r) with the same set of parameters during inference.

3.5. Model Training and Fusion Hyperparameters

In our experiments, CNN and PIPO models are trained separately. More precisely, PIPO-BLSTMs are trained on CNN outputs, with CNN weights fixed. The PIPO-BLSTM of the MSHMM*e*-PIPO is trained on the already fused CNN output streams. All models are trained to minimize the cross entropy (CE) loss with stochastic gradient descent learning. As ground truth we use context-independent state targets. Learning rates start at 0.1 and are halved, once the CE loss does not decrease on the TIMIT development set data. For fusion hyperparameter

Single window size baseline approaches	Development set 10ms	20ms	25ms	30ms	50ms	Avg.	Core test set 10ms	20ms	25ms	30ms	50ms	Avg.
Baseline	19.63	18.62	18.40	18.76	18.77	18.84	21.65	20.49	20.29	20.79	20.57	20.76
Baseline-PIPO	18.69	17.53	17.69	17.79	17.95	17.93	20.94	19.78	19.51	20.51	19.79	20.10
Baseline-PIPO-CF	18.42	17.23	17.53	17.63	17.88	17.74	20.40	19.79	19.50	19.88	19.61	19.83

Table 1: *Phoneme error rates (in %) for* ***baseline approaches*** *using a single window size.*

Multiple window size fusion approaches	Development set 10 ms 30 ms	10 ms 50 ms	10 ms 25 ms	20 ms 30 ms	25 ms 50 ms	Avg.	Core test set 10 ms 30 ms	10 ms 50 ms	10 ms 25 ms	20 ms 30 ms	25 ms 50 ms	Avg.
FComb	18.67	18.77	18.79	18.93	18.60	18.75	20.58	20.61	20.28	20.64	20.68	20.56
MSHMM	18.28	18.21	18.14	18.18	18.07	18.18	20.36	20.08	20.15	19.96	19.97	20.10
FComb-PIPO	17.25	17.70	17.68	17.69	17.33	17.53	19.42	19.67	19.47	19.81	19.70	19.61
MSHMMe-PIPO	17.27	17.28	17.48	17.25	17.36	17.32	19.70	19.58	19.57	19.92	19.40	19.63
MSHMMℓ-PIPO	17.06	17.08	17.27	16.98	17.13	17.10	19.54	19.06	19.27	19.56	19.07	19.30
Turbo-PIPO	17.01	16.98	17.22	16.96	17.12	17.06	19.57	18.99	19.10	19.47	19.07	19.24
FComb-PIPO-CF	16.99	17.51	17.88	17.35	17.23	17.39	19.13	19.31	19.57	19.20	18.99	19.24
MSHMMe-PIPO-CF	16.77	17.05	17.19	16.78	17.15	16.99	18.97	18.95	18.99	19.15	18.85	18.98
MSHMMℓ-PIPO-CF	17.10	17.07	17.10	16.87	16.98	17.02	19.28	18.75	18.99	19.21	19.07	19.06
Turbo-PIPO-CF	16.98	16.95	17.03	16.76	16.90	16.92	19.04	18.77	18.89	18.96	18.92	18.91
Turbo-PIPOIA-CF	**16.34**	**16.18**	**16.47**	**16.51**	**16.27**	**16.35**	**18.35**	**18.50**	**18.24**	**18.89**	**18.02**	**18.40**

Table 2: *Phoneme error rates (in %) for* ***fusion approaches*** *using multiple window sizes. All* -PIPO *approaches use the PIPO-BLSTM for fusion and the additional* -CF *suffix denotes the PIPO-BLSTM training strategy with context-free posteriors. The use of iterative training data augmentation (exclusive to turbo fusion) is dubbed* Turbo-PIPOIA-CF*. Best results are printed* ***bold***.

tuning of the MSHMM and turbo fusion approaches, we used a Bayesian optimization algorithm [37] after a quasi-random initialization of the two-dimensional search space. For the Turbo-PIPO approaches we simulated both possible successions of initial PIPO-BLSTMs ((r) or (s)) using $z_{\max} = 5$ iterations and choose the one iteration with the best PER of both successions on the development set. All models used in this paper were trained with the PyTorch toolkit [38] while acoustic filterbank features and context independent HMM state targets for training were acquired using the Kaldi toolkit [29].

4. Results and Discussion

Considering first the single-window-size baseline approaches shown in Table 1 using only CNNs for different window sizes (Baseline), our results confirm the good choice of 25 ms as the standard window size for ASR with a phoneme error rate (PER) of 20.29% on the TIMIT core test set, while most other window sizes perform only slightly worse. Using the additional PIPO-BLSTM (Baseline-PIPO) improves the performance of all window sizes by 0.66% absolute and by another 0.27% absolute on average on the test set when using the context-free training strategy (Baseline-PIPO-CF) from [10].

For fusion experiments we investigate several combinations of DFT window sizes shown in the columns of Table 2. Comparing fusion approaches *without* the involvement of BLSTM layers (1^{st} and 2^{nd} row), the MSHMM approach outperforms the FComb approach, suggesting that fusion hyperparameters on posterior-level indeed help, as a good balance of both streams appears to be difficult to learn implicitly by the FComb models. As expected, the PERs of all fusion approaches *with* the PIPO-BLSTM structure (tagged as -PIPO, 3^{rd} to 6^{th} row) decrease as the PIPO-BLSTM is able to capture more temporal information compared to the limited context ($T_C = 4$) at the CNN's input layer. Among all -PIPO approaches, MSHMMℓ-PIPO and Turbo-PIPO perform best and quite close to each other. Using the context-free PIPO-BLSTM training (-PIPO-CF, 7^{th} to 10^{th} row) we substantiate the results from [10] as all fusion methods take profit from this training strategy. Especially MSHMMe-PIPO-CF improves from the CF training with an average PER decrease of 0.65 % absolute on the test set, making it the third strongest fusion approach in this study only slightly behind the Turbo-PIPO-CF approach, which is second best among all approaches both on development and test data with average PERs of 16.92% and 18.91%, respectively. *The best overall performance is achieved by* Turbo-PIPOIA-CF*, which strongly profits from retraining the PIPO-BLSTMs with the training data augmented by the iterative PIPO-BLSTM inputs, achieving a PER improvement of 2.8% relative on the test set compared to the best of all other approaches (*Turbo-PIPO-CF*).* Compared to a single window approach (Baseline-PIPO-CF, 25 ms, 19.50%), the best PER by the 25/50 ms Turbo-PIPOIA-CF approach (**18.02**%) is a remarkable PER decrease of 8.2% relative on the test data.

5. Conclusion

In this contribution we investigate several stream fusion approaches on a multi-size window fusion example. We show that the recently proposed posterior-in-posterior-out (PIPO-)BLSTM state sequence enhancer provides benefit to all fusion approaches, especially when they are trained on (input) context-free feature extractor networks. The fusion approach that profits the most from the PIPO-BLSTM is turbo fusion that is best among all other approaches. Utilizing a novel training strategy, where PIPO-BLSTMs are trained with iteratively gathered data, the turbo fusion outperforms the best single-window setup by 8.2% relative.

ACKNOWLEDGMENTS

The research leading to these results has received funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) for project number 414091002.

6. References

[1] A. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic Modeling Using Deep Belief Networks," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 1, pp. 14–22, Jan. 2012.

[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[3] O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional Neural Networks for Speech Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 10, pp. 1533–1545, Oct. 2014.

[4] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme Recognition Using Time-Delay Neural Networks," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 3, pp. 328–339, Mar. 1989.

[5] V. Peddinti, D. Povey, and S. Khudanpur, "A Time Delay Neural Network Architecture for Efficient Modeling of Long Temporal Contexts," in *Proc. of Interspeech*, Dresden, Germany, Sep. 2015, pp. 3214–3218.

[6] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[7] A. Graves, A. Mohamed, and G. Hinton, "Speech Recognition With Deep Recurrent Neural Networks," in *Proc. of ICASSP*, Vancouver, BC, Canada, May 2013, pp. 6645–6649.

[8] N. Kanedera, T. Arai, H. Hermansky, and M. Pavel, "On the Importance of Various Modulation Frequencies for Speech Recognition," in *Proc. of Eurospeech*, Rhodes, Greece, Sep. 1997, pp. 1079–1082.

[9] 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.

[10] T. Lohrenz, M. Strake, and T. Fingscheidt, "On Temporal Context Information for Hybrid BLSTM-Based Phoneme Recognition," in *Proc. of ASRU*, Singapore, Singapore, Dec. 2019, pp. 516–523.

[11] G. Potamianos, J. Luettin, and C. Neti, "Hierarchical Discriminant Features for Audio-Visual LVCSR," in *Proc. of ICASSP*, Salt Lake City, UT, USA, May 2001, pp. 165–168.

[12] J. Luettin, G. Potamianos, and C. Neti, "Asynchronous Stream Modeling for Large Vocabulary Audio-Visual Speech Recognition," in *Proc. of ICASSP*, Salt Lake City, UT, USA, May 2001, pp. 169–172.

[13] H. Misra, H. Bourlard, and V. Tyagi, "New Entropy Based Combination Rules in HMM/ANN Multi-Stream ASR," in *Proc. of ICASSP*, vol. 2, Hong Kong, China, Apr. 2003, pp. 741–744.

[14] S. Receveur, R. Weiss, and T. Fingscheidt, "Turbo Automatic Speech Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 5, pp. 846–862, May 2016.

[15] T. Lohrenz and T. Fingscheidt, "Turbo Fusion of Magnitude and Phase Information for DNN-Based Phoneme Recognition," in *Proc. of ASRU*, Okinawa, Japan, Dec. 2017, pp. 118–125.

[16] T. Lohrenz, W. Li, and T. Fingscheidt, "A New TIMIT Benchmark for Context-Independent Phone Recognition Using Turbo Fusion," in *Proc. of SLT*, Athens, Greece, Dec. 2018, pp. 498–505.

[17] W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, and A. Stolcke, "The Microsoft 2017 Conversational Speech Recognition System," *arXiv:1708.06073*, Aug. 2017.

[18] B. Hoffmeister, T. Klein, R. Schlüter, and H. Ney, "Frame Based System Combination and a Comparison With Weighted ROVER and CNC," in *Proc. of Interspeech*, Pittsburgh, PA, USA, Sep. 2006, pp. 537–540.

[19] G. Potamianos, C. Neti, G. Gravier, A. Garg, and A. W. Senior, "Recent Advances in the Automatic Recognition of Audiovisual Speech," *Proceedings of the IEEE*, vol. 91, no. 9, pp. 1306–1326, Sep. 2003.

[20] S. Receveur, D. Scheler, and T. Fingscheidt, "A Turbo-Decoding Weighted Forward-Backward Algorithm for Multimodal Speech Recognition," in *Situated Dialog in Speech-Based Human-Computer Interaction*, A. Rudnicky, A. Raux, A. Lane, and I. Misu, Eds. Springer-Verlag, 2016, pp. 179–192.

[21] T. Afouras, J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, "Deep Audio-Visual Speech Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–11, 2018.

[22] R. M. Hegde, H. A. Murthy, and V. R. R. Gadde, "Significance of the Modified Group Delay Feature in Speech Recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 1, pp. 190–202, Jan. 2007.

[23] M. Ravanelli, T. Parcollet, and Y. Bengio, "The PyTorch-Kaldi Speech Recognition Toolkit," in *Proc. of ICASSP*, Brighton, United Kingdom, May 2019, pp. 6465–6469.

[24] J. Du *et al.*, "The USTC-iFlytek Systems for CHiME-5 Challenge," in *CHiME 2018 Workshop on Speech Processing in Everyday Environments*, Hyderabad, India, Sep. 2018, pp. 11–15.

[25] B. T. Tan, M. Fu, and P. Dermody, "The Use of Wavelet Transforms in Phoneme Recognition," in *Proc. of ICSLP*, vol. 4, Philadelphia, PA, USA, Oct. 1996, pp. 2431–2434.

[26] T. Sainath, R. J. Weiss, K. Wilson, A. W. Senior, and O. Vinyals, "Learning the Speech Front-End With Raw Waveform CLDNNs," in *Proc. of Interspeech*, Dresden, Germany, Sep. 2015, pp. 1–5.

[27] K. Paliwal, J. Lyons, and K. Wojcicki, "Preference for 20-40 ms Window Duration in Speech Analysis," in *Proc. of ICSPCS*, Gold Coast, QLD, Australia, Dec. 2010, pp. 1–4.

[28] D. T. Toledano, M. P. Fernández-Gallego, and A. Lozano-Diez, "Multi-Resolution Speech Analysis for Automatic Speech Recognition Using Deep Neural Networks: Experiments on TIMIT," *PLOS ONE*, vol. 13, no. 10, pp. 1–24, Oct. 2018.

[29] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The Kaldi Speech Recognition Toolkit," in *Proc. of ASRU*, Waikoloa, HI, USA, Dec. 2011, pp. 1–4.

[30] "The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus CD-ROM," National Institute of Standards and Technology (NIST), Oct. 1990.

[31] K. F. Lee and H. W. Hon, "Speaker-Independent Phone Recognition Using Hidden Markov Models," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 11, pp. 1641–1648, Nov. 1989.

[32] A. Graves, N. Jaitly, and A. Mohamed, "Hybrid Speech Recognition With Deep Bidirectional LSTM," in *Proc. of ASRU*, Olomouc, Czech Republic, Dec. 2013, pp. 273–278.

[33] J. Deng, Z. Zhang, F. Eyben, and B. Schuller, "Autoencoder-Based Unsupervised Domain Adaptation for Speech Emotion Recognition," *IEEE Signal Processing Letters*, vol. 21, no. 9, pp. 1068–1072, Sep. 2014.

[34] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, Jun. 2014.

[35] O. Abdel-Hamid, A. Mohamed, H. Jiang, and G. Penn, "Applying Convolutional Neural Networks Concepts to Hybrid NN-HMM Model for Speech Recognition," in *Proc. of ICASSP*, Kyoto, Japan, Mar. 2012, pp. 4277–4280.

[36] L. Tóth, "Phone Recognition With Hierarchical Convolutional Deep Maxout Networks," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2015, no. 1, pp. 1–13, Sep. 2015.

[37] J. Snoek, H. Larochelle, and R. Adams, "Practical Bayesian Optimization of Machine Learning Algorithms," in *Proc. of NIPS*, Lake Tahoe, NV, USA, Dec. 2012, pp. 2951–2959.

[38] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Proc. of NeurIPS*, Vancouver, BC, Canada, Dec. 2019, pp. 8024–8035.