INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Improving English Conversational Telephone Speech Recognition

Ivan Medennikov[1,2], *Alexey Prudnikov*[2,3], *Alexander Zatvornitskiy*[1,2,3]

[1]STC-innovations Ltd, St. Petersburg, Russia
[2]ITMO University, St. Petersburg, Russia
[3]Speech Technology Center Ltd, St. Petersburg, Russia
{medennikov,prudnikov,zatvornitskiy}@speechpro.com

Abstract

The goal of this work is to build a state-of-the-art English conversational telephone speech recognition system. We investigated several techniques to improve acoustic modeling, namely speaker-dependent bottleneck features, deep Bidirectional Long Short-Term Memory (BLSTM) recurrent neural networks, data augmentation and score fusion of DNN and BLSTM models. Training set consisted of the 300 hour Switchboard English speech corpus. We also examined the hypothesis rescoring using language models based on recurrent neural networks. The resulting system achieves a word error rate of 7.8% on the Switchboard part of the HUB5 2000 evaluation set which is the competitive result.

Index Terms: conversational telephone speech recognition, deep neural networks, recurrent neural networks

1. Introduction

English conversational telephone speech (CTS) recognition systems are becoming better and better each year. This is caused by a large number of studies carried out on the Switchboard English task, such as [1–7]. In recent years major improvement of English CTS recognition systems has been obtained by the use of the techniques listed below. First, acoustic models (AM) based on deep neural networks (DNN) significantly outperformed Gaussian mixture models (GMM) [1]. Sequence-discriminative training of DNN acoustic models [2] also led to substantial recognition accuracy improvement. Second, applying acoustic models based on convolutional neural networks or recurrent neural networks in combination with DNN acoustic models showed high effectiveness. Last but not least, sophisticated language models (LM) based on feedforward or recurrent neural networks demonstrated their superiority over n-gram language models.

So, the state-of-the-art results in terms of word error rate (WER) on the Switchboard subset of the HUB5 2000 evaluation set were improved from about 16% in 2011 to about 12% in 2013, 10.4% in 2014 and 8% in 2015. The impressive WER of 8% reported by IBM researchers [6] is not too far from the human word error rate on the Switchboard English CTS recognition task, which was estimated to be around 4% in [8].

In this work we present the study on building a state-of-the-art English CTS recognition system. We used the approach of finding and investigating the effective techniques and combining them. The resulting system achieves the competitive results on the HUB5 2000 evaluation set: 7.8% WER on the Switchboard subset (which is the state-of-the-art result at the moment as far as we know) and 16.0% WER on the CallHome subset.

The rest of this paper is organized as follows. Section 2 presents the investigation of several techniques of acoustic modeling improvement, namely speaker-dependent bottleneck features, deep BLSTM acoustic models, data augmentation and score fusion of DNN and BLSTM acoustic models. Section 3 describes the experiments on hypothesis rescoring with RNN-based language models. Finally, Section 4 concludes the paper and discusses future work.

2. Acoustic modeling

In this section we study several acoustic modeling techniques which are perspective for improving English CTS recognition. All experiments were performed on Switchboard-1 Release 2 (LDC97S62) training set. We report results in terms of word error rate on both Switchboard and CallHome subsets of the HUB5 2000 evaluation set.

2.1. Speaker-dependent bottleneck features

Bottleneck features are widely used in ASR systems [9, 10]. Here we present the acoustic modeling approach based on speaker-dependent bottleneck (SDBN) features. This approach was proposed in our previous work [11] for Russian spontaneous speech recognition and demonstrated high effectiveness. The idea is to extract high-level features from DNN model, which is adapted to the speaker and acoustic environment by the use of i-vectors. The extracted features are applied to training another acoustic model (see Figure 1).

Our approach consists of the following main steps:

1. Training the DNN model on the source features using the Cross-Entropy (CE) criterion.
2. Expanding an input layer of the DNN trained at the first step and retraining using input feature vector appended with i-vector. The regularizing term

$$R = \lambda \sum_{l=1}^{L} \sum_{i=1}^{N_l} \sum_{j=1}^{N_{l-1}} (\mathbf{W}_{ij}^{l} - \bar{\mathbf{W}}_{ij}^{l})^2 \qquad (1)$$

   is added to the CE criterion for penalizing parameters deviation from the source model. Here $\mathbf{W}^l$ and $\bar{\mathbf{W}}^l$ are weight matrices of l-th layer ($1 \leq l \leq L$) of the current and the source DNNs, N_l is the size of l-th layer, and N_0 is the dimension of the input feature vector.
3. Transforming the last hidden layer into two layers. The first one is a bottleneck layer with weight matrix $\mathbf{W}_{bn}$, zero bias vector and linear activation function. The second one is a non-linear layer with the dimension of the source layer, with weight matrix $\mathbf{W}_{out}$ and the original

http://dx.doi.org/10.21437/Interspeech.2016-473

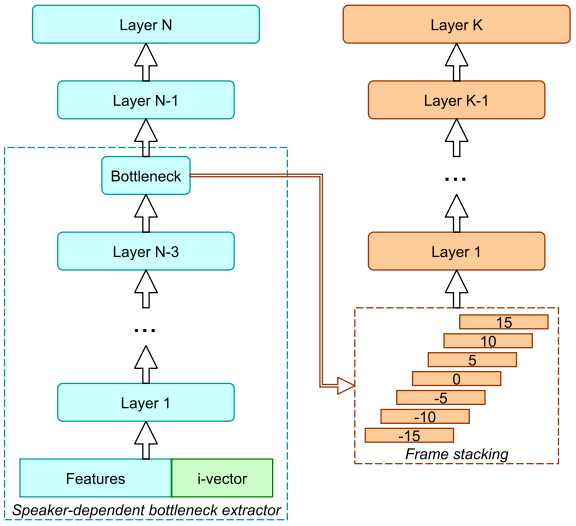

Figure 1: *Speaker-dependent bottleneck approach scheme*

bias vector $\mathbf{b}$, activation function f and the dimension of the source layer.

$$\mathbf{y} = f(\mathbf{W}\mathbf{x} + \mathbf{b}) \approx f(\mathbf{W}_{out}(\mathbf{W}_{bn}\mathbf{x} + \mathbf{0}) + \mathbf{b}). \quad (2)$$

These layers are formed by applying Singular Value Decomposition (SVD) to the weight matrix $\mathbf{W}$ of the source layer:

$$\mathbf{W} = \mathbf{U}\mathbf{S}\mathbf{V}^T \approx \tilde{\mathbf{U}}_{bn}\tilde{\mathbf{V}}_{bn}^T = \mathbf{W}_{out}\mathbf{W}_{bn}, \quad (3)$$

where $_{bn}$ designates reduced dimension.

4. Retraining the network formed at the previous step using the CE criterion with the penalty (1) for parameters deviation from original values.
5. Discarding all layers after the bottleneck and extracting high-level SDBN features using the resulting DNN.
6. Training the GMM-HMM acoustic model using the constructed SDBN features and generating the senone alignment of the training data.
7. Training the final DNN-HMM acoustic model using SDBN features and the generated alignment.

For experiments we used the Kaldi speech recognition toolkit [12], which contains the recipe for the Switchboard task. The performance of models was evaluated on the Switchboard part of HUB5 2000 evaluation set. In this experiment we used 3-gram language model (750K n-grams, vocabulary of 30.3K words) from the Kaldi recipe. This model was trained on the transcriptions of the Switchboard corpus only.

The DNN-HMM model from this recipe (local/run_dnn.sh) [2] was considered to be a baseline (DNN-baseline). This DNN with 6 hidden layers with 2048 sigmoidal neurons in each and the output softmax layer with about 9000 neurons was trained using 11 spliced 40-dimensional fMLLR-adapted features and state-level Minimum Bayes Risk (sMBR) sequence-discriminative criterion.

80-dimensional SDBN features were constructed using the presented approach. We applied 100-dimensional i-vectors extracted by the use of Universal Background Model with 512 Gaussian, which was trained with our toolset [13] on the full Switchboard corpus. DNN training with the constructed SDBN features (SDBN-DNN) was performed using the temporal context of 31 frames taking every 5th frame. We applied the following DNN configuration: 4 sigmoidal hidden layers with 2048 neurons in each, the output softmax layer with about 9000 neurons corresponding to senones of the GMM-HMM model, which was trained using the same SDBN features. The training was carried out with the sMBR criterion. For the comparison, we also performed sMBR training of the speaker-adapted with i-vectors DNN model (DNN-ivec). The results given in Table 1 demonstrate effectiveness of the presented approach.

Table 1: *Speaker-dependent bottleneck approach results on the HUB5 2000 evaluation set*

Acoustic model	SWB WER, %	CH WER, %
DNN-baseline	12.9	24.5
DNN-ivec	12.5 (-0.4)	24.2 (-0.3)
SDBN-DNN	12.1 (-0.8)	23.3 (-1.2)

2.2. Bidirectional Long Short-Term Memory recurrent neural networks

Acoustic models based on deep Bidirectional Long Short-Term Memory (BLSTM) recurrent neural networks demonstrate high effectiveness in various ASR tasks [7, 14, 15]. In this subsection we describe our experiments with these models carried out with *nnet3* setup of the Kaldi speech recognition toolkit.

We used BLSTM architecture with projection layers described in paper [16]. The following configuration of the network was applied: 3 forward and 3 backward layers, cell and hidden dimensions are 1024, recurrent and non-recurrent projection dimensions are 128, input features are taken with the temporal context of 5 frames. Training examples consisted of chunks of 20 frames with additional left context of 40 frames and right context of 40 frames. We performed 8 epochs of cross-entropy training with initial learning rate of 0.0003 and final learning rate of 0.00003. Model parameters were updated using BPTT algorithm with the momentum value equal to 0.5.

We tried a few input features configurations and chose 23-dimensional log mel filterbank energy (FBANK) features. First, we found that training data alignments prepared using SDBN-DNN acoustic model provide substantial improvement compared with GMM-derived alignments. Second, cepstral mean normalization (CMN) of input features granted an additional improvement of the acoustic model. Third, we applied speaker adaptation of BLSTM acoustic model using i-vectors and obtained significant WER reduction. Lastly, the resulting BLSTM was retrained using a wider chunk (80 frames) and the same left and right contexts as used before. The retraining provided a substantial gain, we suppose this is due to the better network performance on longer sequences.

The main results of the experiments are summarized in Table 2.

2.3. Data augmentation

For further improvement of our acoustic models, we tried the data augmentation approach presented in [17]. Two additional copies of the training data were created by modifying the speed to 90% and 110% of the original speed. The alignments for the speed perturbed data were generated using SDBN-DNN

Table 2: *BLSTM results on the HUB5 2000 evaluation set*

Acoustic model	SWB WER, %	CH WER, %
baseline BLSTM	12.6	23.8
+DNN alignment	12.2 (-0.4)	22.6 (-1.2)
+CMN	12.1 (-0.5)	21.7 (-2.1)
+i-vectors	11.3 (-1.3)	21.4 (-2.4)
+retraining	11.1 (-1.5)	20.9 (-2.9)

acoustic model from subsection 2.1. We applied the augmentation of the training data for both SDBN-DNN and BLSTM acoustic models. For BLSTM model, we also applied volume perturbation of the training data [18]: each recording was scaled with a factor chosen randomly in range $\left[\frac{1}{8}, 2\right]$.

As can be seen in Table 3, data augmentation provided a considerable gain on the HUB5 2000 evaluation set. Note

Table 3: *Data augmentation results on the HUB5 2000 evaluation set*

Acoustic model	SWB WER, %	CH WER, %
SDBN-DNN	12.1	23.3
SDBN-DNN + augm	11.8 (-0.3)	22.5 (-0.8)
BLSTM	11.1	20.9
BLSTM + augm	10.8 (-0.3)	20.4 (-0.5)

that for SDBN-DNN model we did not retrain the bottleneck extractor with the augmented data.

2.4. Score fusion of SDBN-DNN and BLSTM acoustic models

Score fusion of acoustic models is a well known technique. Its underlying idea is in combining the benefits of both different model architectures and different input features. In this subsection we analyze effectiveness of this technique applied to SDBN-DNN and BLSTM acoustic models. We used log-likelihoods (LLH) determined by the formula

$$\text{LLH} = \alpha \log\left(\frac{\text{P}_1(\mathbf{s}|\mathbf{x})}{\text{P}_1(\mathbf{s})}\right) + (1-\alpha)\log\left(\frac{\text{P}_2(\mathbf{s}|\mathbf{x})}{\text{P}_2(\mathbf{s})}\right) \quad (4)$$

for the decoding with fusion of these acoustic models. Here $\text{P}_1(\mathbf{s}|\mathbf{x})$ and $\text{P}_2(\mathbf{s}|\mathbf{x})$ are posterior probabilities of state $\mathbf{s}$ given input vector $\mathbf{x}$ on the current frame, $\text{P}_1(\mathbf{s})$ and $\text{P}_2(\mathbf{s})$ are prior probabilities of state $\mathbf{s}$ for SDBN-DNN and BLSTM models respectively. We estimated prior probability of state $\mathbf{s}$ as average posterior probability calculated with the corresponding model on the training data. α value was chosen equal to 0.5. The results of the experiments are given in Table 4. One can see the

Table 4: *Score fusion results on the HUB5 2000 evaluation set*

Acoustic model	SWB WER, %	CH WER, %
SDBN-DNN + augm	11.8	22.5
BLSTM + augm	10.8	20.4
score fusion	9.9 (-0.9)	18.9 (-1.5)

significant WER improvement obtained by the score fusion of SDBN-DNN and BLSTM acoustic models.

3. Language modeling

In this section we describe the experiments with language models. Word lattices obtained on the decoding pass with 3-gram LM and the best DNN+BLSTM models fusion in subsection 2.4 were taken as a starting point for these experiments.

At the first stage, we applied lattice rescoring with the 4-gram language model (4.7M n-grams) from the Kaldi recipe. 4-gram LM was obtained by the linear interpolation of 4-gram models trained on the transcriptions of Switchboard and Fisher corpora. This LM had the same vocabulary as the 3-gram model used in our previous experiments.

We also built two neural network LMs (NNLMs). We took utterances from the transcriptions of Switchboard and Fisher corpora, shuffled them and replaced Out-Of-Vocabulary words with <UNK> token. These utterances were divided into two parts: a valid set (20K utterances) and a train set (all other, about 2.5M utterances). The transcriptions of the HUB5 2000 evaluation set were used as a test set.

Table 5: *Perplexity results on the train, valid and test data*

Language model	PPL train	PPL valid	PPL test
4-gram (baseline)	66.366	62.946	87.039
RNNLM	57.982	78.578	76.123
LSTM-LM (medium)	51.104	58.964	56.822
LSTM-LM (large)	46.033	54.821	52.892

The first NNLM was Recurrent Neural Network Language Model (RNNLM) [19]. It was shown that RNNLM significantly outperforms n-gram LM in various speech recognition tasks. In particular, the results demonstrated by RNNLM in the English CTS recognition task can be found in the paper [20]. We trained our model using Mikolov's RNNLM Toolkit [21]. We applied the following RNNLM configuration: 256 neurons in the hidden layer, 4×200 MB of direct connections. To speed-up the training we used the factorized output layer with 200 classes.

Table 6: *Rescoring results on the HUB5 2000 evaluation set*

Language model	SWB WER, %	CH WER, %
3-gram (SWB)	9.9	18.9
4-gram (SWB+FSH)	9.1 (-0.8)	17.6 (-1.3)
RNNLM	8.4 (-1.5)	16.8 (-2.1)
LSTM-LM (medium)	8.0 (-1.9)	16.2 (-2.7)
LSTM-LM (large)	7.8 (-2.1)	16.0 (-2.9)

The second NNLM was LSTM recurrent neural network LM (LSTM-LM) trained with dropout regularization [22]. This model demonstrated state-of-the-art results in terms of perplexity (PPL) on the English Penn Treebank data set.

The architecture of this LSTM-LM model with L layers is given by the following equations [22]:

$$\text{LSTM} : h_t^{l-1}, h_{t-1}^l, c_{t-1}^l \rightarrow h_t^l, c_t^l, \quad (5)$$

$$\begin{pmatrix} i_t^l \\ f_t^l \\ o_t^l \\ g_t^l \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} T_{2n,4n} \begin{pmatrix} \text{D}(h_t^{l-1}) \\ h_{t-1}^l \end{pmatrix}, \quad (6)$$

$$c_t^l = f_t^l \odot c_{t-1}^l + i_t^l \odot g_t^l, \quad (7)$$

$$h_t^l = o_t^l \odot \tanh(c_t^l). \quad (8)$$

Table 7: *WER comparison with existing English CTS recognition systems on the HUB5 2000 evaluation set*

System	AM training data	LM training data	SWB	CH
Vesely et al. [2]	SWB	SWB,FSH-1	12.6	24.1
Hannun et al. [5]	SWB,FSH	SWB,FSH	12.6	19.3
Peddinti et al. [18]	SWB	SWB,FSH	11.0	—
Soltau et al. [4]	SWB	SWB,FSH	10.4	19.1
Mohamed et al. [7]	SWB,FSH,other	SWB,FSH,other	9.2	—
Saon et al. [6]	SWB,FSH,CH	SWB,FSH,CH	8.0	14.1
This system	SWB	SWB,FSH	7.8	16.0

Here $h_t^l, c_t^l, i_t^l, f_t^l, o_t^l, g_t^l \in \mathbb{R}^n$ denote hidden state, memory cell state and the activations of input gate, forget gate, output gate and input modulation gate in layer $l \in [1, L]$ at time t, respectively; $h_t^0 \in \mathbb{R}^n$ is an input word vector at time t; $T_{2n,4n} : \mathbb{R}^{2n} \to \mathbb{R}^{4n}$ is a linear transform with a bias; D is the dropout operator that sets a random subset of its argument to zero; symbol $\odot$ denotes element-wise multiplication. Logistic (sigm) and hyperbolic tangent (tanh) activation functions in these equations are applied element-wise. Activations $h_t^L \in \mathbb{R}^n$ are used to predict the word at time t.

We used the Tensorflow toolkit [23] to train this model. We trained two LSTM-LMs: "medium" (2 layers with 650 units each, 50% dropout on the non-recurrent connections) and "large" (2 layers with 1500 units each, 65% dropout on the non-recurrent connections) configurations from the paper [22]. For the "large" model forget gate biases were initialized with value of 1.0. Training on NVIDIA GTX Titan X GPU took 40 hours for the "medium" network and 146 hours for the "large" one.

The perplexity values of these LMs on the train, valid and test data are given in Table 5. Note that valid PPL of the baseline 4-gram model is low due to the presence of valid texts in the training data for this LM.

Both the trained NNLMs were applied for the hypothesis rescoring. We generated 100-best lists from the 4-gram rescored lattices using Kaldi scripts. For the rescoring we took the weighted sum of n-gram LM and NNLM scores. The results of the rescoring are given in Table 6. It can be seen that RNNLM provided substantial improvement over n-gram LM, as well as LSTM-LM over RNNLM.

4. Discussion

The architecture of our system is depicted in Figure 2. In Table 7 we present the results of comparison with existing English CTS recognition systems. For clarity, we also specify the data used for training acoustic and language models for each system. Our system achieves the competitive results on the HUB5 2000 evaluation set: 7.8% WER on the Switchboard part (which is the state-of-the-art result at the moment as far as we know) and 16.0% WER on the CallHome part. Note that acoustic models used in the system were trained only on the 300 hour Switchboard English CTS corpus.

We consider several ways of further improvement of our system. First, a great accuracy gain can be obtained by adding Fisher and CallHome corpora into the AM training set. Second, sequence-discriminative training of BLSTM acoustic models can lead to substantial WER reduction [24]. Third, retraining the SDBN extractor with the augmented data can provide additional improvement. Last but not least, we plan to carry out experiments with other promising language model architectures such as Character-Aware Neural Language Models [25], End-To-End Memory Networks [26] and others. We are going to investigate more complicated approaches of applying sophisticated language models than simple n-best rescoring as well.

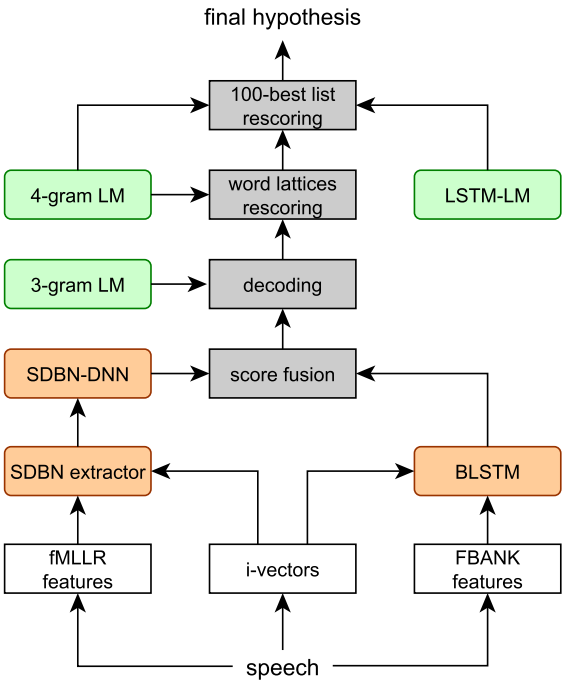

Figure 2: *System architecture*

5. Acknowledgements

The work was partially financially supported by the Government of the Russian Federation, Grant 074-U01.

6. References

[1] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2011.

[2] K. Vesely, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2013.

[3] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors,"

Proc. Automatic Speech Recognition and Understanding (ASRU), pp. 55–59, 2013.

[4] H. Soltau, G. Saon, and T. Sainath, "Joint training of convolutional and non-convolutional neural networks," *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[5] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Ng, "Deep Speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[6] G. Saon, H.-K. Kuo, S. Rennie, and M. Picheny, "The IBM 2015 english conversational telephone speech recognition system," *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2015.

[7] A. Mohamed, F. Seide, D. Yu, J. Droppo, A. Stolcke, G. Zweig, and G. Penn, "Deep bi-directional recurrent networks over spectral windows," *Proc. Automatic Speech Recognition and Understanding (ASRU)*, pp. 78–83, 2015.

[8] R. P. Lippmann, "Speech recognition by machines and humans," *Speech communication*, vol. 22, no. 1, pp. 1–15, 1997.

[9] F. Grezl, M. Karafiat, S. Kontar, and J. Cernocky, "Probabilistic and bottle-neck features for LVCSR of meetings," *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 4, pp. 757–760, 2007.

[10] J. Gehring, Y. Miao, F. Metze, and A. Waibel, "Extracting deep bottleneck features using stacked auto-encoders," *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3377–3381, 2013.

[11] A. Prudnikov, I. Medennikov, V. Mendelev, M. Korenevsky, and Y. Khokhlov, "Improving acoustic models for russian spontaneous speech recognition," *Speech and Computer, Lecture Notes in Computer Science*, vol. 9319, pp. 234–242, 2015.

[12] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 1–4, 2011.

[13] A. Kozlov, O. Kudashev, Y. Matveev, T. Pekhovsky, K. Simonchik, and A. Shulipa, "SVID speaker recognition system for NIST SRE 2012," *Speech and Computer, Lecture Notes in Computer Science*, vol. 8113, pp. 278–285, 2013.

[14] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," *Proc. ICML*, pp. 1764–1772, 2014.

[15] H. Sak, A. Senior, K. Rao, and F. Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 1468–1472, 2015.

[16] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2014.

[17] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2015.

[18] V. Peddinti, D. Povey, and K. S., "A time delay neural network architecture for efficient modeling of long temporal contexts," *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2015.

[19] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky, and S. Khudanpur, "Recurrent neural network based language model," *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 1045–1048, 2010.

[20] Y. Shi, W.-Q. Zhang, M. Cai, and J. Liu, "Empirically combining unnormalized NNLM and back-off n-gram for fast n-best rescoring in speech recognition," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 19, 2014.

[21] T. Mikolov, S. Kombrink, A. Deoras, L. Burget, and J. Cernocky, "RNNLM — recurrent neural network language modeling toolkit," *ASRU 2011 Demo Session*, 2011.

[22] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv preprint arXiv:1409.2329*, 2014.

[23] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/

[24] H. Sak, O. Vinyals, G. Heigold, A. Senior, E. McDermott, R. Monga, and M. Mao, "Sequence discriminative distributed training of long short-term memory recurrent neural networks," *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2014.

[25] Y. Kim, Y. Jernite, D. Sontag, and A. Rush, "Character-aware neural language models," *arXiv preprint arXiv:1508.06615*, 2015.

[26] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, "End-to-end memory networks," *arXiv preprint arXiv:1503.08895*, 2015.