INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Improved Single System Conversational Telephone Speech Recognition with VGG Bottleneck Features

William Hartmann, Roger Hsiao, Tim Ng, Jeff Ma, Francis Keith, Man-Hung Siu

Raytheon BBN Technologies, Cambridge, MA, USA
{whartman,jma,fkeith,msiu}@bbn.com, {hsiao06,estim.ng}@gmail.com

Abstract

On small datasets, discriminatively trained bottleneck features from deep networks commonly outperform more traditional spectral or cepstral features. While these features are typically trained with small, fully-connected networks, recent studies have used more sophisticated networks with great success. We use the recent deep CNN (VGG) network for bottleneck feature extraction—previously used only for low-resource tasks—and apply it to the Switchboard English conversational telephone speech task. Unlike features derived from traditional MLP networks, the VGG features outperform cepstral features even when used with BLSTM acoustic models trained on large amounts of data. We achieve the best BBN single system performance when combining the VGG features with a BLSTM acoustic model. When decoding with an n-gram language model, which are used for deployable systems, we have a realistic production system with a WER of 7.4%. This result is competitive with the current state-of-the-art in the literature. While our focus is on realistic single system performance, we further reduce the WER to 6.1% through system combination and using expensive neural network language model rescoring.

Index Terms: Conversational speech recognition, VGG, bottleneck features, Switchboard

1. Introduction

Neural networks were commonly used to learn features even before the recent resurgence of deep neural networks (DNN) [1]. Several different approaches have been used, but the most common technique is the bottleneck feature [2]. The size of a single hidden layer before the output layer is reduced in size—hence, the term bottleneck. After training the network, the output layer is discarded and the output of the bottleneck layer is used as features for the acoustic model. Bottleneck features offer several advantages. They provide a framework for learning features without relying on handcrafting, though, the bottleneck features must also be trained on some type of feature themselves. In addition, if a two-pass decoding approach is used, the bottleneck features can be speaker-adapted.

The plethora of deep neural networks have mostly been applied to acoustic and language modeling. When applied to feature extraction, the models typically use the standard feed-forward DNN structure. In this work we extend our previous work [3] and use a more sophisticated network for feature extraction. While we have previously demonstrated this can be beneficial for small datasets, it was unknown whether the results would hold for a larger dataset like Switchboard. Sercu et al. [4] have also explored using VGG features in the low-resource setting, but with multilingual training data.

Switchboard has long been the standard benchmark for evaluating speech recognition performance on conversational speech. The rise of deep neural networks [5] has led to a flurry of publications on the dataset [6, 7, 8, 9]. With each publication, the error rate on the task slowly decreases, both from improved acoustic models and improved language models.

As the performance has improved, there has also been interest in comparing with human performance. A recent study from Xiong et al. [10] used human transcribers to retranscribe the test set. Surprisingly, they found their combined systems had finally reached parity with human performance on the same task. In the latest result, Saon et al. [9] further improved performance on the same task. However, they also found significantly improved performance when using professional transcribers on the same test set. It seems the question of whether a system has achieved human level performance depends on the quality of both the system and the human.

Regardless of whether these combined systems have reached human level performance, they are computationally expensive. They require many systems, all decoded separately, to be combined. Each system also uses expensive neural network language model rescoring. These neural network language models improve performance, but are currently too expensive to be used in realistic systems. In this work we focus on a small number of systems using n-gram language models. However, in order to fairly compare with previous results, we also include results using recurrent neural network (RNN) language model rescoring. Using VGG-based bottleneck features allows us to achieve performance comparable with other results in the literature using n-gram language models. The bottleneck networks are also smaller and faster to train than full VGG acoustic models. We show that the VGG features can be used in BLSTM acoustic models to combine the benefits of the convolutional and recurrent networks.

In Section 2 we describe the VGG-based bottleneck features. Our experimental setup and data are described in Section 3. Results comparing the VGG bottleneck features with other types of features are shown in Section 4. In Section 5, we discuss system combination performance and results with RNN language model rescoring. Finally, conclusions are presented in Section 6.

2. VGG-based Bottleneck Features

The vision community has had success using very deep convolutional networks [11]—commonly called VGG networks. Recently, these networks have been applied to speech recognition too [12]. Prior to this recent work, most systems that used convolutional layers used either just one or two layers [13, 14]. The VGG networks use a large number of convolutional layers, each with a filter size of 3x3. Typically, pooling is applied after every two or three convolutional layers. As the pooling reduces the total number of features, the number of filters is typically increased after the pooling to maintain the same modeling capacity. The additional layers give a large improvement over the

http://dx.doi.org/10.21437/Interspeech.2017-1513

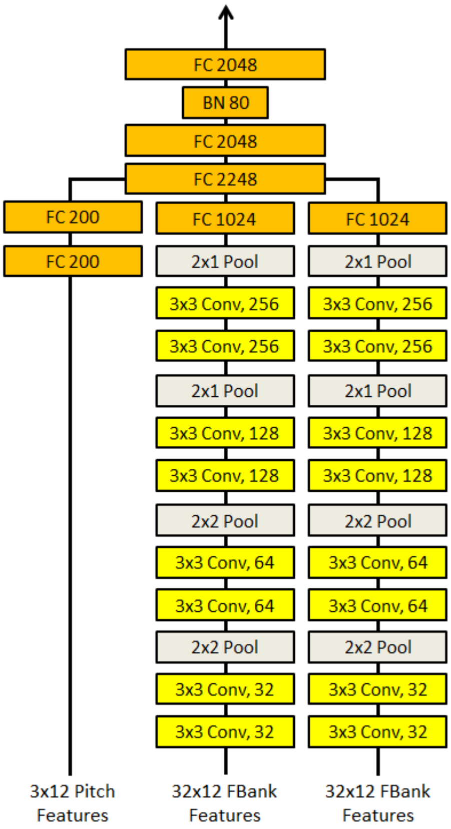

Figure 1: *VGG Network used for the bottleneck feature extractor. Note that the filterbank features used for the two parallel convolutional networks are identical.*

more shallow networks.

An obvious extension is to use the VGG network to generate bottleneck features. Bottleneck features have long been used in ASR and have seen a resurgence along with deep modeling. They are now commonly used for feature extraction [15, 16, 17] and are one approach to incorporating large amounts of multilingual data for under-resourced languages [18]. These networks nearly always consist of several fully connected layers followed by a bottleneck layer and a final fully connected layer before the output. After training, all layers after the bottleneck layer are discarded.

VGG networks have been used for feature extraction in the low-resource setting [3]. The VGG-based features gave significant gains over other types of bottleneck features. Sercu et al. [4] also applied VGG features in a similar setting, but used multilingual training. However, while the performance using VGG features was better than other types of features, using multilingual LSTM models gave the best performance. Results using VGG features on small datasets are promising, but there is no evidence they will outperform other features on larger tasks, especially when used in conjunction with stronger recurrent acoustic models. We apply this work to the standard Switchboard task to see if the performance improvements hold on a much larger dataset.

One issue we found with VGG networks was their inordinate training time compared to feed forward fully-connected networks. In order to make the training tractable, we use a smaller network compared to other recent work [12, 4]. An illustration of our VGG-based bottleneck feature network can be seen in Figure 1. We use eight convolutional layers with a max pooling layer between every two convolutional layers. Two networks with this same structure are trained in parallel. This allows us to increase the total number of filters, and still manage the training time. We also took advantage of batch normalization [19] as it leads to much faster convergence. Using batch normalization and an aggressive learning schedule—we divide the learning rate by ten after each epoch—we were able to train the network in just six passes over the training data. While each epoch is slow compared to other models, by reducing the total number of epochs, we were able to significantly reduce the overall training time.

Table 1: *Source and amount of text data used for language model training.*

Text Source	Number of words
Acoustic transcripts	27M
Broadcast News transcripts	260.3M
CNN transcripts	115.9M
University of Washington web data	525M

3. Experimental Setup

3.1. Switchboard Data

We use the same training data as used in [20]. Acoustic models are trained on 2300 hours of speech from Switchboard I and II, CallHome, and Cellular corpora. The sources for language model data and the associated amount of data is shown in Table 1. Our baseline n-gram language model is a 4-gram model with a 75k vocabulary. We report results on the standard Hub5 2000 evaluation set—both the Switchboard and CallHome test sets.

3.2. System Description

All of our models are trained using Sage [21], the BBN speech processing platform that integrates multiple sources including Kaldi [22] and CNTK [23]. We employ three main types of neural network acoustic models—DNN, TDNN, and (B)LSTM—and three types of front-end features—MFCC+i-vectors, DNN-based bottleneck features, and VGG-based bottleneck features. In general, the models are first trained using cross-entropy before sequence training with sMBR. A more detailed description of the individual features and models follows below.

4. Bottleneck Feature Comparison

4.1. Feature Types

We use two types of bottleneck features. The first is the standard setup using a feed-forward, fully-connected network. It consists of two fully-connected layers with 1500 hidden nodes, a bottleneck layer of dimension 40, and another hidden layer before the final output layer. The second network is the previously described VGG-based bottleneck network with a bottleneck layer of dimension 80. We increased the dimension of the bottleneck layer for the VGG network based on promising preliminary results on a smaller dataset. Both networks use 13 spliced frames of 32-dimensional filterbank features with pitch features [24]. For the VGG network, the pitch features are fed into separate small network and later combined in the first fully-connected layer of the VGG network. This is sim-

Table 2: *Performance comparison of bottleneck features from a DNN and a VGG network on the Switchboard (SWB) and Call-Home(CH) subets of Hub5 2000 evaluation set.*

Acoustic Model	Features	SWB	CH
DNN	DNN-BN	9.0	15.7
DNN	VGG-BN	8.6	14.8
BLSTM	MFCC+ivec	7.8	13.9
BLSTM	DNN-BN	7.9	14.0
BLSTM	VGG-BN	7.4	12.7

ilar to the approach used in [14]. The bottleneck features are always CMLLR-transformed prior to being used in the final acoustic model, requiring a second-pass during decoding. In addition, we also consider 13-dimensional MFCC features plus deltas and double-deltas concatenated with 100-dimensional i-vectors for comparison.

4.2. Acoustic Models

For comparing the different features, two types of acoustic models are considered. The first is a DNN network with six hidden layers of 2048 nodes each. The input consists of 13 spliced frames of the bottleneck features. The second network is a bi-directional long short-term memory network (BLSTM). Each direction of the BLSTM has 1024 memory cells and a 300 dimensional projection layer as described in [25]. As the BLSTM is a recurrent network, no frame-splicing is used on the input. Both networks are trained using cross-entropy and then subsequently sequence trained with the sMBR criterion. Note that the BLSTM seems to converge much quicker than other models during sequence training, so we stop training before even a single pass through the training data.

4.3. Results

The word error rates for each of the networks on the various features can be seen in Table 2. For both acoustic models, the VGG-based features give a similar gain over the more typical DNN-BN features. The comparison between the MFCC features and the bottleneck features is interesting. The speaker-adapted DNN-BN features provide no gain over the MFCC features used in a single decoding pass. This is consistent with our previous experience that when training BLSTMs on large amounts of data, more sophisticated features do not improve over standard filterbank or cepstral features. However, the VGG-BN features do provide a gain. The VGG network is able to learn an improved feature representation. When the features are passed to the BLSTM, they combine to give our best performance on the Switchboard task.

5. The BBN Conversational Telephone Speech System

5.1. Additional Acoustic Models

In addition to the BLSTM and DNN acoustic models described above, we also compare performance with several other types of models that were used in our previous system [26]—TDNN, LSTM, and Chain-BLSTM.

Our time-delay neural network (TDNN) is similar to the one described in [27]. The TDNN has seven total hidden layers. Each has 3000 nodes and is followed by a p-norm nonlinearity to reduce the dimension down to 1500. As in [28], the training data is augmented with two additional speed perturbed copies using speed factors of 0.9 and 1.1.

Similar to the bi-directional variant, we also use an LSTM, though the structure is different. Before the recurrent layers, two fully connected layers of size 2048 are prepended. The LSTM uses two recurrent layers with 2048 memory cells, each with a projection layer of 1024. While the recurrent layers are randomly initialized, the fully-connected layers are pre-trained with RBM.

Finally, we also include the recent chain model, specifically, the BLSTM variant [29]. The BLSTM-based chain model has three BLSTM layers each with 650 units, and 160-dimensional recurrent and non-recurrent projections. In contrast to all other models used in this work, the chain model uses a 3-fold reduced frame rate and lattice-free MMI (LF-MMI) for training. As with the TDNN, the chain model is also trained on two additional copies of speed perturbed data.

Table 3: *Results for all single systems using n-gram language models.*

Acoustic Model	Features	SWB	CH
DNN	DNN-BN	9.0	15.7
DNN	VGG-BN	8.6	14.8
TDNN	MFCC+ivec	8.8	15.4
LSTM	DNN-BN	8.5	15.4
Chain BLSTM	MFCC+ivec	8.4	14.0
BLSTM	MFCC+ivec	7.8	13.9
BLSTM	DNN-BN	7.9	14.0
BLSTM	VGG-BN	7.4	12.7

Table 4: *Joint decoding results. The models are combined by averaging the posteriors at the frame level.*

First Model	Second Model	SWB	CH
TDNN	BLSTM (MFCC+ivec)	7.4	12.9
DNN	BLSTM (VGG-BN)	7.3	12.5

5.2. Results

Complete results for each of the single systems are show in Table 3. In all cases, the BLSTM using VGG bottleneck features is the best performing single system. Regardless of the feature, the BLSTM models are always the best performing single systems.

We also use two approaches to combining systems at decode time—joint decoding [30] and linear least squares (LLS) adaptation [31, 26]. For joint decoding, we average the posteriors from two models at the frame level. Since we only perform joint decoding when the targets of the models are identical, we only have two setups to choose from. Results are shown in Table 4. Both results are better than any of the individual models used in the combination, but the models trained on MFCC features do obtain the largest overall gain. This may mean the TDNN and BLSTM models are more complementary than the DNN and BLSTM models.

Linear least squares adaptation is an unsupervised adaptation technique. The parameters of a the final model used for decoding are updated based on the one-best hypothesis of a first pass model, or on the posteriors of the first pass model directly. Results using our LLS model adaptation are shown in Table 5. The gain from LLS varies depending on the models involved.

Table 5: *Results using LLS model adaptation. The results from the first model are used to adapt the second model.*

First Model	Second Model	SWB	CH
TDNN	LSTM	7.8	14.2
TDNN	BLSTM (MFCC)	7.5	13.3
BLSTM (MFCC)	BLSTM (VGG-BN)	7.5	12.5

Table 6: *Comparing the BBN Conversational Telephone Speech system to results in the literature. All models use n-gram language models during decoding. System results are the result of system combination.*

System Description	SWB	CH
Microsoft VGG+ResNet [10]	8.4	14.5
Microsoft System [10]	7.3	13.0
IBM BLSTM [9]	7.2	12.7
IBM System [9]	6.7	12.1
BBN VGG-BN BLSTM	7.4	12.7
BBN System	6.7	11.3

Typically we want the second model to be at least as good as the first model. While reversing the order of the models (e.g. adapting the DNN based on the BLSTM output) would still improve the second model, it is typically no better than using the first model directly. Unfortunately, our two best models do not help each other. This is likely due to a lack of diversity between the two models; model diversity was shown to be important in the performance of LLS in [26].

Finally, given the performance of all of our individual decodings, we also perform system combination using ROVER. In order to avoid overfitting, we limit the total number of systems used in combination to be four. Results comparing our final system to other state-of-the-art results in the literature are shown in Table 6. Our overall system performance is competitive with the best results in the literature when using n-gram language models. We do note that some of the systems listed differ in the data used for training their acoustic or language model.

5.3. RNN Language Model Results

There is a caveat with these numbers though. Obviously, the goal has always been to achieve the best overall performance. To that end, the recent work from Microsoft and IBM have focused on results using both RNN language models and LSTM language models. It is reasonable to believe their n-gram numbers could potentially improve if that was their focus. However, we can only compare with the results that exist in the literature.

While our focus was on performance with n-gram language models, as we believe they are currently more practical, we also added a basic RNN language model to compare results with other work. We use the CUED-RNNLM toolkit [32] and follow a similar training procedure to Xiong et al. [10]. Both a forward and backward RNN-LM were used. Each network had two hidden layers with 1000 nodes and the ReLU nonlinearity function. The networks were trained with cross-entropy loss, as opposed to the noise-contrastive estimation used in [10]. The vocabulary was the same 75k words used in the n-gram language model. After decoding with an n-gram language model, n-best lists were rescored using a weighted combination of the n-gram model and the two RNN language models.

Results comparing our system using RNN-LM rescoring and other results in the literature are shown in Table 7. Our performance is competitive with the Microsoft [10] result, but our Switchboard number is behind the best result from IBM [9]. We note that the gap in performance at least partially comes from their use of LSTM language models.

Table 7: *Comparison of results in the literature using neural network language models. The final two results show the performance of human transcribers on the same test set. System results are the result of system combination.*

System Description	SWB	CH
Microsoft VGG+ResNet [10]	6.4	12.2
Microsoft System [10]	5.8	11.0
IBM System [9]	5.5	10.3
BBN VGG-BN BLSTM	6.7	11.5
BBN System	6.1	10.4
Human Transcription Microsoft [10]	5.9	11.3
Human Transcription IBM [9]	5.1	6.8

If we consider the human performance reported in the Microsoft paper, then we can also claim our system has reached parity with humans on this task. However, when compared to the better humans used by IBM for transcription, we still see a large gap, especially in CallHome subset—though that is true for all current systems in the literature. Just like with automatic speech recognition, the WER from humans likely varies based on the real-time factor.

6. Conclusions

We successfully applied VGG-based bottleneck features to the Switchboard conversational telephone speech task. VGG features give a gain over traditional features, even when used with BLSTM acoustic models. When used in a BLSTM acoustic model, using the VGG features as input give the best overall performance in the BBN conversational telephone speech system. Our best single system achieves comparable performance to the best currently published result when using n-gram language models. When using system combination with n-gram based models, our result of 6.7% is also consistent with the best published results. Our results with RNN-LM rescoring are competitive with the previous best result from Microsoft [10], but is behind the latest result from IBM [9]. In future work, we will continue to improve our system with a focus on practical, real-time performance.

7. Acknowledgements

This research was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense US Army Research Laboratory contract number W911NF-12-C-0013. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government.

8. References

[1] H. Hermansky, D. Ellis, and S. Sharma, "Tandem Connectionist Feature Extraction for Conventional HMM Systems," in *ICASSP*,

2000.

[2] V. Fontaine, C. Ris, and J. M. Boite, "Nonlinear discriminant analysis for improved speech recognition," in *Eurospeech*, 1997.

[3] W. Hartmann, R. Hsiao, and S. Tsakalidis, "Alternative networks for monolingual bottleneck features," in *ICASSP*, 2017.

[4] T. Sercu, G. Saon, J. Cui, X. Cui, B. Ramabhadran, B. Kingsbury, and A. Sethy, "Network architectures for multilingual speech representation learning," in *ICASSP*, 2017.

[5] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[6] I. Medennikov, A. Prudnikov, and A. Zatvornitskiy, "Improving english conversational telephone speech recognition," in *Interspeech*, 2016.

[7] G. Saon, T. Sercu, S. Rennie, and H.-K. J. Kuo, "The ibm 2016 english conversational telephone speech recognition system," in *Interspeech*, 2016.

[8] W. Xiong, J. Droppo, X. Huang, F. Seide, M. L. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "The microsoft 2016 conversational speech recognition system," in *ICASSP*, 2017.

[9] G. Saon, G. Kurata, T. Sercu, K. Audhkhasi, S. Thomas, D. Dimitriadis, X. Cui, B. Ramabhadran, M. Picheny, L.-L. Lim, B. Roomi, and P. Hall, "English conversational telephone speech recognition by humans and machines," *arXiv preprint arXiv:1703.02136*, 2017.

[10] W. Xiong, J. Droppo, X. Huang, F. Seide, M. L. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "Achieving human parity in conversational speech recognition," *arXiv preprint arXiv:1610.05256*, 2016.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional neural networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[12] T. Sercu, C. Puhrsch, B. Kingsbury, and Y. LeCun, "Very deep multilingual convolutional neural networks for LVCSR," in *ICASSP*, 2016.

[13] T. Sainath, O. Vinalys, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *ICASSP*, 2015.

[14] K. Vesely, M. Karafiát, and F. Grézl, "Convolutive bottleneck network features for LVCSR," in *ASRU*, 2011.

[15] F. Grézl, M. Karafiát, and L. Burget, "Investigation into bottleneck features for meeting speech recognition," in *Interspeech*, 2009.

[16] K. Vesely, M. Karafiát, F. Grézl, M. Janda, and E. Egorova, "The language-independent bottleneck features," in *SLT*, 2012.

[17] Z. Tüske, J. Pinto, D. Willett, and R. Schluter, "Investigation on cross-and multilingual MLP features under matched and mismatched acoustical conditions," in *ICASSP*, 2013.

[18] T. Alumäe, S. Tsakalidis, and R. Schwartz, "Improved multilingual training of stacked neural network acoustic models for low resource languages," in *Interspeech*, 2016.

[19] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[20] R. Prasad, S. Matsoukas, C. Kao, J. Ma, D. Xu, T. Colthurst, O. Kimball, R. Schwartz, J. Gauvain, L. Lamel, H. Schwenk, G. Adda, and F. Lefevre, "The 2004 bbn/limsi 20xrt english conversational telephone speech recognition system," in *Interspeech*, 2005.

[21] R. Hsiao, R. Meermeier, T. Ng, Z. Huang, M. Jordan, E. Kan, T. Alumäe, J. Silovsky, W. Hartmann, F. Keith, O. Lang, M. Siu, and O. Kimball, "Sage: The new BBN speech processing platform," in *Interspeech*, 2016.

[22] 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," in *ASRU*, 2011.

[23] D. Yu, A. Eversole, M. Seltzer, K. Yao, B. Guenter, O. Kuchaiev, F. Seide, H. Wang, J. Droppo, Z. Huang, Y. Zhang, G. Zweig, C. Rossbach, J. Currey, J. Gao, A. May, A. Stolcke, and M. Slaney, "An introduction to computational networks and the computational network toolkit," Microsoft Research, Tech. Rep., 2014.

[24] P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Trmal, and S. Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," in *ICASSP*, 2014, pp. 2494–2498.

[25] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," in *Interspeech*, 2014.

[26] R. Hsiao, T. Ng, and M.-H. Siu, "Unsupervised adaptation for deep neural networks using alternating direction method of multipliers," in *ICASSP*, 2017.

[27] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Interspeech*, 2015.

[28] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Interspeech*, 2015.

[29] D. Povey, V. Peddinti, D. Galvez, P. Ghahrmani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free MMI," in *Interspeech*, 2016.

[30] W. Hartmann, L. Zhang, K. Barnes, R. Hsiao, S. Tsakalidis, and R. Schwartz, "Comparison of multiple system combination techniques for keyword spotting," in *Interspeech*, 2016.

[31] R. Hsiao, S. Tsakalidis, T. Ng, L. Nyugen, and R. Schwartz, "Unsupervised adaptation for deep neural networks using linear least square method," in *Interspeech*, 2015.

[32] X. Chen, X. Liu, Y. Qian, M. Gales, and P. Woodland, "CUED-RNNLM: An open-source toolkit for efficient training and evaluation of recurrent neural network language models," in *ICASSP*, 2016, pp. 6000–6004.