ALTERNATIVE NETWORKS FOR MONOLINGUAL BOTTLENECK FEATURES

William Hartmann, Roger Hsiao, Stavros Tsakalidis

Raytheon BBN Technologies, Cambridge, MA, USA
{whartman, whsiao, stavros}@bbn.com

ABSTRACT

While recent advances in deep neural networks have lead to significant improvements in speech recognition, they have been applied mainly to acoustic and language modeling. We instead apply the models to bottleneck feature extraction. Several DNN, CNN, and BLSTM-based bottleneck feature networks are compared using both DNN and BLSTM acoustic models. Multiple variations in network architecture and feature input are explored. Results are reported on four languages from the IARPA Babel program. The shallow CNN and BLSTM both improve performance by a similar amount. The best network is a deep CNN and improves WER by 1.4% and ATWV by 2% absolute compared to the baseline DNN network when using a DNN acoustic model. Relative gains hold when using stronger BLSTM acoustic models.

Index Terms— bottleneck features, deep neural network, babel

1. INTRODUCTION

Prior to the resurgence of deep neural networks for acoustic modeling in automatic speech recognition (ASR) [1], smaller neural networks had long been used for feature extraction. Neural network-based features, also known as MLP features, provided a method for applying discriminative training without the acoustic model. They could potentially learn a feature representation more suitable for GMM acoustic models than standard cepstral features. These features could also be trained on large windows of input features, implicitly giving the acoustic models access to a larger context, but with a small feature vector.

The Tandem approach [2] trained a phone recognizer for use as features in a GMM-HMM system. The phone posteriors, prior to the final softmax, were used directly. As the final model was a GMM, PCA was typically applied to decorrelate the final features. Often, these features were concatenated with the original PLP features for best performance [3]. Fontaine et al. [4] were one of the first to use a bottleneck layer prior to the output layer for feature extraction. However, the number of targets and the network size were small, so the dimensionality of the bottleneck layer differed little from the output layer. The more modern structure with an additional hidden layer after the bottleneck layer was demonstrated in [5] to significantly outperform features based on the output of the networks. Since then, the use of bottleneck features and their performance has increased greatly [6, 7], especially in the context of multilingual speech recognition [8, 9].

Along with the typical deep neural network, a host of more sophisticated models and networks have been introduced. Convolutional neural networks (CNN) exploit the local structure of filterbank features [10]. Long-short term memory networks (LSTM)—and their bi-directional (BLSTM) variant—use a recurrent structure to model long term temporal information to improve recognition [11, 12]. Recent work has even combined all of these ideas into a single model [13]. However, there has been little work in applying these models to bottleneck feature extraction.

Vesely et al. did compare standard bottleneck features with simple CNN-based bottleneck features [14]. Several other studies have mentioned the use of CNN-based bottleneck features, but without much discussion of the features themselves [15, 16]. In this work we compare using a DNN, two types of CNNs, and a BLSTM for extracting bottleneck features. We demonstrate significant improvements over the DNN-based baseline with both DNN and BLSTM acoustic models on four languages from the IARPA Babel Project. A detailed discussion of the bottleneck networks is presented in Section 2. Section 3 describes the experimental setup. Results are presented in Section 4, and conclusions in Section 5.

2. BOTTLENECK FEATURE NETWORKS

2.1. Fully Connected Network

Even before the resurgence of deep neural networks, fully connected networks were used to generate bottleneck or MLP features [5]. The traditional structure of the network is two hidden layers, a single bottleneck layer, and an additional hidden layer before the output layer. Given that we are exploring more sophisticated and larger alternative networks in this study, we also test increasing the number of hidden layers in the traditional fully connected network. If other networks provide better performance, we want to eliminate the possibility that the only factor is increased size. While it is true that all networks considered in this work are deep neural networks, we will refer to the network with only fully connected layers as a DNN.

2.2. Convolutional Network

Popular among the image community, convolutional networks have also found success in speech recognition [15]. The convolutional neural network (CNN) exploits local correlations—both temporal and across frequency—in the input features. Filters are learned and shared across the entire input image. However, we have found that with smaller datasets, the CNN does not perform as well as the DNN for acoustic modeling [16]—likely due to the difficulty in applying speaker adaptation to the input features. An alternative approach to utilizing CNNs in ASR is for bottleneck feature generation.

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.

Initially, CNNs used the same network structure as DNNs, except two convolutional layers were prepended to the network [10]. More recent work has used deeper networks with many more layers, but smaller 3x3 filters [17]. While first introduced for image recognition, the very deep CNNs—referred to as VGG networks—have shown promising results in speech recognition [18]. They have also been used in a multilingual setting [19]. In this work, we consider the VGG network—the 8-layer variant with a max pooling layer after every two convolutional layers—in addition to the traditional 2-layer CNN. In both cases the convolutional layers are prepended to the bottleneck layer in the network, and the number of layers refer only to the total number of convolutional layers. As the VGG network uses a large number of layers, both the number of parameters and training time increases. We explore using similar networks with smaller filter sizes in an attempt to reduce the computational cost of the network without hurting performance.

2.3. BLSTM Network

The final type of network is the recurrent network. In particular, we use the bi-directional long short term memory (BLSTM) network [12]. Memory cells allow the network to retain information over a large time span. Our networks are based on the variant proposed in [20] with peephole connections and a recurrent projection layer. In our experience, and in this study, BLSTM acoustic models perform better than DNNs and CNNs on the datasets used in this paper. Given their strength in acoustic modeling, it is worth exploring their utility in generating bottleneck features.

We explore only a single BLSTM network in this study. The bottleneck feature network consists of three recurrent layers, with the middle layer being a bottleneck layer. This differs from the other type of networks as there are no fully connected layers in the network. An alternative structure would use a fully connected bottleneck layer following the recurrent layers, however, we found training to be unstable using this approach and do not report results. We have not previously seen BLSTM-based bottleneck features used for ASR, but a recent study applied BLSTM bottleneck features to textual and intonation feature extraction [21].

3. EXPERIMENTAL SETUP

All experiments use languages from the IARPA Babel project. The IARPA Babel dataset consists of conversational telephone speech for 25 languages collected across a variety of environments. The total amount of transcribed audio data varies depending on the language and condition. Our focus was on the full language pack (FLP) from the fourth year—approximately 40 hours of transcribed audio for the languages considered. The 10 hour development set is used for testing. While we report results on WER, the primary focus is actual term-weighted value (ATWV). ATWV is a keyword spotting metric with values ranging from $-\infty$ to a maximum of one. We use a set of approximately 2000 keywords per language. The final ATWV score is the average of the individual scores for each keyword. See [22] for a more detailed discussion of ATWV.

We selected four development languages from the final year of the program: Amharic (IARPA-babel307b-v1.0b), Guarani (IARPA-babel305b-v1.0c), Igbo (IARPA-babel306b-v2.0c), and Pashto (IARPA-babel104b-v0.bY). Igbo is initially used for determining input features and parameter setups for each type of network. Once all training parameters are set, results are compared across languages. Only the transcribed audio was used for training trigram language models, though we can obtain improved performance with additional text collected from the web [23]. Pronunciations lexicons were generated using simple G2P rules [24].

We use the Sage ASR toolkit [25] for building the system. Sage is BBN's newly developed speech-to-text transcription (STT) platform that integrates technologies from multiple sources, each of which has a particular strength. In Sage, we combine proprietary sources, such as BBN's Byblos [26], with open source toolkits, such as Kaldi [27] and CNTK [28]. Sage also includes a cross-toolkit FST recognizer that supports models built using the various component technologies, and software supporting keyword search from Byblos [29, 30, 31]. Two types of acoustic models are used in this work. The first are 6 hidden layer DNNs with 2048 hidden nodes in each layer. The second are 3-layer BLSTM networks. Each layer has both a forward and backward direction with 512 memory cells and a projection layer of 300 in each direction. Both acoustic models are sequence trained.

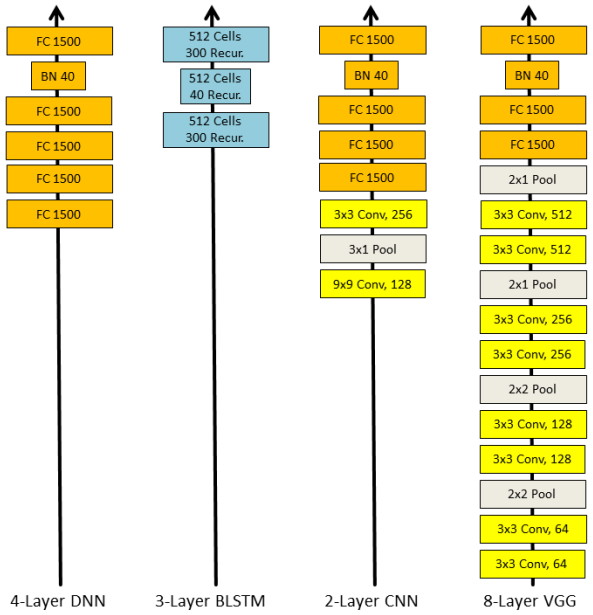

Fig. 1. Bottleneck feature architectures. Each layer describes the type of node and the number of parameters. The orange fully-connected (FC) layers denote the number of hidden nodes. The blue recurrent layers describe the number of memory cells and the number of recurrant projection nodes. The yellow convolution layers denote the dimension and number of filters, and the gray max pooling layer denote the dimension of pooling.

All acoustic models use speaker adapted training (SAT) and the final input to the acoustic model is 13 spliced frames of fMLLR-transformed bottleneck features without delta features. The networks used for bottleneck feature extraction were trained using CNTK. For the DNN and CNN, we use ReLU activation units and batch normalization [32] during training. Batch normalization greatly improved the speed of convergence for the CNN and DNN bottleneck feature extractors; it also improved performance when not using pretraining. We also attempted to use batch normalization for BLSTM training, but it did not improve convergence or performance. All fully-connected hidden layers use 1500 hidden units and a bottleneck layer of size 40. Preliminary experiments with larger sizes did not improve performance, but larger values have

BN Network	w/ Deltas	w/o Deltas
2-Layer DNN	56.8	56.5
2-Layer CNN	56.1	55.0
BLSTM	55.3	55.3

Table 1. WER Results on Igbo when using delta features as input to the bottleneck network.

been shown to help in the multilingual setting [33]. Inputs to all BN networks are 13 frames of 32-dimensional filterbank features with pitch features [34]. For the CNN, the filterbank and pitch features are treated separately; a parallel fully-connected network uses the pitch features and the output is concatenated with the output from the CNN before the bottleneck layer. The two-layer CNN uses a 9x9 convolution followed by a 3x3 convolution. The number of filters is 128 and 256, respectively. The VGG model uses only 3x3 convolutional layers with the number of filters ranging form 64 to 512. Smaller VGG models are also considered where the number of filters is reduced by a constant factor at each layer. The BLSTM BN network uses a similar structure and parameters to the BLSTM acoustic model previously described. A detailed representation of the networks can be seen in Figure 1.

4. RESULTS

4.1. Delta Features

Delta and double-delta features have long been used in GMM-based ASR. CNNs also typically benefit from the inclusion of delta features. We experiment with including delta features with all three kinds of networks for BN feature extraction. For the BLSTM and DNN networks, the delta features are simply concatenated with the static features. For the CNN, the three types of features all pass through their own parallel convolutional network. Their outputs are joined together before the bottleneck layer. While the use of delta features only increase the input layer of the DNN and BLSTM by a small amount, it effectively triples the number of convolutional layers.

Results are shown in Table 1. The delta features have minimal effect on the DNN, but do not improve performance. Performance with the CNN is more dramatic; delta features degrade performance drastically. We have experimented with a large number of CNN variations and this pattern holds. Delta features always decrease performance for CNN-based BN features, though, they are beneficial when used in acoustic modeling.

One possibility for this reduction is as further context is added to the bottleneck network, the bottleneck layer captures more information about the surrounding context and less information about the current frame. However, this is unnecessary as the final acoustic model also uses a stack of input features. This would also explain why the BLSTM sees no difference in performance with the delta features. The BLSTM already captures information about a large window given its recurrent nature; the introduction of delta features does little to increase the amount of contextual information available to the network. Another possibility is it causes the final acoustic model to overfit, since both the acoustic model and the BN network are trained on the same data. Given the failure of delta features to improve performance, they are not used for any further experiments.

BN Network	WER
2-Layer DNN	56.5
3-Layer DNN	56.1
4-Layer DNN	55.9
5-Layer DNN	55.9

Table 2. WER Results on Igbo using a DNN for BN feature extraction with different numbers of hidden layers before the bottleneck layer.

BN Network	Filter Size	WER
2-Layer CNN	full	55.0
8-Layer VGG	full	54.1
8-Layer VGG	half	54.2
8-Layer VGG	quarter	54.8

Table 3. WER Results on Igbo for various CNN model configurations.

4.2. Number of Fully Connected Layers

As we are comparing performance against very deep CNN models, it may not be fair to just use the traditional DNN bottleneck feature network for comparison. In Table 2 we show results using a larger number of hidden layers prior to the bottleneck layer. The gains are modest, but there is a definite trend from increasing the number of layers. Both the four and five layer network give the same performance. For further comparisons with other languages, we will use the 4-layer network as our baseline.

4.3. CNN Model Structure

We are interested in varying the CNN model structure over various dimensions. The first is the depth of the network, specifically, the 2-layer CNN vs. the 8-layer VGG model. For the VGG model, we also test whether we can reduce the total number of filters at each layer. Training the larger VGG model is computationally more expensive, so reducing the number of filters could significantly decrease training time and memory usage.

Results are shown in Table 3. Compared with the DNN results in Table 2, the basic 2-layer CNN improves performance by almost one point in WER. Moving to the larger VGG model doubles the gain. While using the VGG model with only a quarter of the parameters, negatively impacts performance, the model with half the number of parameters is nearly identical in terms of WER. Halving the filter size approximately halves the training time as well. Since the reduction in training time comes at no cost in performance, we will use this reduced version of the VGG model for the remainder of the experiments. Compared to Figure 1, this version uses half the number of filters at each layer, starting with 32 and growing to 256.

4.4. Performance Across Languages

Now that we have chosen the input features and the model structures, we compare the DNN, CNN, and BLSTM-based BN features on three additional languages, in addition to Igbo. WER results for all models are shown in Table 4 and ATWV results are shown in Table 5. Note that WER and ATWV measure separate aspects of models and an improvement in one does not necessarily guarantee an improvement in the other. Better features can sometimes produce sharper models that only improve WER, but that is not the case with these features. For three of the languages, the results are similar.

BN Network	Amharic	Guarani	Igbo	Pashto
4-Layer DNN	43.4	45.9	55.9	48.4
2-Layer CNN	42.7	45.6	55.0	47.4
8-Layer VGG	41.6	45.3	54.2	46.8
BLSTM	42.4	46.1	55.3	47.1

Table 4. WER results using a DNN acoustic model with the various bottleneck features.

BN Network	Amharic	Guarani	Igbo	Pashto
4-Layer DNN	0.599	0.553	0.351	0.416
2-Layer CNN	0.609	0.565	0.361	0.432
8-Layer VGG	0.622	0.568	0.375	0.446
BLSTM	0.604	0.562	0.364	0.438

Table 5. ATWV results using a DNN acoustic model with the various bottleneck features.

Both the CNN and BLSTM model outperform the DNN by a similar amount, but the VGG features clearly give the best performance. This is true for WER and ATWV.

Guarani does not follow this pattern in terms of WER. The BLSTM features provide no gain, and even the VGG features give less than one point improvement. Gains in ATWV are a little larger, but still less than the other three languages. It is unclear why the different bottleneck features would not perform as well for Guarani as for the other languages. On average the gain for the VGG features over the baseline is 1.4% absolute for WER and 2% for ATWV. These gains are on the same order as we have previously seen with joint decoding [16] and data augmentation [35].

4.5. BLSTM Acoustic Models

While the improvements from the VGG features over the baseline DNN are good, we also wanted to test whether the gains would still hold when using a stronger acoustic model. On average, the BLSTM is 1.4% better in terms of WER, but there is no gain in ATWV. A comparison of the VGG and DNN-based BN features are presented in Table 6.

Overall the numbers are better than those with the DNN acoustic model in Tables 4 and 5. Furthermore, the relative improvements are just as good when moving from the baseline DNN-based features to the VGG-based features. Improving the acoustic model did not reduce the effect of the improved VGG-based features.

Language	BN Network	WER	ATWV
Amharic	4-Layer DNN	42.2	0.573
Amharic	8-Layer VGG	40.7	0.607
Guarani	4-Layer DNN	45.1	0.553
Guarani	8-Layer VGG	43.3	0.576
Igbo	4-Layer DNN	54.0	0.368
Igbo	8-Layer VGG	52.7	0.389
Pashto	4-Layer DNN	46.6	0.423
Pashto	8-Layer VGG	45.7	0.438
Average	4-Layer DNN	47.0	0.479
Average	8-Layer VGG	45.6	0.503

Table 6. Comparison of the VGG-based and DNN-based BN features using a BLSTM acoustic model across languages.

5. CONCLUSION

We have explored the use of the larger, more sophisticated models typically used in acoustic modeling for bottleneck feature extraction. After establishing a DNN-based baseline, several alternative structures were tested. Both the basic 2-layer CNN and the BLSTM BN features gave similar performance across a set of four languages when using DNN acoustic models. The VGG features significantly outperformed all other features, even with half the number of filters typically used. We should note that training the VGG and BLSTM networks required a similar amount of computational effort. In contrast, the baseline DNN features could be trained in about one third the amount of time. The effect on decode time was limited, though, as the bottleneck feature extraction step is not typically the most expensive part of decoding. We also demonstrated that the relative gains from the VGG features held when using the stronger BLSTM acoustic models, both in terms of WER and ATWV.

6. REFERENCES

[1] 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.

[2] H. Hermansky, D. Ellis, and S. Sharma, "Tandem Connectionist Feature Extraction for Conventional HMM Systems," ICASSP, 2000.

[3] N. Morgan, B. Y. Chen, Q. Zhu, and A. Stolcke, "TRAPping conversational speech: Extending TRAP/TANDEM approaches to conversational telephone speech recognition," in *Proceedings of ICASSP*, 2004.

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

[5] F. Grézl, M. Karafiát, S. Kontár, and J. Cernocky, "Probabilistic and bottle-neck features for LVCSR of meetings," in *Proceedings of ICASSP*, 2007.

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

[7] D. Yu and M. Seltzer, "Improved bottleneck features using pretrained deep neural networks," in *Proceedings of Interspeech*, 2011.

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

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

[10] O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recogntion," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 10, pp. 1533–1545, 2014.

[11] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proceedings of ICASSP*, 2013.

[12] A. Graves, N. Jaitly, and A. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *Proceedings of ASRU*, 2013.

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

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

[15] T. Sainath, A. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *Proceedings of ICASSP*, 2013.

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

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

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

[19] "Very deep multilingual convolutional neural networks for LVCSR, author=T. Sercu and C. Puhrsch and B. Kingsbury and Y. LeCun, booktitle=Proceedings of IEEE ICASSP, pages=4955–4959, year=2016, organization=IEEE," .

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

[21] L. Li, Z. Wu, M. Xu, H. Meng, and L. Cai, "Combining CNN and BLSTM to extract textual and acoustic features for recognizing stances in mandarin ideological debate competition," in *Proceedings of Interspeech*, 2016, pp. 1392–1396.

[22] S. Wegmann, A. Faria, A. Janin, K. Riedhammer, and N. Morgan, "The tao of ATWV: Probing the mysteries of keyword search performance," in *Proceedings of IEEE ASRU*, 2013, pp. 192–197.

[23] L. Zhang, D. Karakos, W. Hartmann, R. Hsiao, R. Schwartz, and S. Tsakalidis, "Enhancing low resource keyword spotting with automatically retrieved web documents," in *Interspeech*, 2015, pp. 839–843.

[24] M. Davel, E. Barnard, C. van Heerden, W. Hartmann, D. Karakos, R. Schwartz, and S. Tsakalidis, "Exploring minimal pronunciation modeling for low resource languages," in *Interspeech*, 2015, pp. 538–542.

[25] 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.

[26] S. Tsakalidis, R. Hsiao, D. Karakos, T. Ng, S. Ranjan, G. Saikumar, L. Zhang, L. Nyugen, R. Schwartz, and J. Makhoul, "The 2013 BBN Vietnamese telephone speech keyword spotting system," in *ICASSP*, 2014.

[27] 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.

[28] 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," Tech. Rep., Microsoft Research, 2014.

[29] D. Karakos, I. Bulyko, R. Schwartz, S. Tsakalidis, L. Nguyen, and J. Makhoul, "Normalization of phonetic keyword search scores," in *Proc. of ICASSP*, Florence, Italy, 2014.

[30] D. Karakos and R. Schwartz, "Combination of search techniques for improved spotting of OOV keywords," in *Proc. of ICASSP*, 2015.

[31] D. Karakos, R. Schwartz, S. Tsakalidis, L. Zhang, S. Ranjan, T. Ng, R. Hsiao, G. Saikumar, I. Bulyko, L. Nguyen, J. Makhoul, F. Grezl, M. Hannemann, M. Karafiat, I. Szoke, K. Vesely, L. Lamel, and V.-B. Le, "Score normalization and system combination for improved keyword spotting," in *Proc. of ASRU*, Olomouc, Czech Republic, 2013.

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

[33] Tanel Alumäe, Stavros Tsakalidis, and Richard Schwartz, "Improved multilingual training of stacked neural network acoustic models for low resource languages," in *Interspeech*, 2016.

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

[35] William Hartmann, Tim Ng, Hsiao Roger, Stavros Tsakalidis, and Richard Schwartz, "Two-stage data augmentation for low-resourced speech recognition," in *Interspeech*, 2016.