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

Improved Multilingual Training of Stacked Neural Network Acoustic Models for Low Resource Languages

Tanel Alumäe, Stavros Tsakalidis, Richard Schwartz

Raytheon BBN Technologies, Cambridge, MA, USA
{talumae, stavros, rschwartz}@bbn.com

Abstract

This paper proposes several improvements to multilingual training of neural network acoustic models for speech recognition and keyword spotting in the context of low-resource languages. We concentrate on the stacked architecture where the first network is used as a bottleneck feature extractor and the second network as the acoustic model. We propose to improve multilingual training when the amount of data from different languages is very different by applying balancing scalers to the training examples. We also explore how to exploit multilingual data to train the second neural network of the stacked architecture. An ensemble training method that can take advantage of both unsupervised pretraining as well as multilingual training is found to give the best speech recognition performance across a wide variety of languages, while system combination of differently trained multilingual models results in further improvements in keyword search performance.

Index Terms: speech recognition, keyword spotting, multilingual training, deep learning, system combination

1. Introduction

Multilingual training strategies for speech recognition and keyword search (KWS) of low-resourced languages have recently gained much attention [1, 2, 3, 4]. Deep neural network (DNN) based acoustic models trained jointly on several languages and later ported to the target language outperform models trained only on the target language data, especially when the amount of labelled data from the target language is limited.

This paper investigates two methods for enhanced multilingual training of DNN acoustic models, using the IARPA Babel project [5] final year (OP3) development languages as the main focus. The first method tackles the unbalanced data problem when training a multilingual bottleneck (BN) feature extractor on languages with varying amounts of data. Research has shown that increasing the number of languages used for training multilingual DNN models improves the accuracy of speech recognition and keyword spotting in the target language [3]. However, it has also been shown that the choice of languages that are used for training multilingual models has an impact on the model performance on the target language, even after fine-tuning the multilingual model on target language data: using languages similar to the target language are more beneficial as donor languages for multilingual training [6].

Although careful selection of donor languages can improve the final system accuracy, it can require prohibitively large resources when rapid porting to several target languages is needed. Therefore, it is beneficial to have a robust multilingual donor model that works well across all target languages, without major risks of having the multilingual model dominated by languages that are very distinct from the target. Another point to consider when training multilingual models is the skewness of the data: in many real-world scenarios, the amount of available training data from different languages can vary hugely. Combining such unbalanced corpora for multilingual training has the danger of having the final model completely dominated by the language(s) with large amounts of data, while the minor languages do not have much effect of on the final multilingual model. Furthermore, it has been shown that training a multilingual feature extractor on heavily inbalanced data can cause degradation of the final system [7].

One method to solve the data skewness problem is undersampling the languages with large amounts of training data, or oversampling the languages with less data. However, if we over-sample the languages with less data to make them balanced with the language with the most data, we also increase the training time of the model, possibly by many orders of magnitude. On the other hand, under-sampling the language with large amount of data has the risk of potentially removing important training examples at each epoch. Instead of data resampling, we propose to scale the training examples so that the total contribution from each language over each epoch will be equal. A similar approach has been previously used for balancing training data across labels in a different machine learning task [8]. We evaluate the method on an unbalanced dataset with six languages. Five different languages are used as targets.

The second problem that is investigated in the paper considers multilingual training for the stacked hybrid architecture. This architecture consists of two DNNs that are trained separately: the first model is used for extracting BN features and the second for computing state-level posteriors for HMM decoding. Most previous experiments with multilingual DNNs have focused on two other common architectures used in speech recognition: bottleneck based tandem architecture, where a DNN (or a stack of two DNNs) is used for computing the BN features and a GMM is used for computing observation probabilities [9, 10, 11, 12, 3, 6], and the (unstacked) hybrid architecture, consisting of one DNN (or CNN) that computes state posteriors directly based on MFCC, PLP or filterbank features [13, 14, 15, 4]. We are not aware of any published work where multilingual data is used for training the second DNN of the stacked hybrid architecture. We focus on this architecture because we have found it to work robustly and accurately across many different decoding tasks. We compare the performance of three systems where the second DNN of the stacked architecture is either pre-trained on target language data, initialized using the multilingual model, or trained jointly using both of the initializations. We also compare keyword search performance of the three systems and their different combinations.

http://dx.doi.org/10.21437/Interspeech.2016-1426

2. Experimental setup

We use the Sage ASR toolkit [16] for all experiments. Sage is BBN's newly developed 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 [17], with open source toolkits, such as Kaldi [18], CNTK [19] and Tensorflow. Sage also includes a cross-toolkit FST recognizer that supports models built using the various component technologies, and software supporting keyword search from Byblos [20].

We use 32 filterbank features, combined with 3 Kaldi pitch features as input to the BN DNNs. The input features are stacked to accumulate a temporal context of 11 frames. The bottleneck networks have two hidden layers before and one after the bottleneck layer. The bottleneck layer has a dimensionality of 40. The bottleneck features are used for GMM-based speaker-adapted training (SAT), and the fMLLR-transformed bottleneck features are used as input to the second DNN which is used as the acoustic model (AM) in the hybrid DNN-HMM setup. The DNN-AM uses a temporal context of 13 stacked frames, it has six 2048-dimensional hidden layers with the sigmoid activation function, and the output layer has approximately 4500 tied-state targets. Unsupervised RBM-based pretraining is used to initialize the weights of the hidden layers.

The multilingual bottleneck networks have language-specific block-softmax output layers [10] with approximately 4500 context-dependent targets per language. The multilingual BN DNN is ported to the target language by first replacing the block-softmax with the target language softmax, training the softmax for two epochs while keeping the rest of the model fixed, and finally training the whole model for four epochs, using a 10 times smaller learning rate than the original.

Experiments are performed on data from the final year of the IARPA Babel program [5]: Amharic (IARPA-babel307b-v1.0b), Dholuo (IARPA-babel403b-v1.0b), Guarani (IARPA-babel305b-v1.0c), Igbo (IARPA-babel306b-v2.0c), Javanese (IARPA-babel402b-v1.0b), Mongolian (IARPA-babel401b-v2.0b) and Pashto (IARPA-babel104b-v0.bY). For each language, the full language pack (FLP) is used, containing approximately 40 hours of transcribed audio — the audio is conversational telephone speech collected in a variety of conditions. Lexicons are derived using simple G2P rules [21]. Trigram language models are built from the training transcriptions. Decoding is performed on an additional 10 hours of development data, and keyword search uses a set of approximately 2000 keywords for each language. Both whole word and phonetic search are used for keyword spotting [22].

Actual term-weighted value (ATWV) is the primary measure of interest for the IARPA Babel program. ATWV was also used in the NIST 2006 Spoken Term Detection evaluation [23]. The keyword specific ATWV for keyword k at a specific threshold t can be computed by

$$\text{ATWV}(k,t) = 1 - P_{FR}(k,t) - \beta P_{FA}(k,t) \tag{1}$$

where P_{FR} and P_{FA} refer to the probability of a false reject (miss) and false accept, respectively. The constant β — set to a value of 999.9 — defines the trade off between false accepts and false rejects. In this performance metric, all keywords are equally weighted. Missing a single occurrence of a rare word can affect the final score as much as missing a more common word dozens of times. While ATWV numbers are commonly reported for in-vocabulary and out-of-vocabulary (OOV) keywords separately, we report only the overall performance.

3. Balanced training of multilingual bottleneck features

This experiment investigates whether a strongly imbalanced multilingual dataset can be artificially rebalanced in order to increase the performance of the multilingual bottleneck feature extractor trained on the dataset.

During training of the bottleneck feature extractor, we usually pool the data across all languages together and train a DNN, using a block-softmax output layer. Since one or more languages are over-represented in the language pool, it becomes profitable for the learning process to optimize the hidden layers, including the bottleneck, so that the targets of the dominating language(s) are classified with high probability while putting less emphasis on the classification accuracy of the minor languages. Although we are not really interested in the classification accuracy of the DNN but the performance of the bottleneck feature extractor, such imbalanced training may also make the bottleneck skewed towards the dominating language(s).

In order to make the training balanced across languages, we scale each example using a language-specific parameter α_i:

$$\alpha_i = \frac{\hat{N}}{N_i} \qquad \hat{N} = \frac{N}{|L|}$$

where N is the number of total frames in the multilingual dataset, N_i is the number of frames per language i and $|L|$ is the number of languages in the pool. $\hat{N}$ is thus the average number of frames per language. The scaler α_i is larger than one for under-represented languages and less then one for over-represented languages. During training, the log probability of each training example in the cross-entropy cost function is simply multiplied by the scaler:

$$\mathcal{F}(\theta) = -\sum_{i=1}^{K} \alpha_{l_i} \log p(y_i|x_i;\theta)$$

where K is the total number of examples, and l_i the language of the i-th example. Note that the sum of the scalers over all data is equal to the number of examples. This means that we shouldn't have to worry about changing the learning rate of training when we use the scaled examples.

To evaluate the method, we trained two bottleneck feature extractors on a dataset of six languages (see Table 1). The dataset is imbalanced: English, a random 800 hour subset of the Fisher corpus[1], has about 10 or more times data than the other five languages (all from the previous BABEL program periods). In one of the experiments we applied language-specific scalers to training examples to make the dataset balanced. Table 1 lists the final validation set frame accuracies of the trained DNNs. The results are as expected: balanced training decreases frame accuracy for the one language that is scaled down and increases it for the under-represented languages that are scaled up.

Next, we ported both bottleneck DNNs to five target languages, and built two separate hybrid DNN systems on top of the extracted features for each of the target languages, using the procedure described in section 2. The WER results are listed in Table 2. For comparison, we also list the WERs of the systems with a monolingual bottleneck feature extractor. All the multilingual systems perform better than the baseline monolingual systems. The differences between using the unbalanced and balanced feature extractors are small. The largest difference is for Javanese, where the balanced system gives a 0.8% absolute improvement in WER.

[1] LDC2004S13, LDC2004T19, LDC2005S13, LDC2005T19

Table 1: *Amount of training data per language, language scaler, and final validation set frame accuracies for imbalanced and balanced bottleneck DNN training.*

Language	*Hours*	*Scaler*	*Imbalanced Frame Acc*	*Balanced Frame Acc*
English	798	0.24	**43.1**	40.3
Turkish	88	2.14	37.4	**38.4**
Lao	85	2.21	46.7	**48.2**
Haitian	80	2.35	43.4	**45.1**
Lithuanian	41	4.53	36.6	**37.9**
Telugu	40	4.63	38.8	**39.9**

Table 2: *WER results for the systems with monolingual, unbalanced multilingual and balanced multilingual bottlenecks.*

Language	Monolingual	*Unbalanced multilingual*	*Balanced multilingual*
Amharic	45.0	**43.0**	43.3
Guarani	47.5	**45.0**	45.1
Igbo	57.1	54.7	54.7
Javanese	55.7	53.1	**52.3**
Mongolian	50.7	48.7	48.7

Although it is evident that the balanced system does not give significant improvements across all languages, we claim that it reduces the risk of having a multilingual system unsuitable for some of the target languages, if the variety of the languages in the multilingual pool is high enough. Also, it is possible that the balancing method is actually too aggressive and it has a negative impact on the robustness of the DNN: in our experiment, some of the languages received a scaler of over 4.5, while English was suppressed using a scaler of 0.24 (see Table 1). It is possible to reduce the magnitude of the scaling factors by introducing a constant $k < 1$, as proposed in [8]: $\alpha_i = \left(\frac{\hat{N}}{N_i}\right)^k$. For example, setting $k = 0.5$ would bring the scalers of our experiment closer to one, from the range of [0.24, 4.5] to [0.49, 2.2]. Experimenting with the magnitude is subject of our future work.

4. Porting multilingual stacked DNN acoustic models

This section describes our approach to training and porting multilingual stacked DNN acoustic models (DNN-AMs).

In this experiment, 21 languages were used for training multilingual models: 17 languages from the previous Babel program periods (Turkish, Pashto, Tagalog, Cantonese, Vietnamese, Assamese, Bengali, Haitian, Lao, Zulu, Tamil, Cebuano, Kazakh, Kurdish, Lithuanian, Telugu, Tok Pisin), with 40-80 hours per language, and four non-Babel languages: Arabic (Levantine Arabic QT training data[2], 250 hours), Spanish (Fisher, CallHome and Hub5[3], 250 hours), Mandarin (HKUST, CallHome and Hub5[4], 250 hours) and English (a random 250-hour subset of the Fisher corpus).

Table 3: *WER results of various mono/multilingual systems for Swahili. Unported multilingual DNN-AM (*) includes a softmax output layer trained on target language data.*

BN	*DNN-AM*	*WER*
Monolingual	Monolingual	41.9
Unported multilingual	Monolingual	41.1
Ported multilingual	Monolingual	39.6
Unported multilingual	Unported multilingual*	41.1
Ported multilingual	Unported multilingual*	40.6
Unported multilingual	Ported multilingual	39.8
Ported multilingual	Ported multilingual	**38.7**
Ported multilingual	Mono/multi DPET [25]	39.3

4.1. Training multilingual stacked DNN-AMs

After training a multilingual BN extractor, we dump the BN features for all training languages. Next, we train a GMM-HMM system with SAT for each language. As in monolingual training, the GMM system built on bottleneck features is only needed for obtaining fMLLR transforms. The fMLLR transforms are estimated independently for each language. Next, the second multilingual DNN is trained on the fMLLR-transformed multilingual bottleneck features. We initialize the weights of the multilingual DNN-AM using RBM-based unsupervised pre-training. The architecture of the multilingual DNN-AM is similar to our monolingual DNN-AM, only the hidden layer before the block-softmax is replaced with a dimension-reducing p-norm non-linearity [24] with 3500 input units and 350 output units, as opposed to 2048 sigmoid units that is used in the monolingual DNN-AM. Since the block-softmax layer of a multilingual DNN-AM is very large (around 100 000 in our experiments), this change greatly reduces the number of parameters of the final hidden layer and thus the whole multilingual DNN-AM and makes DNN training much faster.

The strategy of porting the multilingual stacked hybrid model to the target language is similar to porting of stacked bottleneck feature extractors [3]. First, we fine-tune the bottleneck DNN, as described in Section 2. The fine-tuned BN-DNN is used for extracting bottleneck features for the target language. A GMM system is trained on the features to obtain the fMLLR transforms for the target language, which are used for transforming the input features for the second DNN. Next, we initialize the softmax of the second DNN, train it for two epochs, and finally train the whole DNN for four epochs, using a smaller learning rate.

The described porting procedure was verified using Swahili (IARPA-babel202-B-v1d) as a development language. Table 3 lists WERs of several cross-entropy trained systems that differ in their degree on multilingual training and porting to target language. It is worth noting that a system with unported multilingual features and unported multilingual DNN-AM, having only the softmax layer trained for the target language, achieves better WER (41.1%) than the pure monolingual system (41.9%). The results also confirm that although the multilingual DNN-AM is trained on unported multilingual data, and porting the feature extractor to target language prior to porting the DNN-AM might seem counter-intuitive, such porting strategy decreases the WER of the system. A similar pattern has been noticed when porting stacked bottleneck models [3]. Furthermore, it is surprising that porting the bottleneck model helps (41.1 → 40.6) even when the multilingual DNN-AM is not ported and only the final softmax is language-specific.

[2] LDC2006S29, LDC2006T07

[3] LDC2010S01, LDC2010T04, LDC96S35, LDC96T17, LDC96L16, LDC98S70, LDC98T27

[4] LDC2005S15, LDC2005T32, LDC96S34, LDC96T16, LDC96L15, LDC98S69, LDC98T26, LDC96L15

Table 4: *WER/ATWV results of various monolingual and multilingual systems and their KWS combinations. Mono/multi DNN-AM is trained using the diveristy penalizing ensemble method [25] from both monolingual and multilingual initializations.*

	Bottleneck	DNN-AM	*Amharic*	*Dholuo*	*Guarani*	*Igbo*	*Javanese*	*Mongolian*	*Pashto*
A	Mono	Mono	44.0/0.594	39.0/0.618	46.3/0.551	56.0/0.342	55.2/0.437	49.8/0.487	48.1/0.419
B	Multi	Mono	41.4/0.631	36.8/0.653	43.3/**0.599**	53.0/0.401	51.6/0.485	46.9/0.530	45.6/0.454
C	Multi	Multi	41.7/0.631	36.7/**0.659**	43.3/0.596	53.3/0.394	**51.2/0.494**	47.4/0.523	45.4/0.465
D	Multi	Mono/multi	**41.3/0.633**	**36.6**/0.651	**43.0**/0.598	**52.7/0.402**	**51.2**/0.491	**46.8/0.531**	**45.1/0.465**
	A+B		0.641	0.658	0.604	0.403	0.497	0.541	0.463
	A+B+C		0.647	0.667	0.609	0.407	0.512	0.548	0.475
	A+B+C+D		0.648	0.668	0.609	0.410	0.514	0.549	0.476

4.2. Ensemble training from monolingual and multilingual initialization

In previous work we have established that for training data sizes similar to the ones used in this experiment (40 hours per language), layer-wise unsupervised pre-training gives significant improvement in WER and keyword search performance, as opposed to starting from randomly initialized weights or using layer-wise discriminative pre-training. While the porting of the multilingual DNN acoustic model to the target language can exploit the high level phonetic features that emerge during multilingual training, it cannot take advantage of hierarchical patterns discovered using pre-training on target language data. However, unsupervised layer-wise pre-training and porting the multilingual model are very similar in how they work: they are both different forms of weight initialization, and they both start with a randomly initialized softmax layer. Therefore we also experimented with training two DNNs with the different initializations jointly, using a diversity-penalizing ensemble training (DPET) method [25]. This method uses an objective function that encourages the models to produce similar values across their output space, letting the models thereby learn from each other. Note that although the two models are trained as an ensemble, only one of them is used in decoding. As the models are trained to produce similar outputs, the final models can be used interchangeably. In our experiments, we used the model that was initialized with pre-training on target language data, as its frame accuracy on held-out data was slightly higher on most languages. We also observed that DPET resulted in consistently lower heldout data log likelihoods and higher frame accuracies than training individual models either from pre-trained or multilingual initialization. Performance of the model trained using the ensemble method on Swahili is also given in Table 3.

4.3. Results on the main test languages

Based on the Swahili results, we concentrated our further experiments on three multilingual stacked hybrid models. In all models, we used the multilingual bottleneck features ported to the target language as input. The DNN-AM was based either on monolingual unsupervised pre-training, ported from the multilingual DNN-AM, or trained from both initializations using DPET. The speech recognition and keyword search performance was tested on all seven BABEL OP3 languages. Contrary to the development experiments on Swahili, we also applied sMBR-based sequence-discriminative training [26] to the models for further improvement. Tabel 4 lists the WER and ATWV results of the monolingual model and the three multilingual models with differently trained DNN-AMs. We observe that all multilingual models give large gains over the monolingual baseline. The multilingual model trained using the ensemble method results in the best WER for all seven languages, although the differences are quite small. For keyword spotting, the results are more varied, although the jointly trained model is again the best or very close to the best for all languages.

We also report ATWV results of different system combinations, using the hitlist combination technique [22]. Instead of showing all 4x4 system combination results for each of the languages, we report the results of a realistic scenario when a monolingual system is trained first, followed by different multilingual systems. It is not surprising that combining a pure monolingual model with a system using multilingual features (*A*+*B*) gives 0.5-3% relative gain over the latter (*B*). However, adding the model with a multlingual DNN-AM to the mix (*A*+*B*+*C*) results in additional 1-3% relative increase in ATWV, suggesting that the multilingual models with different DNN-AMs are somewhat complementary. As the jointly trained model is aimed to combine the benefits of multilingual and monolingual models, adding it to the combination (*A*+*B*+*C*+*D*) gives almost no gains over the three-way combination.

5. Conclusions

The paper explored scaling the examples for training multilingual DNN models, in order to equalize the contribution of each language. Although this approach gave a small gain in WER on the average, there was no significant difference for most tested languages. The paper also experimented with multilingual training of the stacked DNN architecture where the first model acts as a feature extractor and the second is used as the acoustic model. Contrary to previous works, we port the second DNN from the multilingual model. An ensemble training method, where the second DNN is jointly trained from multilingual and monolingual initializations, was shown to produce consistent gains in WER over the model where only the feature extractor is multilingual.

Although the paper experimented with stacked DNNs, the methods of balancing language contributions and training a DNN jointly from both monolingual and multilingual initializations can also be applied to simple unstacked DNN-AMs.

6. 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.

7. References

[1] P. Golik, Z. Tüske, R. Schlüter, and H. Ney, "Multilingual features based keyword search for very low-resource languages," in *Interspeech*, 2015.

[2] J. Cui, B. Kingsbury, B. Ramabhadran, A. Sethy, K. Audhkhasi, X. Cui, E. Kislal, L. Mangu, M. Nußbaum-Thom, M. Picheny, Z. Tüske, P. Golik, R. Schlüter, H. Ney, M. J. F. Gales, K. M. Knill, A. Ragni, H. Wang, and P. C. Woodland, "Multilingual representations for low resource speech recognition and keyword search," in *ASRU*, 2015.

[3] F. Grézl and M. Karafiát, "Adapting multilingual neural network hierarchy to a new language," in *SLTU*, 2014.

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

[5] M. Harper, "The BABEL program and low resource speech technology," *ASRU*, 2013.

[6] Y. Zhang, E. Chuangsuwanich, and J. R. Glass, "Language ID-based training of multilingual stacked bottleneck features," in *Interspeech*, 2014.

[7] F. Grézl, E. Egorova, and M. Karafiát, "Study of large data resources for multilingual training and system porting," in *SLTU*, 2016.

[8] Y. Song, L.-P. Morency, and R. W. Davis, "Distribution-sensitive learning for imbalanced datasets," in *IEEE International Conference and Workshops on Automatic Face and Gesture Recognition*, 2013.

[9] A. Stolcke, F. Grezl, M.-Y. Hwang, X. Lei, N. Morgan, and D. Vergyri, "Cross-domain and cross-language portability of acoustic features estimated by multilayer perceptrons," in *ICASSP*, 2006.

[10] S. Scanzio, P. Laface, L. Fissore, R. Gemello, and F. Mana, "On the use of a multilingual neural network front-end," in *Interspeech*, 2008.

[11] S. Thomas, S. Ganapathy, and H. Hermansky, "Cross-lingual and multi-stream posterior features for low resource LVCSR systems." Interspeech, 2010.

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

[13] J.-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *ICASSP*, 2013.

[14] A. Ghoshal, P. Swietojanski, and S. Renals, "Multilingual training of deep neural networks," in *ICASSP*, 2013.

[15] G. Heigold, V. Vanhoucke, A. Senior, P. Nguyen, M. Ranzato, M. Devin, and J. Dean, "Multilingual acoustic models using distributed deep neural networks," in *ICASSP*, 2013.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] T. Ng, R. Hsiao, L. Zhang, D. Karakos, S. H. Mallidi, M. Karafiát, K. Vesely, I. Szoke, B. Zhang, L. Nyugen, and R. Schwartz, "Progress in the BBN keyword search system for the DARPA RATS program," in *Interspeech*, 2014, pp. 959–962.

[21] 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.

[22] D. Karakos, R. Schwartz, S. Tsakalidis, L. Zhang, S. Ranjan, T. Ng, R.-C. Hsiao, G. Saikumar, I. Bulyko, L. Nguyen *et al.*, "Score normalization and system combination for improved keyword spotting," in *ASRU*, 2013.

[23] J. G. Fiscus, J. Ajot, J. S. Garofolo, and G. Doddington, "Results of the 2006 spoken term detection evaluation," in *Proceedings of ACM SIGIR*, 2007, pp. 51–55.

[24] X. Zhang, J. Trmal, D. Povey, and S. Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks," in *ICASSP*, 2014.

[25] X. Zhang, D. Povey, and S. Khudanpur, "A diversity-penalizing ensemble training method for deep learning," in *Interspeech*, 2015.

[26] K. Veselỳ, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks." in *Interspeech*, 2013.