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

Two-Stage Data Augmentation for Low-Resourced Speech Recognition

William Hartmann, Tim Ng, Roger Hsiao, Stavros Tsakalidis, Richard Schwartz

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

Abstract

Low resourced languages suffer from limited training data and resources. Data augmentation is a common approach to increasing the amount of training data. Additional data is synthesized by manipulating the original data with a variety of methods. Unlike most previous work that focuses on a single technique, we combine multiple, complementary augmentation approaches. The first stage adds noise and perturbs the speed of additional copies of the original audio. The data is further augmented in a second stage, where a novel fMLLR-based augmentation is applied to bottleneck features to further improve performance. A reduction in word error rate is demonstrated on four languages from the IARPA Babel program. We present an analysis exploring why these techniques are beneficial.
Index Terms: speech recognition, deep neural networks, data augmentation

1. Introduction

When training data is limited—whether it be audio or text—the obvious solution is to collect more data from similar sources [1, 2]. If the language is not widely spoken, collecting additional resources may be difficult. Another alternative is to simulate data [3, 4]. A common tactic in the robustness community to improve performance on unseen noise and microphone conditions is to augment the original data [5]. Given the original data, additional copies are generated through random perturbations and through augmentation with additional signals.

Most previous work focuses on a single type of augmentation at a time: reverberation [6], noise addition [7], and speaker characteristics [8]. Some more recent work has combined multiple augmentation techniques. Ragni et. al [9], combined semi-supervised training and vocal tract length perturbation (VTLP). The recent ASpIRE Challenge [10] saw several teams achieve success with data augmentation. Peddinti et. al [11] combined reverberation and volume perturbation, and Hsiao et. al [12] added both artificial noise and reverberation to the original data. VTLP and a speaker-based augmentation were combined in two stages by Cui et. al [13].

We also combine multiple techniques at multiple stages to further improve performance. We use a two-stage approach to increase the types of augmentation and the efficiency with which they can be added to our standard pipeline. The acoustic data is augmented by adding noise and then perturbing the speed of the audio. Both techniques are simple to apply. Features are derived from the augmented data and used to train a bottleneck feature extractor. The bottleneck features are augmented in the second stage with a novel fMLLR-based approach. The two-stage approach provides an improvement over either technique individually and offers flexibility.

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.

2. Data Augmentation

Our goal is to produce a simple pipeline, where each additional copy of data is augmented in multiple ways. Two techniques described below: noise augmentation and speed perturbation. Other approaches were also tested, but did not produce a consistent gain in combination with other techniques.

2.1. Noise Augmentation

Noise augmentation has long been used to improve the robustness of acoustic models. Most of this work has been applied to GMMs [14, 15], but the same approaches can be applied to DNNs [16]. Some datasets have pre-specified multi-style training sets that have been artificially created [17]. Noise sources are added to the original data at a random SNR. This increases the variance of the resulting models. The motivation is to improve recognition in unseen conditions.

As the IARPA Babel project typically does not allow outside audio sources, we collected noise sources from the Babel data itself. Non-speech segments from the previous year's languages were identified based on speech activity detection. Since the data itself is often quite noisy, we assumed any non-speech segments could be valid candidates for noise sources. This method has the additional benefit of ensuring the noise data is similar to the conditions found during testing, though preliminary experiments with other noise sources produced similar results. The augmented datasets were generated by copying the clean data and adding a random noise sample at an SNR between 0dB and 20dB; several other SNR ranges were tested, but did not improve performance. Estimated SNR of the Babel training set ranges from 0dB to 50dB.

2.2. Speed Perturbation

Ko et al. [18] showed success by manipulating the speed of the data. They demonstrated a performance improvement over the more common vocal tract length perturbation (VTLP) technique [8]. Using the Sox utility [19], the original data is perturbed by a warping factor that effects both the frequencies and the duration of the speech. The speed change is accomplished by resampling the waveform, which not only changes the duration, but also scales the pitch, vocal tract length, and all spectral frequencies by the same factor. Our setup uses a randomly selected warping factor between 0.9 and 1.1 (this was also the

http://dx.doi.org/10.21437/Interspeech.2016-1386

range used in [18]). We experimented both with other ranges and using discrete steps, but those did not perform as well.

3. Speaker-Based Augmentation

This first stage of augmentation affects the original audio and is pushed through feature extraction all the way to the final system. Previous work has also experimented with using the augmented data during bottleneck feature training only [20]. Another approach from previous work is to augment the speaker-adapted features [21]. We explore this technique at a second stage—after bottleneck features have been trained. The usual motivation is to simulate the speech coming from a different speaker.

One recent technique shown to be successful on Babel data is Stochastic Feature Mapping (SFM) [21]. Two linear transformations are applied to the features. Speaker-dependent models are estimated for all training speakers. The parameters of the training speaker are first mapped to the feature space of a random other speaker. Then the second transformation maps these features to the canonical speaker-independent space. The goal is to simulate the features as having come from a different speaker.

SFM is a well-motivated technique for augmentation, but it is unclear if the gain is actually from mapping the features to another speaker's space. It is possible the gains seen from SFM are due solely to the perturbations added to the features—increasing the robustness of the model—and not from the simulation of additional speakers. In our pipeline, SFM is also an expensive technique as a separate acoustic model must be generated for each speaker.

We propose a simpler approach that only perturbs the final speaker-adapted features without additional computation. For the additional data, only a small change is made to the system pipeline. When applying the fMLLR transform to create the speaker-adapted features, a random speaker's transformation is used instead of the true speaker's transformation. Our motivation is that this simulates the imperfect fMLLR transformations that can be derived during decoding from inaccurate automatic transcriptions. Regardless of the motivation, this fMLLR-based augmentation (FBA), provides a realistic perturbation of the features. The detailed algorithm is shown in Algorithm 1.

Algorithm 1 fMLLR-based Speaker Augmentation (FBA)

Input: set of speaker transformations S, number of speakers n.
  Let D be an n x n similarity matrix.
  for $i = 1, \ldots, n$ **do**
    for $j = 1, \ldots, n$ **do**
      $D_{i,j} \leftarrow \exp\left(\frac{-\|S_i - S_j\|^2}{2\sigma^2}\right)$
    end for Let $N = \Sigma_j D_{i,j}$
    for $j = 1, \ldots, n$ **do**
      $D_{i,j} \leftarrow D_{i,j}/N$
    end for
    Select index k based on the distribution $D_{i,\cdot}$
    $S'_i \leftarrow S_k$
  end for
Output: new set of speaker transformations S'.

We test several variations of this approach that differ only in how we select the random speaker. For each speaker, we compute the Euclidean distance between the fMLLR transformation from every other speaker. Our informal listening tests confirm this distance is reasonable. Similar transformations come from similar speakers and transformations with a large distance typically come from a speaker of a different gender in a different condition. Given the distances, they still need to be transformed into a similarity. We use the same approach commonly used in spectral clustering [22], the Gaussian similarity function

$$sim(A, B) = \exp\left(\frac{-\|A - B\|^2}{2\sigma^2}\right) \quad (1)$$

The σ value controls the width of the distribution. As σ decreases, dissimilar points move further apart. The similarities are then normalized to create a probability distribution. Figure 1 illustrates the effect of the σ on the cumulative distribution. For instance, with $\sigma = 0.1$, the 10% most similar matrices cover 75% of the probability space. Also note that in this case, the identity matrix covers nearly 40% of the distribution. Now that we can generate a distribution over the matrices, we can select a random speaker based on that distribution. The other option we explore is selecting a random speaker uniformly.

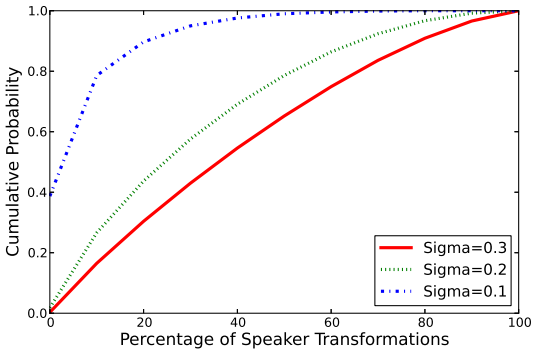

Figure 1: Cumulative probability of selecting a speaker transformation given σ value. Assumes transformations are sorted in terms of similarity.

4. Experimental Setup

We use the Sage ASR toolkit [23]. 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 [24], with open source toolkits, such as Kaldi [25], CNTK [26] and Tensorflow [27]. For example, DNN can be trained using Byblos, Kaldi nnet1 [28] or nnet2, CNN using Kaldi or Caffé [29], and LSTM using Kaldi or CNTK. Sage also includes keyword search from Byblos [30]. The integration of these technologies is achieved through wrapper modules around major functional blocks that can be easily connected or interchanged. Sage also includes a cross-toolkit FST recognizer that supports models built using the various component technologies.

All experiments are performed on data from the IARPA Babel project. 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). Amharic is used as our development language to test augmentation approaches and setups. For each language, the full language pack (FLP) is used, containing approximately 40 hours of transcribed audioLexicons are derived using simple G2P rules [31]. Trigram language models are built only from the transcriptions. Decoding is performed on an additional 10 hours of development data.

Language	Augmentation Type x Copies	WER
Amharic	none	44.2
Amharic	Speed x 2	44.0
Amharic	Noise x 2	43.4
Amharic	Reverb x 2	43.8
Amharic	Speed x 1, Noise x 1	43.5
Amharic	(Speed+Noise) x 2	42.8
Amharic	(Speed+Noise+Reverb) x 2	43.4

Table 1: Comparison of multiple augmentation types and combinations on Amharic. All augmented models use a total of two additional copies of the data.

Language	Model Type	Copies	WER
Amharic	DNN	0	44.2
Amharic	DNN	1	43.4
Amharic	DNN	2	42.8
Amharic	CNN	0	45.0
Amharic	CNN	1	44.1
Amharic	CNN	2	43.8

Table 2: Comparison of CNN and DNN models using augmented data. Zero copies refers to the baseline system. Additional copies are both noise augmented and speed perturbed.

In previous years, actual term-weighted value (ATWV) was the primary measure of interest for IARPA Babel program. This year reduction in WER has been added as a goal. Although—due to space constraints—we present results on WER only, the gains on ATWV are generally larger than the gains in WER, because word spotting depends more on model robustness.

5. Results

In order to determine the best combination of augmentation types, we first test them on Amharic. While the IARPA Babel data itself is not reverberant, we also tested adding artificial reverberation in addition to noise and speed augmentation. A set of artificial room impulse responses (RIR) were generated [32]. Using these RIR the data was artificially reverberated.

The results in Table 1 contain a subset of combinations that were tested. All techniques produce a small gain when used individually, but the combinations are mixed. Best performance is obtained by combining noise and speed augmentation. For the remainder of the paper we only consider this combination for our first stage data augmentation. A wide range of additional variations can be applied using the previously discussed techniques: varying SNR of added noise, varying the speed factor, separating the augmentations between copies, etc. More variations were tried than could be reported in this work, but they all either resulted in the same or decreased performance.

It has been previously shown that CNNs may be more resilient to noise and channel variation [33]. We test whether this translates to improved performance with data augmentation. Table 2 presents results on Amharic using both DNN and CNN acoustic models. Our CNN setup is similar to the described DNN setup, except that the top first layers of the DNN are replaced by convolutional layers, and the entire system is trained on filter bank features as opposed to bottleneck features. The baseline CNN performs worse than the DNN, likely due to the speaker-adapted features used by the DNN. Both models see a similar improvement in WER from the data augmentation, but the absolute performance is still better with the DNN—though it does demonstrate the results are not dependent on the model.

Language	Baseline	One Copy	Two Copies
Amharic	44.2	43.4	42.8
Guarani	46.7	45.6	45.2
Igbo	55.5	54.5	54.3
Pashto	48.1	46.8	47.1

Table 3: Results using the first stage noise and speed augmentation during training for four languages.

Language	None	SFM	FBA; Random	FBA; $\sigma = 0.2$
Amharic	42.8	42.6	42.4	42.2
Guarani	45.2	44.7	44.9	44.6
Igbo	54.3	54.0	54.1	53.9
Pashto	47.1	46.7	46.8	46.7

Table 4: Results for applying speaker-based augmentation on top of the noise and speed augmentation reported in Table 3. None refers to just using noise augmentation and speed perturbation. SFM [21], and the two FBA approaches use the additional second stage augmentation.

All further experiments use DNN acoustic models since they give better performance. It is also simpler to apply the second stage augmentation when using the DNN.

Table 3 shows results for noise+speed augmentation on four Babel languages. In all cases, the languages see a significant reduction in WER from the augmentation. Three of the four languages benefit from the addition of a second copy of data. We also tested adding additional copies of data beyond two for Amharic, but this produced no further gains. This first stage of augmentation reduces absolute WER from 1% to 1.5% for the four languages with two copies of augmented data.

The second stage of augmentation, speaker-based augmentation, modifies the speaker-adapted bottleneck features. Results are shown in Table 4 for four languages. Note that the *None* result still uses two copies of data that are noise+speed augmented. Three variations of speaker-based augmentation are compared against results using only the first stage of augmentation. The value of σ was selected to be 0.2 as that produced the best performance with our preliminary Amharic experiments.

In the best case, FBA decreased WER a further 0.6%, providing a total absolute improvement over the baseline of 2.1%. In all cases, the addition of FBA further improves results, but the random selection performs similarly whether it is uniform or based on a similarity with the current speaker. The SFM approach also gives similar results, but requires additional computation. While the additional gains provided by FBA are small, they come with no additional training cost. On average, the first stage reduces WER by 1.2% and the second stage produces an additional reduction of 0.5% absolute.

6. Analysis

It is not obvious why the augmentation produces gains. The noise augmentation uses noise from the same corpora and we do not expect a large mismatch between the training and testing conditions—the greater the mismatch, the greater the gain expected from data augmentation. The FBA approach to speaker-based augmentation produces similar gains as the SFM approach, but without the motivation of simulating additional speakers. We further analyze the results below.

The results for the development data can be further broken down based on recording condition. Seven conditions are listed

for each language: *car kit, home office landline, home office mobile, microphone, public, street, and vehicle*. Since *home office landline* is a controlled location, it is unexpected to see large gains. The *microphone* condition uses a far field microphone and is the most challenging condition. Remaining conditions are mobile phones used in a variety of environments.

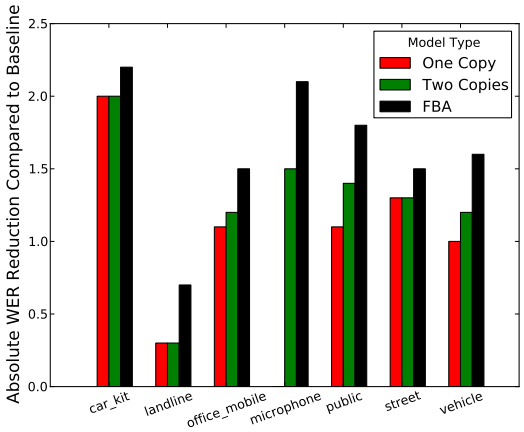

Figure 2: Relative WER reduction versus the baseline system. Single stage augmentation approaches using one and two copies of augmented data, and the two stage FBA approach are shown. Results are broken down based on recording condition.

Figure 2 shows the average improvement in WER for the augmented systems over the baseline averaged over all four languages. The gains from noise+speed augmentation are spread evenly through most conditions; all but *home office landline* see at least a 1% absolute reduction in WER. The major outlier is the *microphone* result for the system trained on one copy of additional data; it sees no gain. This discrepancy is alleviated when the second copy of data is added. However, nearly all of the gain from adding the second copy of data comes from the *microphone* condition. This helps explain why Pashto did not see an improvement from the second copy. It is the only language without any *microphone* data in the the development set. Based on this analysis it appears the first stage of augmentation improves performance across all conditions, but additional copies are required to improve performance on the more difficult *microphone* condition. The second stage of augmentation gives a consistent gain across conditions.

In order to better understand how the augmented data is helping, we decoded the training sets for Amharic. The three training sets—the original data plus the two augmented copies—are kept separated. First we look at performance using the GMM models. Figure 3 shows results for each system on the three training sets. Note that the FBA model is not shown as the second stage of augmentation is not applied before GMM training. The baseline model sees a drastic reduction in performance when tested on the augmented data. The models trained on the augmented data show improvements on the augmented data, but performance on the original data is not affected.

Figure 4 shows similar experiments with the associated DNN models. Again, the baseline model clearly has trouble dealing with the augmented data, while the models trained on the augmented data see large reductions in WER. These results are to be expected when testing on the training data. The surprising result is that training on augmented data improves performance on the original data–the typical motivation for training on augmented data is to improve performance on unseen conditions. This is a significant difference in the effects of training on augmented data for GMM and DNN models.

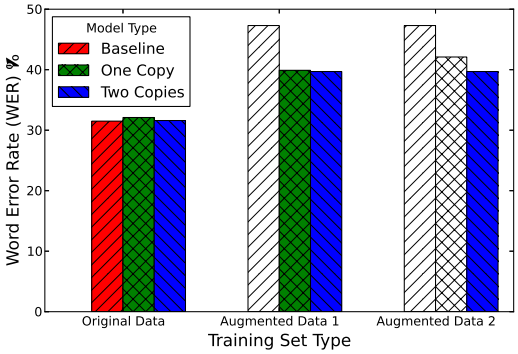

Figure 3: WER results on Amharic training sets using GMMs. If the model has seen the data in training, the bar is filled.

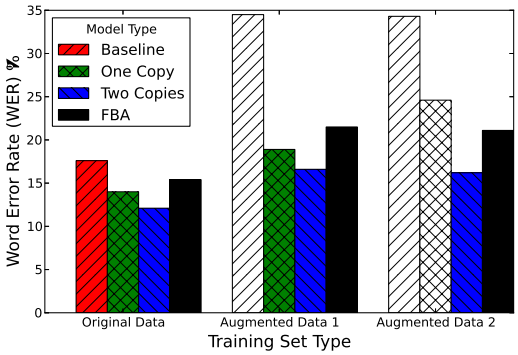

Figure 4: WER results on Amharic training sets using DNNs. If the model has seen the data in training, the bar is filled.

Adding the second stage of augmentation, speaker-based augmentation, degrades performance on the three training sets. It is still better than the baseline model, but significantly worse than the model using only the noise+speed augmentation. Since it does improve performance on unseen data, it seems likely it is performing a function similar to regularization. In future experiments we will compare the FBA technique to other standard regularization approaches.

7. Conclusions

We presented a two-stage approach to data augmentation. The first stage combines previously proposed techniques to add noise and speed perturbation. This first stage augmentation is used to train all stages of our system. After bottleneck features have been trained, a second stage of augmentation is used. Bottleneck features are augmented by using a random speaker's fMLLR transformation. In all cases the first stage provides significant gains in performance. The second stage produces a further reduction in WER. Additional analysis further helps explain why the augmentation process produces such gains.

8. References

[1] 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.

[2] S. Novotney and C. Callison-Burch, "Cheap, fast, and good enough: Automatic speech recognition with non-expert transcription," in *HLT-NAACL*, 2010.

[3] M. Gales, A. Ragni, H. AlDamarki, and C. Gautier, "Support vector machines for noise robust ASR," in *ASRU*, 2009, pp. 205–210.

[4] A. Jensson, "Development of a speech recognition system for icelandic using machine translated text," in *SLTU*, 2008.

[5] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 745–777, April 2014.

[6] A. Sehr, C. Hofmann, R. Maas, and W. Kellermann, "Multistyle training of HMMs with stereo data for reverberation-robust speech recognition," in *Joint Workshop on Hands-free Speech Communication and Microphone Arrays*, 2011, pp. 196–200.

[7] D. Yu, M. Seltzer, J. Li, J. Huang, and F. Seide, "Feature learning in deep neural networks - a study on speech recognition tasks," in *International Conference on Learning Representations*, 2013.

[8] N. Jaitly and G. E. Hinton, "Vocal tract length perturbation improves speech recognition," in *ICML*, 2013.

[9] A. Ragni, K. Knill, S. Rath, and M. J. F. Gales, "Data augmentation for low resource languages," in *Interspeech*, 2014, pp. 810–814.

[10] M. Harper, "The automatic speech recognition in reverberant environments (ASpIRE) challenge," in *ASRU*, 2015.

[11] V. Peddinti, G. Chen, V. Manohar, T. Ko, D. Povey, and S. Khudanpur, "JHU aspire system: Robust LVCSR with TDNNs, ivector adaptation and RNN-LMs," in *ASRU*, 2015.

[12] R. Hsiao, J. Ma, W. Hartmann, M. Karafiát, F. Grézl, L. Burget, I. Szoke, J. Cernocky, S. Watanabe, Z. Chen, S. H. Mallidi, H. Hermansky, S. Tsakalidis, and R. Schwartz, "Robust speech recognition in unknown reverberant and noisy conditions," in *ASRU*, 2015.

[13] X. Cui, V. Goel, and B. Kingsbury, "Data augmentation for deep convolutional neural network acoustic modeling," in *ICASSP*, 2015.

[14] J. Li, M. Seltzer, and Y. Gong, "Improvements to vts feature enhancement," in *ICASSP*, 2012.

[15] J. Droppo and A. Acero, "Maximum mutual information SPLICE transform for seen and unseen conditions," in *Interspeech*, 2005, pp. 989–992.

[16] M. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *ICASSP*, 2013, pp. 7398–7402.

[17] N. Parihar and J. Picone, "Analysis of the Aurora large vocabulary extensions," in *Proceedings of Eurospeech*, vol. 4, Geneva, Switzerland, September 2003, pp. 337–340.

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

[19] "SoX, audio manipulation tool," http://sox.sourceforge.net.

[20] Z. Tüske, P. Golik, D. Nolan, R. Schluter, and H. Ney, "Data augmentation, feature combination, and multilingual neural networks to improve ASR and KWS performance for low-resource languages," in *Interspeech*, 2014.

[21] X. Cui, V. Goel, and B. Kingsbury, "Data augmentation for deep neural network acoustic modeling," in *ICASSP*, 2014.

[22] U. V. Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[23] 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 *submission to Interspeech*, 2016.

[24] Y. Chow, M. Dunham, O. Kimball, M. Krasner, G. Kubala, J. Makjoul, P. Price, S. Roucos, and R. Schwartz, "BYBLOS: The BBN Continuous Speech Recognition System," in *ICASSP*, 1987.

[25] 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 *Proceedings of IEEE ASRU*, 2011.

[26] A. Agarwal, E. Akchurin, C. Basoglu, G. Chen, S. Cyphers, J. Droppo, A. Eversole, B. Guenter, M. Hillebrand, R. Hoens, X. Huang, Z. Huang, V. Ivanov, A. Kamenev, P. Kranen, O. Kuchaiev, W. Manousek, A. May, B. Mitra, O. Nano, G. Navarro, A. Orlov, M. Padmilac, H. Parthasarathi, B. Peng, A. Reznichenko, F. Seide, M. L. Seltzer, M. Slaney, A. Stolcke, Y. Wang, H. Wang, K. Yao, D. Yu, Y. Zhang, , and G. Zweig, "An introduction to computational networks and the computational network toolkit," Microsoft, Tech. Rep. MSR-TR-2014-112, August 2014.

[27] 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/

[28] K. Vesely, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *Interspeech*, 2013, pp. 2345–2349.

[29] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[30] 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.

[31] 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.

[32] E. Habets, "Room impulse response generator," Technische Universiteit Eindhoven, Tech. Rep., 2006.

[33] V. Mitra, W. Wang, H. Franco, Y. Lei, C. Bartels, and M. Graciarena, "Evaluating robust features on deep neural networks for speech recognition in noisy and channel mismatched conditions," in *Interspeech*, 2014.