Investigation on Bandwidth Extension for Speaker Recognition

Phani Nidadavolu, Cheng-I Lai, Jesús Villalba, Najim Dehak

Center for Language and Speech Processing, The Johns Hopkins University

{snidada1,clai24,jvilla17,ndehak3}@jhu.edu

Abstract

In this work, we investigate training speaker recognition systems on wideband (WB) features and compare their performance with narrowband (NB) baselines. NIST speaker recognition evaluations have mainly driven speaker recognition research in the past years. Because of the target application of these evaluations, most data available to train speaker recognition systems is NB telephone speech. Meanwhile, WB data have been more scarce not being enough to train factor analysis and PLDA models. Thus, the usual practice when dealing with WB speech consists in downsampling the signal to 8 kHz, which implies potential loss of useful information. Instead, we experimented upsampling the training telephone data and leaving the WB data unchanged. We adopt two techniques to upsample telephone data: (1) using a feed-forward neural network, termed Bandwidth Extension (BWE) network, to predict WB features given NB features as input; and (2) using basic upsampling with a low-pass filter interpolator. While the former intends to estimate the high frequency information, the latter does not. The upsampled features are used to train state-of-the art i-vector and recently proposed x-vector models. We evaluated the systems on Speakers In The Wild (SITW) database obtaining 11.5% relative improvement in detection cost function (DCF) with x-vector model.

Index Terms: deep neural network, bandwidth extension, speaker recognition, i-vector, x-vectors

1. Introduction

Solving sampling mismatch without information loss has been a research topic in speech. Different devices record speech at different sampling rates and thus create a mismatch later on while training speech models. With regard to the paradigm of training speaker recognition models, a considerable amount of speech data is recorded at 8 kHz, mostly telephone conversations. We will refer to these data as narrowband speech, NB, in the context. On the other hand, a limited amount of speech is recorded at 16 kHz, such as far field microphone speech. We will refer to these data as wideband speech, WB. The traditional approach to solving sampling rate mismatch is restricted to downsampling WB data to match the sampling rate of NB data. However, downsampling operation degrades performance of speech models as it throws away information that could potentially be meaningful later on. Therefore, bridging the gap between sampling mismatch and information loss could increase the quality of training data and potentially enhance the result of the model task. Two main approaches in literature have attempted to address this issue: (1) bandwidth extension (BWE) techniques and (2) mixed BW networks.

Early work showed that WB spectrum can be predicted from extending NB spectral envelope by a linear model and exciting it with white noise [1]. The linear model poses the assumption that speech is relatively smooth and linear in frequency domain. Most recent work focused on exploiting the capability of deep neural net (DNN) given its success in several tasks in speech processing and analysis, and DNN-based speech BWE has demonstrated improvement in automatic speech recognition (ASR). A feed-forward DNN trained with log spectrogram features of NB and WB data was incorporated in an ASR system [2]. The authors were able to show that DNN is capable of extending BW of a signal, and that these features when used to train a downstream task like ASR would give better performance compared to system trained only on NB data. Multi-task learning and transfer learning were explored as a means of assisting multi-lingual task and cross-lingual task in [3]. Their BWE was trained on bandlimited WB data and further retrained on NB data and achieves a subsequent 45% relative WER reduction.

An alternative approach is to modify the NB features by applying some transformation on them to match some specific properties of WB data and then train a neural network, along with the available WB data, to perform the task of interest. Authors in [4] trained a mixed BW ASR on log-mel filter banks. Authors uses 22 and 29 dimensional filter banks for NB and WB data respectively. The filters are designed such that the first 22 filters of WB data are aligned with that of NB data. The NB features are zero padded (transformed) to match the dimension of WB features. The neural connections of the network are optimized to learn from the first 22 filter banks for the NB data and the entire feature vector for the WB data.

All the techniques discussed so far are developed for ASR applications. To our knowledge, there is not yet any literature that has applied similar approaches to speaker recognition. In this work, we present two independent case studies: (1) training the state-of-the art i-vector speaker recognition system [5] on BW extended speech and (2) training the recently introduced x-vector [6, 7] model on mixed BW speech. In the former we study the impact of BW extension on the performance of the model when tested on WB data. The later study should be seen as a precursor to our interest in training mixed BW systems in end-to-end fashion (without using a separate BW extension network). In this approach, instead of using a separate network to do bandwidth extension, we intend to evaluate whether the x-vector network can learn to work with NB and WB at the same time. To achieve this, we upsampled the NB audio to WB using standard interpolation with a low-pass filter. We have observed encouraging results with this approach (11.5% relative improvement in detection cost function (DCF) on Speakers In The Wild (SITW) [8] database).

The outline of the paper is as follows. Section 2 introduces the state-of-the-art speaker recognition system used in our experiments. Section 3 describes the upsampling techniques used in this work. Section 4 presents our experimental setup. Results are discussed in Section 5 followed by conclusions in Section 6.

2. Speaker Recognition Systems

This section describes the two main speaker recognition systems used in this work, i-vector and x-vector models. Both systems were built using the Kaldi speech recognition toolkit [9].

2.1. i-vector system

The i-vector extractor [5] transforms the recording feature sequence into a fixed-dimensional embedding. To do it, each speech segment is modeled by a Gaussian mixture model (GMM) whose super-vector mean $\mathbf{M}$ is assumed to be

$$\mathbf{M}_s = \mathbf{m} + \mathbf{T}\mathbf{w}_s \tag{1}$$

where $\mathbf{m}$ is the GMM-UBM means super-vector (speaker-independent mean), $\mathbf{T}$ is a low-rank matrix and $\mathbf{w}$ is a standard normal distributed vector. $\mathbf{M}$ defines the total variability space, i.e. the directions in which we can move the UBM to adapt it to a specific segment. The *maximum a posteriori* (MAP) point estimate of $\mathbf{w}$ is the i-vector embedding. We used a 2048 component UBM with 600 dimensional i-vectors.

2.2. x-vector system

Recent works [6, 7] introduced a successful neural network architecture to map sequences into speaker discriminant fixed-length vectors. The authors denominated these embeddings as x-vectors. The network receives a sequence of feature frames, which are processed by several layers. The result is summarized by a pooling layer that computes mean and standard deviation over time. Mean and standard deviation are concatenated together and propagated to the output through a series of feed-forward layers. The output is a dense layer with softmax activation predicting the speaker posteriors. Before the pooling layer, we used a time delay neural network (TDNN) (a.k.a. 1D convolutions). The sequence embedding is extracted from the first affine transform after the pooling layer (before applying the non-linear activation).

The results in [10] indicate that x-vector can outperform i-vectors and be robust across datasets. However, x-vector modelling is a data greedy approach and is able to beat i-vector models when presented with large amounts of training data. Authors in [10] used various data augmentation techniques to overcome this problem.

2.3. PLDA

We use full-rank probabilistic discriminant analysis [11] as our back-end to test both the x-vector and i-vector models. They were centered and projected to a lower dimensional space using LDA. LDA dimension was set to 150. All the vectors are length normalized and log-likelihood ratios are evaluated using the back-end. Finally, scores were normalized using adaptive symmetric norm (S-Norm) [12].

3. Upsampling Techniques

This section describes the two main upsampling techniques we used in our work.

3.1. Bandwidth Extension Network Training Procedure

We used a feed-forward Deep Neural Network (DNN) for BW extension. Since our goal was to predict WB features given NB features as input, we need to have matched WB and NB feature pairs for training. For this purpose, we took real WB data and we downsampled it to 8 kHz. Then, the latter were used as input to the network and the former as the target output. Similar to [2], we used log-power spectrum (LPS) features as input and output. The input spectrogram had dimension 129 and the output dimension 257, including the offset component. A context of 5 past and future frames was used to predict the WB spectrogram of the current frame. Hence the input to the network is 1419 dimensional and the output was of 257 dimensional. Utterance level mean variance normalization was applied on both the input and output of the network before training.

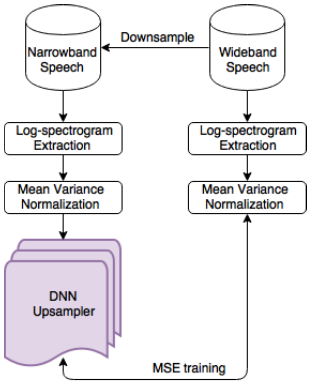
Narrowband Speech
Downsample
Wideband Speech
Log-spectrogram Extraction
Log-spectrogram Extraction
Mean Variance Normalization
Mean Variance Normalization
DNN Upsampler
MSE training

Figure 1: *The DNN-based BWE Training Pipeline*

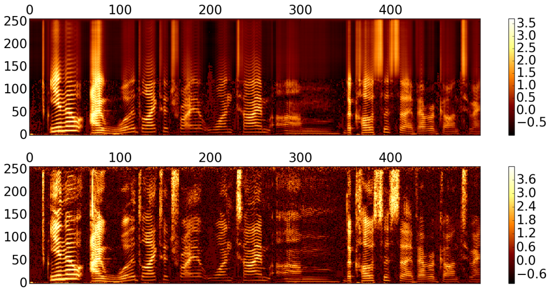

Figure 2: *log-spectrogram comparison of estimated WB from BWE (upper) and real WB (lower)*

The network had 3 hidden layers with 2048 neurons per layer. Rectified linear unit (ReLu) nonlinearity is used in each layer. Stochastic Gradient Descent (SGD) optimizer was used to train the network with an initial learning rate of 0.01 and a momentum of 0.9. Mean squared error objective function was used as objective. Learning rate was reduced by a factor of 2 when the loss did not improve for two consecutive epochs. Mini batch size was set to 128 frames and in each epoch the network was trained on 80,000 mini batches. In each epoch, all the mini batches were randomly sampled from the training data. Figure 1 shows the pipeline we used to train the BWE. Figure 2 shows the spectrogram output of the BWE network along with the ground truth for a randomly picked utterance (The network is not trained on this utterance)

3.2. Upsampling with low-pass filter interpolator

The baseline upsampling is the traditional method used in signal processing. Zeros are interpolated between each wav sample. A low pass filter eliminates the aliases created in the higher band, i.e., interpolates the unknown signal values. We used the

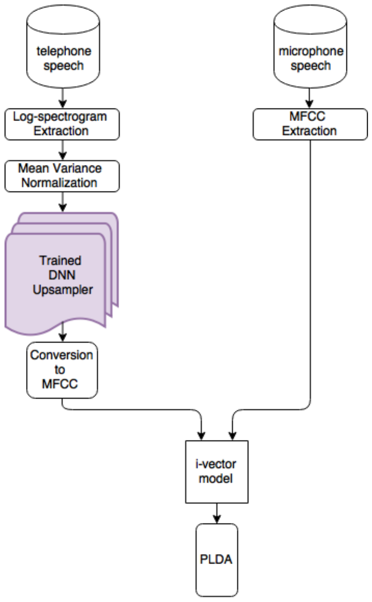

Figure 3: *The i-vector pipeline*

implementation in SoX [1], which used a filter with 125 dB of attenuation for the rejected band.

4. Experimental Setup

In this section, we introduce the datasets we used to train and evaluate our models. We then explain the experimental setup of i-vector and x-vector systems.

4.1. Datasets Description

A small portion of WB dataset that we used in this work is collected from Mixer6 and NIST SRE08 corpus. These data comprises of microphone recordings of telephone calls. The same speaker is recorded on several microphones. Hence, there exists a lot of redundancy. There is also speaker overlap between the telephone corpus (NB) and the microphone data (WB). Major portion of the WB corpus comes from recently introduced *VoxCeleb* dataset [13] which contains speech from celebrity speakers. WB consists of 30974 utterances collected from 1871 speakers.

NB data that we used for this work comprises of SwitchBoard 2 Phases 1, 2 and 3, SwitchBoard Cellular and NIST 2004 - 2010 including Mixer 6. For NIST SRE08 and MX6 there exists some speaker overlap between the NB and WB datasets. NB dataset used in our work consists of 86594 utterances collected from 7001 speakers.

Speakers In The Wild (SITW) [8] is used for evaluating our models. SITW consists of variable length utterances from 6-240 seconds. Speech from this corpus consists of video audio from native English speakers, with naturally occurring noises, reverberation and device variability. The sampling frequency of this dataset is 16 kHz (WB). We tested our models on the *core* and *assist* conditions of SITW.

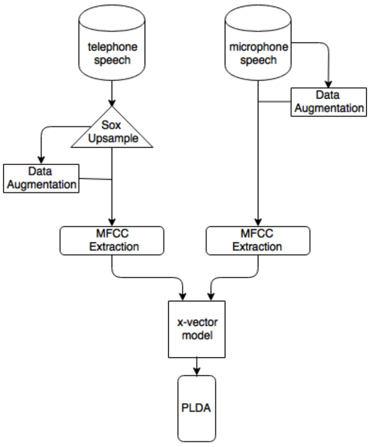

Figure 4: *The x-vector pipeline*

4.2. i-vector system trained on BW extended data

We train an i-vector system on wideband data obtained by extending the BW of NB data using the BWE network and original wideband data. Since we trained the BWE network on LPS features, we compute LPS features for the NB data and forward pass them through the BWE. The output of the network is LPS features in WB domain. These features are converted to MFCCs with 24 cepstral coefficients and 30 filter banks. We combined these MFCCs with the ones obtained from original WB data. We used this combined data to train the i-vector system. i-vectors obtained for the train and evaluation data are used to train and score the back-end PLDA model. We did not use any data augmentation for training the i-vector extractor and PLDA back-end. The i-vector system trained on BW extended data and original WB data is explained pictorially in Figure 3.

We compare the results obtained using the above model with two i-vector baselines: NB and WB baselines. NB and WB baselines are trained on 23 and 24 dimensional MFCC features respectively. NB system has 23 mel-filter banks and WB data has 30 mel-filterbanks. No data augmentation is used for training. NB baseline is trained with the entire NB data mentioned in 4.1 along with downsampled WB data. WB baseline is trained only on the available WB data. To evaluate the NB baseline, SITW corpus is downsampled to 8 kHz to match the sampling frequency of the training set.

4.3. Mixed BW x-vector system

To train the mixed BW x-vector system, we upsampled the NB data using the technique described in Section 3.2. Note that this technique is not going to fill in any additional frequency information in the upper half of the spectrum but preserves the information present in the lower half. This upsampled data is combined with original WB data. We used data augmentation on the training data to increase the amount and diversity of the data. The protocol we followed to augment the data is similar to Section 3.3 in [10]. We extract 24 dimensional MFCCs with 30 mel-filterbanks on the clean and augmented data. We then train a mixed BW x-vector system on these features. The training procedure for the mixed BW x-vector system is depicted pictorially in Figure 4.

We train a NB baseline system to compare the results obtained from the mixed BW x-vector system. The NB baseline system is trained on 23 dimensional MFCC features with 23

[1] http://sox.sourceforge.net

Table 1: *Results of i-vector system trained on WB data and DNN based BW extended data (no data augmentation is used to train the i-vector extractor or PLDA)*

	EVAL CORE			EVAL ASSIST		
i-vector Systems	EER	DCF 1E-3	DCF 1E-2	EER	DCF 1E-3	DCF 1E-2
NB Baseline	9.68	0.809	0.661	11.78	0.822	0.655
WB Baseline	6.30	0.652	0.466	8.99	0.658	0.526
WB-BWE	8.99	0.775	0.608	10.86	0.756	0.610
WB-BWE + PLDA trained on WB	6.29	0.639	0.484	8.99	0.660	0.529

mel filter banks. Entire NB data available is used for training the system along with downsampled WB data. To test the NB baseline system evaluation test set is downsampled to 8 kHz to match the sampling frequency of the training set. We used augmentation for the baseline system to be consistent with the mixed BW system.

5. Results

5.1. i-vector system trained on BW extended data

Table 1 present results for the i-vector system experiments in terms of EER and detection cost function (DCF) in two operating points. The first two rows shows the NB and WB baselines respectively. The WB baseline, even though is trained on much less data compared to NB system, performed better. This can be due to the fact that SITW domain is similar to VoxCeleb. Both are WB speech data collected from Internet videos. This means that having in domain data can be more important than having more training speakers. The i-vector system trained with BW extended data (from NB) and original WB data (we label this experiment WB-BWE) performed better compared to the NB baseline showing significant improvements (9.6% reduction in EER for the core condition). However, the system's performance did not improve compared to the WB baseline (row 2). This means that including the BW expanded data in training along with the WB data did not give any additional advantage when tested on SITW. We modified this experiment by training the PLDA backend on i-vectors of the WB data only (the i-vectors obtained from the BW expanded features are not included in PLDA training). In this last case the only thing trained on BW extended data and WB data are the UBM and i-vector extractors. The result we obtained from this experiment were very similar to the WB baseline system. To explain this result, we hypothesize that for a given speaker, there is still significant mismatch between his/her i-vectors coming from BWE speech and i-vectors coming from real WB data. When pooling BWE and WB i-vectors to train PLDA, we obtain a within-class covariance larger than that of WB data only. Furthermore, in SITW enrollment and test data are true WB speech. That means that training PLDA on pooled BWE and WB data overestimates the within-class covariance we should use for this particular application.

5.2. Mixed BW x-vector system

The NB baseline results for x-vector system are given in row 1 of Table 2. We used data augmentation in this experiments. Since we did not use data augmentation for the i-vector experiments, we cannot compare this baseline to the NB baseline in i-vector experiments. The mixed BW xvector system trained on upsampled data and original WB data is given in row 2 of Table 2. The mixed BW systems perform better for both the evaluation conditions compared to the baseline. This is because the x-vector model, being a DNN, is able to optimize its neurons to respond to the lower half of the spectrum for NB data and to use the entire spectrum (all cepstral coefficients) for the WB data. The main advantage with this approach is that it eliminates the need for a separate BWE network. We did not observe huge gap in performance difference between NB and WB baselines for x-vector system as we did for i-vector. This is mainly because of the limited amount of WB data available to train the WB baseline x-vector model compared to NB baseline. As mentioned in 2.2, x-vector models can outperform i-vector models only when trained on large amounts of data.

Table 2: *Results of mixed BW x-vector model*

	EVAL CORE			EVAL ASSIST		
x-vector Systems	EER	DCF 1E-3	DCF 1E-2	EER	DCF 1E-3	DCF 1E-2
NB Baseline MFCC	4.54	0.623	0.425	6.74	0.650	0.468
Mixed BW system	4.40	0.570	0.376	6.57	0.612	0.435

6. Conclusions

In this work, we presented two independent case studies: (1) training the state-of-the-art i-vector speaker recognition system on BW extended speech and (2) training the recently introduced x-vector system on mixed BW speech. We observed that, for SITW dataset, pooling more data (by extending the BW of NB data) to train the i-vector extractor and PLDA backend did not give improvements over the WB baseline system. Training the i-vector extractor and backend model using fewer amounts of in domain data actually helps more than pooling more data by extending the BW. For mixed BW x-vector system, where the WB data is combined with upsampled NB data, we observed improvement in performance over the baseline system. The advantage with this model is that it eliminates the requirement of a separate BWE network to predict the WB speech. In future, we would like to extend this work by training Teacher-Student model [14], where the student model trained on NB data would mimic the performance of the Teacher model trained on WB data, thus adapting the model to the NB data. That way we achieve task of classification and adaptation with a single model and eliminate the requirement of a separate BWE network.

7. References

[1] C. Avendano, H. Hermansky, and E. A. Wan, "Beyond nyquist: Towards the recovery of broad-bandwidth speech from narrow-bandwidth speech," in *Fourth European Conference on Speech Communication and Technology*, 1995.

[2] K. Li, Z. Huang, Y. Xu, and C.-H. Lee, "Dnn-based speech bandwidth expansion and its application to adding high-frequency missing features for automatic speech recognition of narrow-band speech," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[3] X. Zhuang, A. Ghoshal, A.-V. Rosti, M. Paulik, and D. Liu,

"Improving dnn bluetooth narrowband acoustic models by cross-bandwidth and cross-lingual initialization," *Proc. Interspeech 2017*, pp. 2148–2152, 2017.

[4] J. Li, D. Yu, J.-T. Huang, and Y. Gong, "Improving wideband speech recognition using mixed-bandwidth training data in cd-dnn-hmm," in *Spoken Language Technology Workshop (SLT), 2012 IEEE*. IEEE, 2012, pp. 131–136.

[5] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[6] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *Proceedings of the 2016 IEEE Spoken Language Technology Workshop (SLT)*. San Diego, CA, USA: IEEE, dec 2016, pp. 165–170.

[7] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep Neural Network Embeddings for Text-Independent Speaker Verification," in *Proceedings of the 18th Annual Conference of the International Speech Communication Association, INTERSPEECH 2017*. Stockholm, Sweden: ISCA, aug 2017, pp. 999–1003.

[8] M. McLaren, L. Ferrer, D. Castan, and A. Lawson, "The speakers in the wild (sitw) speaker recognition database." in *Interspeech*, 2016, pp. 818–822.

[9] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[10] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-Vectors : Robust DNN Embeddings for Speaker Recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018*. Alberta, Canada: IEEE, apr 2018.

[11] N. Brummer and E. De Villiers, "The Speaker Partitioning Problem," in *Proceedings of Odyssey 2010 - The Speaker and Language Recognition Workshop*. Brno, Czech Republic: ISCA, jul 2010, pp. 194–201.

[12] N. Brummer and A. Strasheim, "AGNITIO's Speaker Recognition System for EVALITA 2009," in *Proceedings of the 11th Conference of the Italian Association for Artificial Intelligence*, Reggio Emilia, Italy, dec 2009.

[13] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[14] J. Li, M. L. Seltzer, X. Wang, R. Zhao, and Y. Gong, "Large-scale domain adaptation via teacher-student learning," *arXiv preprint arXiv:1708.05466*, 2017.