IberSPEECH 2018
21-23 November 2018, Barcelona, Spain

The SRI International STAR-LAB System Description for IberSPEECH-RTVE 2018 Speaker Diarization Challenge

Diego Castan, Mitchell McLaren, Mahesh Kumar Nandwana

Speech Technology and Research Laboratory, SRI International, California, USA

{diego.castan, mitchell.mclaren, maheshkumar.nandwana}@sri.com

Abstract

This document describes the submissions of STAR-LAB (the Speech Technology and Research Laboratory at SRI International) to the open-set condition of the IberSPEECH-RTVE 2018 Speaker Diarization Challenge. The core components of the submissions included noise-robust speech activity detection, speaker embeddings for initializing diarization with domain adaptation, and Variational Bayes (VB) diarization using a DNN bottleneck i-vector subspaces.

1. Introduction

SRI International has long focused on the task of speaker recognition, but has only recently branched into the field of speaker diarization. For the STAR-LAB submissions, we leveraged the embeddings and diarization systems that we recently developed for the NIST 2018 Speaker Recognition Evaluation [1]. In addition, we attempt to leverage recent work in speech activity detection and in speaker embeddings for speaker recognition [2, 3, 4], the well-known Variational Bayes (VB) approach to diarization [5], and the use of a DNN bottleneck based i-vector subspaces internal to the VB process. We describe the three systems submitted for the open-set condition based on a hybrid embeddings-VB approach with different parameters for the speech activity detection system.

2. System Training Data

System training data included 234,288 signals from 14,630 speakers. This data was compiled from NIST SRE 2004-2008, NIST SRE 2012, Mixer6, Voxceleb1, and Voxceleb2 (train set) data. Voxceleb1 data had 60 speakers removed that overlapped with Speakers in the Wild (SITW), according to the publicly available list[1].

Augmentation of data was applied using four categories of degradations as in [4], including music, and noise at 5 dB signal-to-noise ratio, compression, and low levels of reverb. We used 412 noises of at least 15 seconds in length compiled from both freesound.org and the MUSAN corpus. Music degradations were sourced from 645 files from MUSAN, and 99 instrumental pieces purchased from Amazon music. For reverberation, examples were collected from 47 real impulse responses available on echothief.com, and 400 low-level reverb signals sourced from MUSAN. The random selection of reverb signals gave almost 10x weight to the echothief.com examples in order to balance data sources. Compression was applied using 32 different codec-bitrate combinations with open source tools such as FFmpeg, codec2, Speex, GSM, and opus trans-coding packages. In addition to these augmentations, we down-sampled any 16k or higher data (74,447 files) to 8k before up-sampling to 16k, which we have found to allow the embeddings DNN to generalize across the 8-16 kHz bandwidth range and better accommodate processing of telephone signals.

We augmented the raw speaker embeddings training data (counting the 16k re-sampled to 8k as raw data) to produce 2 copies per file per degradation type (random selection of specific degradation) such that the data available for training was 9-fold the original amount. In total, this was 2,778,615 files for training the speaker embedding DNNs. For PLDA training used for clustering embeddings [6], the same degraded data (excluding the 8k simulated data) was subsampled by a random factor of 6 in order to make PLDA training data manageable and resulted in 343,535 files from 11,461 speakers. Finally, the databases used to train the UBM and the total variability subspaces were Fisher, Switchboard and AMI PRISM (NIST SRE04-08).

3. The STAR-LAB System Submissions

We start with a general overview of the submissions prior to breaking down into details of each module used in the system. Figure 1 shows a block diagram of the different parts of the system. All our submissions use embeddings clustering as seed to VB diarization. The differences between the primary and the contrastive submissions are the thresholds applied for the SAD decisions as detailed in Section 3.2

3.1. Acoustic Features and Bottleneck i-vector extraction

To train the DNN we used Power-Normalized Cepstral Coefficients (PNCC) [7] with 30 dimensions extracted from a bandwidth of 100-7600 Hz using 40 filters. Root compression of 1/15 was applied. All audio was sampled (up or down sampled where needed) to 16 kHz prior to processing. The features were first processed with mean and variance normalization over a sliding window of 3 seconds prior to having SAD applied.

We used Mel Frequency Cepstral Coefficients for speech activity detection and as input to a DNN to extract 80-dimensional bottleneck (BN) features. This BN DNN was trained for the task of automatic speech recognition. The DNN was trained to predict 1933 English tied tri-phone states (senones). MFCCs were used for input to the DNN after transforming them with a pcaDCT transform [8] trained on Fisher data, and restricting the output dimension to 90. The DNN consisted of 5 hidden layers of 600 nodes, except the last hidden layer which was 80 nodes and formed the bottleneck layer from with activations were extracted as features.

BN features were used to train an i-vector extractor [9] consisting of a 2048-component universal background model (UBM) with diagonal covariance and a subspace of rank 400.

[1] http://www.openslr.org/resources/49/voxceleb1_sitw_overlap.txt

10.21437/IberSPEECH.2018-42

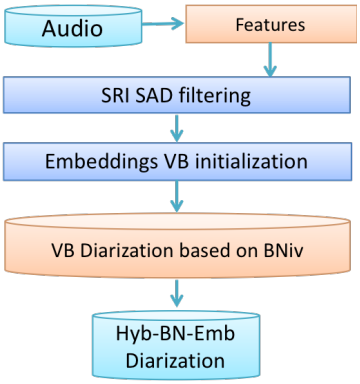

Figure 1: *Flow diagram of components used in the STAR-LAB team submissions to the IberSPEECH-RTVE 2018 Speaker Diarization Challenge.*

3.2. Speech Activity Detection

We used a DNN-based Speech Activity Detection (SAD) model leveraging short-term normalization. The SAD model was trained on clean telephone and microphone data from a random selection of files from the provided Mixer datasets (2004-2008), Fisher, Switchboard, Mixer6, SRE'18 unlabeled and SRE'16 unlabeled data. A 5 minute DTMF tone (acquired from YouTube), and a selection of noise and music samples with and without speech added were added to the pool of data. In all, 11,668 files were used to train the SAD model.

The system uses 19-dimensional MFCC features, which excluded C0 and used 24 filters over a bandwidth of 200-3300 Hz. These features were mean and variance normalized using a sliding window of 3 seconds, and concatenated over a window of 31 frames. The resulting 620-dimensional feature vector formed the input to a DNN which consisted of two hidden layers of sizes 500 and 100. The output layer of the DNN consisted of two nodes trained to predict the posteriors for the speech and non-speech classes. These posteriors are converted into likelihood ratios using Bayes rule (assuming a prior of 0.5), and thresholded at a value of -1.5, -2.0 and -3.0 for the primary and both contrastive systems, respectively. A padding of 0.5 seconds was applied over the final segmentation to smooth the transitions between speech/non-speech.

We applied cross-talk removal on all interview data from the NIST SRE corpora to suppress the interviewer speech that bled through to the target speaker channel. This was especially important for distant microphone channels in which each speaker had similar energy. Cross-talk removal involved using the SAD Log-Likelihood Ratios (LLRs) from the target microphone as well as the close-talking interviewer microphone, and removing any detected speech from the target channel that was detected in the interviewer channel with more than 3.5 in LLR value of the target channel.

For all augmented system training data, the SAD alignments from the raw audio were used rather than running SAD on the degraded signals directly, as done in [4].

3.3. Embeddings VB Initialization

Recent work in [2, 3] has shown significant advances in the related field of *speaker recognition* by replacing the well-known i-vector extraction process with speaker embeddings extracted from a DNN trained to directly discriminate speakers. We decided to apply our findings on what makes a good speaker embeddings extractor [4] to the task of speaker clustering.

We used a multi-bandwidth speaker embedding DNN in our submission. Speaker embedding DNNs were trained following the protocol of [4]. Specifically, Kaldi was used to generate examples for training the DNN with a duration ranging between 2.0 and 3.5 seconds of speech. DNNs were trained using Tensorflow over 6 epochs using a mini batch size of 128 examples, and dropout probability linearly increasing to 10% then back to 0% in the final 2 epochs. The embeddings network starts with five frame-level hidden layers, all using rectified linear unit (ReLU) activation and batch normalization. The first three layers incrementally add time context with stacking of [-2,-1,0,1,2], [-2,0,2], and [-3,0,3] instances of the input frame. A statistics pooling layer then stacks the mean and standard deviation of the frames per audio segment, resulting in a 3000 dimensional segment-level representation. The final two hidden layers of 512 nodes operate at the segment-level and use ReLU activation and batch normalization prior to the output layer, which targets speaker labels for each audio segment using log softmax as the output. The embeddings are extracted from the first segment-level hidden layer of 512 nodes. This system used PLDA classification for clustering after applying an LDA dimensionality reduction. We applied length and mean normalization to embeddings prior to use in PLDA. As a simple method of domain adaptation, we mean normalized the chunked embeddings from an audio file using the mean of all chunks.

The embeddings VB initialization process was performed as follows. The audio was first segmented into 1.5 second segments with 0.2 second shift. Following a similar strategy to VB diarization, we initialized a speaker cluster posterior matrix, q, to for 13 speakers. The number of speakers was selected from previous experiments over the development data. We calculated for each speaker cluster, a weighted-average embedding based on q and the 1.5s embeddings segments. These per-cluster embeddings were compared using PLDA against each individual embedding segment. We scaled the likelihood ratios (LLRs) that resulted from PLDA by 0.05 and performed Viterbi decoding of the LLRs to result in a new q and speaker priors. This process was iterated 10 times before using the result q and speaker priors in the subsequent VB diarization based on BN+MFCC features.

3.4. Variational Bayes diarization

Our diarization approach was based on the work of [10]. This approach uses an i-vector subspace to produce a frame-level diarization output. Our i-vector subspace was trained using concatenated BN and MFCC features [11], resulting in a feature with 140 dimensions. With VB diarization, we have used a left-to-right HMM structure of three states per speaker in order to smooth the transitions between speakers that was proposed in [12].

The initialization of the VB diarization approach is done with the speaker posteriors estimated from the speaker embeddings initialization. We performed a maximum of 20 iterations of VB diarization.

4. Results

This section compares the development performance of the STAR-LAB submissions.

Table 1 shows the system performances for the primary and the two contrastive systems in the development set.

Table 1: *Development results for each system submission*

System Name	Miss Sp	FA Sp	SpkErr	DER
Primary	2.1%	1.9%	13.4%	17.38%
Contrastive 1	2.0%	2.3%	13.4%	17.81%
Contrastive 2	2.0%	2.9%	12.2%	17.08%

Table 2: *Computational requirements of STAR-LAB submissions from based on RT factor (higher than 1.0 is slower than real time) and maximum resident memory needed to diarize 10-minutes of development file millennium-20170522.*

System	x RT	Max. Res. RAM
Primary	1.19	3.6G
Contrastive 1	1.52	3.6G
Contrastive 2	1.28	3.6G

5. Computation

We benchmarked the computational requirements of the STAR-LAB system on a single core. The machine was an Intel Xeon E5630 Processor operating at 2.53GHz. The approximate processing speed and resource requirements are listed in Table 2.

These calculations are based on total CPU time divided by the total duration of the audio.

6. Acknowledgments

We'd like to thank Lukas Burget and the BUT team for their python implementation of Variational Bayes diarization which was leveraged in this work [10].

7. References

[1] Mitchell McLaren, Luciana Ferrer, Diego Castan, Mahesh Nandwana, and Ruchir Travadi, "The sri-con-usc nist 2018 sre system description," in *NIST 2018 Speaker Recognition Evaluation*, 2018.

[2] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and Sa Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 165–170.

[3] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. Interspeech*, 2017, pp. 999–1003.

[4] M. McLaren, D. Castan, M. K. Nandwana, L. Ferrer, and E. Yilmaz, "How to train your speaker embeddings extractor," in *Speaker Odyssey*, 2018, pp. 327–334.

[5] Patrick. Kenny, "Bayesian analysis of speaker diarization with eigenvoice priors," in *Tech. Rep. CRIM*, 2008.

[6] S.J.D. Prince and J.H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Proc. IEEE International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.

[7] C. Kim and R. Stern, "Power-normalized cepstral coefficients (PNCC) for robust speech recognition," in *Proc. ICASSP*, 2012, pp. 4101–4104.

[8] Mitchell McLaren and Yun Lei, "Improved speaker recognition using dct coefficients as features," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4430–4434.

[9] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. on Speech and Audio Processing*, vol. 19, pp. 788–798, 2011.

[10] *VB diarization with eigenvoice and HMM priors*, 2013, `http://speech.fit.vutbr.cz/software/vb-diarization-eigenvoice-and-hmm-priors`.

[11] Mitchell McLaren, Yun Lei, and Luciana Ferrer, "Advances in deep neural network approaches to speaker recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4814–4818.

[12] M. Diez, L. Burget, and P. Matejka, "Speaker diarization based on bayesian hmm with eigenvoice priors," in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop , 2018*, 2018, pp. 147–154.