BUT system description for DIHARD Speech Diarization Challenge 2018

Mireia Diez, Federico Landini, Lukáš Burget, Johan Rohdin, Anna Silnova, Kateřina Žmolíková, Ondřej Novotný, Karel Veselý, Ondřej Glembek, Oldřich Plchot, Ladislav Mošner, Pavel Matějka

Brno University of Technology, Speech@FIT, Czechia

{mireia,landini}@fit.vutbr.cz

Abstract

This paper describes the systems developed by the BUT team for the first DIHARD speech diarization challenge. All our systems are based on our Bayesian Hidden Markov Model with eigenvoice priors system.

Index Terms: Speaker Diarization, Variational Bayes, HMM, i-vector, x-vector, Overlapped speech, DIHARD

1. Data resources

Two main training sets were used for training the systems submitted to the evaluation. The first set consists of (8 kHz mostly telephone) data from NIST SRE 2004 - 2008 datasets, which amounts to around 1400h of speech. We will denote this set as (*Tel*). The second set consists on the data from DIHARD development set[1, 2], excluding utterances coming from VAST, as we found the labeling to be too noisy and experiments in the development set proved that removing it from the training set enhanced performance. We will denote this set with *dev* acronym.

For some systems we made use of the DIHARD evaluation data set in an unsupervised way, details on the specific use are depicted in the system descriptions.

For the systems initialized with x-vectors (see later system descriptions, i.e. 3.2), the x-vector extractor was trained on data from NIST SRE 2004-2008, Fisher English and Switchboard.

2. Description of algorithm

Our main approach to the diarization problem is based on a Bayesian Hidden Markov Model (HMM) with eigenvoice priors [3]. This model assumes that the sequence of speech features representing a conversation is generated from a HMM, where each state represents one speaker and the transitions between the states correspond to speaker turns. An ergodic HMM is used, where transitions from any state to any state are possible. However, the transition probabilities are set in a way that discourages too frequent transitions between states in order to reflect speaker turns durations of a natural conversation. The HMM state (or speaker) specific distributions are modeled by Gaussian Mixture Models (GMMs) with an informative eigenvoice prior imposed on the GMM parameters. Such prior, which is essentially the same as in i-vectors [4] or Joint Factor Analysis (JFA) [5] models, allows us to robustly estimate speaker distributions, which facilitates discrimination between the speaker voices in the input recording. The proposed Bayesian model offers a elegant approach to SD as a straightforward and efficient Variational Bayes (VB) inference in a single probabilistic model addresses the complete SD problem: For each input conversation, we construct a HMM with preferably more states than what is the assumed number of speakers in the conversation and we start with some initial (possibly random) assignment of frames to HMM states. Then, each VB training iteration refines the HMM state specific distributions and recalculates the soft (probabilistic) assignment of frames to the HMM states. During the VB training, the complexity control inherent in the Bayesian learning automatically drops the redundant HMM states (i.e. learns zero transition probabilities into such states) and decides on the number of speakers in the conversation. The final assignment of frames to the "surviving" HMM states gives the solution to the diarization problem. An open source code for the algorithm is provided in [6] and more details on the algorithm can be found in [3]. In the rest of this paper, we will refer to this approach as VB diarization.

Therefore, the VB diarization, as a single probabilistic model, integrates the speaker estimation and clustering method without the need for further resegmentation steps.

The VB diarization system can be initialized setting an upper bound on the number of speakers for the input utterance and using a random assignment of frames to speakers. Also, it can be initialized using a labeling attained with an external diarization algorithm. We will include the description of the initialization used in each of our submission.

3. System Descriptions

3.1. BASELINE (single submission)

This is our diarization system as described in [3]

Signal processing The Voice Activity Detection (VAD) using to process the *Tel* system training is based on the BUT Czech phoneme recognizer [7], dropping all frames that are labeled as silence or noise. The recognizer was trained on the Czech CTS data, but we have added noise with varying Signal to Noise Ratio (SNR) to 30% of the database.

Acoustic features HTK-based Mel Frequency Cepstral Coefficients (MFCC), with 19 coefficients plus Energy extracted from 8kHz audios. The analysis window is 20ms with a shift of 12ms.

System parameters For the speaker subspace of VB algorithm We used GMM with 1024 Gaussians and i-vector extractor 400 dimensions trained in gender independent fashion using the *Tel* training set defined in Section 1. The parameters used

The work was supported by Czech Ministry of Interior project No. VI20152020025 "DRAPAK", European Unions Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 748097, the Marie Sklodowska-Curie cofinanced by the South Moravian Region under grant agreement No. 665860, Google Faculty Research Award program, Czech Science Foundation under project No. GJ17-23870Y, Technology Agency of the Czech Republic project No. TJ01000208 "NOSICI", and by Czech Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project "IT4Innovations excellence in science - LQ1602".

for the VB algorithm were: statscale 0.1, loop probability 0.9, min duration 1 and downsampling 25.

VB diarization initialization We initialize the VB diarization using the output a diarization system which works as follows [8]: each utterance is segmented into 2 second speech segments, overlapped by 0.5 seconds. A 1024 dimensional Gaussian Mixture Model (GMM) is trained as a Universal Background Model (UBM). 64 dimensional i-vectors [4] are extracted from each segment and projected by means of Principal Component Analysis to 3 dimensions [9]. The UBM and the GMM are trained using also the *Tel* set. The generated i-vectors are then clustered using Agglomerative Hierarchical Clustering (AHC) using calibrated Probabilistic Linear Discriminant Analysis (PLDA) similarity scores. [10, 11].

Performance track1 35.85 %DER, 8.33 MI

3.2. dev-s2

Signal processing We used the Weighted Prediction Error (WPE) [12] method to remove late reverberation from the data. We estimated a dereverberation filter on Short Time Fourier Transform (STFT) spectrum for every 100 second block of an utterance. To compute the STFT, we used $32\,\mathrm{ms}$ window with $8\,\mathrm{ms}$ shift. We set the filter length and prediction delay to 20 and 3 respectively for $16\,\mathrm{kHz}$, and 10 and 2 respectively for $8\,\mathrm{kHz}$ data. The number of iterations was set to 3.

Acoustic features Mel Frequency Cepstral Coefficients (MFCC), with 19 coefficients plus Energy plus first order derivatives extracted from 16kHz audios. The analysis window is 20ms with a shift of 12ms.

System parameters For the speaker subspace of VB algorithm, we used GMM with 512 Gausians and i-vector extractor 200 dimensions trained in gender independent fashion using the *dev* training set as defined in Section 1. As unsupervised usage of the evaluation data was allowed, the eval set was included in the UBM training set. The parameters used for the VB algorithm were: statscale 0.1, loop probability 0.9, min duration 1 and downsampling 25.

VB diarization initialization The approach used is based also on the PLDA AHC described in Section 3.1, but uses speaker embeddings instead of i-vectors.

To extract speaker embeddings, referred to as *x-vectors*, we employed the architecture described in [13] (embedding A). We trained the NN with the corresponding Kaldi recipe [14] except that we used the data described in Section 1 in order to comply with the rules of the DIHARD challenge. Also, we found that not using augmented data in the PLDA training would benefit the diarization task, so the training set was not augmented. Finally, we reduced the minimum number of utterances a speaker needs to have in order to be included in the training set from 8 to 6.

The x-vectors were projected to 150 dimensions by LDA with no length normalization and mean subtraction was applied with mean computed over the PLDA training set subtracted.

Overlap handling Since the current diarization system outputs one speaker label per frame, a post processing of the output was carried out. An overlapped speech detector was trained using three corpora in which overlaps are annotated: AMI[15] - 98h (1st microphone from 1st microphone array), Callhome - 17h (multi-lingual subset of train-sets), SRE08 test set - 186h (LDC2011S08). The training data were selected to contain a rich mixture of languages and domains. The model is a modified version of our VAD from Section 3.2. The difference is that the NN has 3 outputs: 'speech', 'non-speech' and 'overlapped speech'. The per-frame score is the logit of posterior of 'overlapped speech' NN output. The rest is the same as for the VAD described in 3.2: fbank+pitch feature front-end, 2 hidden-layer NN topology and averaging of logit-scores over a window of 31 frames.

The detector was applied using two thresholds: one aggressive and one precise. The aggressive threshold was used to filter out *any overlapped speech* in order to feed the first pass of the VB algorithm only with reliable speech frames. Then, the precise threshold was used to detect speech segments that are overlapped speech with high probability. In a second pass of the VB algorithm the speaker models were kept fixed and those frames filtered out by the aggressive threshold but not by the precise one were assigned to speaker models. We saw that this approach helped the most for the noisiest domains on dev data. Then, only the frames spotted by the precise detector were given two speaker labels in order to reduce the false alarm rate. The frames were tagged according to the following rules:

- If the neighboring frames are assigned to different speakers, the overlap segment is assigned to those speakers.
- If only one of the neighboring segments is assigned to a speaker (the other to silence), or both were assigned to the same speaker, the overlap segment is assigned to that speaker and to the next most likely speaker according to the diarization model output.
- If both neighboring segments were silence segments, the overlap segment was assigned to the two most likely speakers according to the diarization model output.

Source identification We built a subsystem that automatically classify evaluation recordings according to the domains given in the dev set. We found that the only *domain dependent* strategy that generalized to the evaluation data was to detect LibriVox recordings, which always contain one speaker and label them with a single speaker.

To classify domains, we trained a Gaussian Linear Classifier (i.e. Gaussian distributed classes with shared covariance matrix) on 64 dimensional i-vectors extracted from the whole recordings. The i-vector extractor was trained on the *Tel* dataset with addition of the LibriSpeech dataset [16]. The classifier was trained on the development data and 150 randomly chosen files from previously released Librivox data [17].

Voice Activity Detection (VAD) For the Track 2 of the challenge, in which no golden segmentation labels were provided, a VAD system was used in order to discard silence and feed the rest of the system only with speech segments. Our VAD is based on a neural network (NN) trained for binary, speech/non-speech, classification of speech frames. The 288-dimensional NN input is derived from 31 frames of 15 log Mel filter-bank outputs and 3 pitch features. The NN with 2 hidden layers of 400 sigmoid neurons was trained on the Fisher English with labels provided from Automatic Speech Recognition alignment. Per-frame logit posterior probabilities of speech were smoothed by averaging over consecutive 31 frames and thresholded to at the value of 0 to give the final hard per frame speech/ non-speech decision. See [18] for more detailed de-

scription of the VAD system.

Performance track1 25.39 %DER, 8.43 MI

Performance track2 35.51 %DER, 8.07 MI

We initialize the VB diarization using the output a diarization system which works as follows [8]: each utterance is segmented into 2 second speech segments, overlapped by 0.5 seconds. A 1024 dimensional Gaussian Mixture Model (GMM) is trained as a Universal Background Model (UBM). 64 dimensional i-vectors [4] are extracted from each segment and projected by means of Principal Component Analysis to 3 dimensions [9]. The UBM and the GMM are trained using also the *Tel* set. The generated i-vectors are then clustered using Agglomerative Hierarchical Clustering (AHC) using calibrated Probabilistic Linear Discriminant Analysis (PLDA) similarity scores. [10, 11].

3.3. dev-s4

Signal processing Dereverberation was applied as defined in section 3.2.

Acoustic features HTK-based Mel Frequency Cepstral Coefficients (MFCC), with 19 coefficients plus Energy extracted from 8kHz audios. The analysis window is 20ms with a shift of 12ms.

System parameters For the speaker subspace of VB algorithm We used GMM with 1024 Gaussians and i-vector extractor 400 dimensions trained in gender independent fashion using the *dev* training set defined in Section 1. The parameters used for the VB algorithm were: statscale 0.1, loop probability 0.9, min duration 1 and downsampling 25.

VB diarization initialization In this case we did not use an external clustering algorithm to initialize the VB algorithm, instead we proceeded as follows: The golden segmentation was applied and silence parts were removed from the signal. For the remaining speech parts, a 5 second segmentation was applied, with no overlap and a different speaker was assigned to every segment.

Performance track1 29.94%DER, 8.39 MI

3.4. dev-s5

Similar to system dev-s2, with two main differences:

VB diarization initialization The initialization of the VB algorithm was as defined for the BASELINE system, using i-vectors instead of x-vectors.

Usage of evaluation data We re-trained the eigenvoice subspace for the VB diarization on the pooled development and evaluation data. However, this procedure required speaker labels for the evaluation data, which were not available. We obtained such labels in an unsupervised way as follows: the evaluation data was labeled using 5 diverse diarization systems developed for this challenge using different features, initializations, etc. The different sets of features in the 5 systems conveyed 19MFCC+E features, but extracted from/using: 8kHz denoised data, 8kHz dereverberated data, 16kHz data plus deltas, 8kHz data using global cepstral mean (CM) subtraction and 8kHz data using 3s floating window cepstral mean and variance normalization (CMVN). For each evaluation recording, the *fused* cluster labels were given by concatenated frame speaker labels from all systems, that is, with the resulting labels, frames would belong to the same cluster only if all the systems agreed on having only one speaker in the cluster. We selected the largest cluster as training data representing one speaker, we discarded all frames/clusters that were believed to belong to the same speaker by any of the system and we continued with the next largest cluster.

Performance track1 26.46 %DER, 8.40 MI

3.5. dev-s6

Similar to dev-s5, but using the x-vectors for initialization.

Performance track1 25.07 %DER, 8.43 MI

Submission to second track This system was submitted to the leaderboard for the second track (which achieved 35.35 %DER and 8.09 MI). After realizing that we were actually making an indirect use of the golden segmentation, as we fuse 5 systems that use the golden segmentation to extract the evaluation set labels, we requested to remove the system from the official set of results.

4. Hardware requirements

The infrastructure used to run the expriments is a CPU, Intel(R) Xeon(R) CPU 5675 @ 3.07GHz, with a total memory of 37GB. The execution time of i-vector extraction process in a single thread is of 18 times faster than real time (FRT) (computed only on detected speech, would be 41FRT computed for whole recordings including silence) for the MFCC system using 3GB of memory respectively. PLDA Enrollment and scoring is negligible with respect to the i-vector extraction time.

Training the neural network for x-vector extraction took approximately 30h using up to 8 Nvidia GTI 1080 TI GPUs per epoch (typically 2-4). Extraction of x-vectors for PLDA training and scoring was done with the Tensorflow library. Extracting x-vectors from 2s chunks with 0.5s overlap plus scoring all segments from the same recording against each other (with two different normalizations) took 36min for the 164 recordings in the development set using a single CPU thread. The total duration of these recordings after VAD is 14.4h which means the processing is approximately 24 FTR. Estimation of LDA and PLDA training with x-vectors required around 80GB of memory and took around 15 minutes due to the large training set (all 3s chunks from the training set with no overlap).

In our experiments, the processing time of the VB diarization algorithm for 10 minute files (considering AHC from the PLDA scores and VB diarization with overlap handling from pre-extracted features), ranged between 17s and 50s (real time). The processing time difference is because the VB algorithm converging time is dependent on the number of speaker models it is initialized with.

5. References

[1] N. Ryant and et al., "DIHARD Corpus. Linguistic Data Consortium." 2018.

[2] E. Bergelson, "Bergelson Seedlings HomeBank Corpus," 2016.

[3] M. Diez, L. Burget, and P. Matejka, "Speaker diarization based on bayesian hmm with eigenvoice priors," in *Proceedings of Odyssey 2018, The speaker and Language Recognition Workshop*.

[4] 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, May 2011.

[5] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Transactions Audio Speech Language Processing*, vol. 15, no. 4, pp. 1435–1447, 2007.

[6] L. Burget, "VB Diarization with Eigenvoice and HMM Priors," http://speech.fit.vutbr.cz/software/vb-diarization-eigenvoice-and-hmm-priors, 2013, [Online; January-2017].

[7] P. Matějka, L. Burget, P. Schwarz, and J. Černocký, "Brno University of Technology System for NIST 2005 Language Recognition Evaluation," in *Proceedings of Odyssey 2006*, San Juan, PR, 2006.

[8] G. Sell and D. Garcia-Romero, "Speaker diarization with plda i-vector scoring and unsupervised calibration," in *2014 IEEE Spoken Language Technology Workshop (SLT)*, Dec 2014, pp. 413–417.

[9] S. H. Shum, N. Dehak, R. Dehak, and J. R. Glass, "Unsupervised methods for speaker diarization: An integrated and iterative approach," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2015–2028, Oct 2013.

[10] P. Kenny, "Bayesian Speaker Verification with Heavy-Tailed Priors," Jun. 2010.

[11] P. Matějka, O. Glembek, F. Castaldo, J. Alam, O. Plchot, P. Kenny, L. Burget, and J. Černocký, "Full-covariance UBM and heavy-tailed PLDA in I-vector speaker verification," in *Proceedings of the 2011 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2011*. IEEE Signal Processing Society, 2011, pp. 4828–4831.

[12] T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, and B. H. Juang, "Speech dereverberation based on variance-normalized delayed linear prediction," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1717–1731, Sept 2010.

[13] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Interspeech 2017*, Aug 2017.

[14] Kaldi, "SRE16 v2," https://github.com/kaldi-asr/kaldi/tree/master/egs/sre16/v2, [Downloaded: 2017-12].

[15] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. Post, D. Reidsma, and P. Wellner, "The ami meeting corpus: A pre-announcement," in *Proceedings of the Second International Conference on Machine Learning for Multimodal Interaction*, ser. MLMI'05. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 28–39. [Online]. Available: http://dx.doi.org/10.1007/11677482_3

[16] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 5206–5210.

[17] "LIBRIVOX data," https://librivox.org/.

[18] P. Matějka and et.al., "BUT-PT system description for nist lre," in *Proceedings of NIST Language Recognition Workshop 2017*. National Institute of Standards and Technology, 2017, pp. 1–6.