ZCU-NTIS (ws-cr) Speaker Diarization System for the DIHARD 2018 Challenge

Zbyněk Zajíc[1], Marie Kunešová[1,2], Jan Zelinka[1,2], Marek Hrúz[1]

University of West Bohemia, Faculty of Applied Sciences
[1]NTIS - New Technologies for the Information Society and [2]Dept. of Cybernetics,
Univerzitní 8, 306 14 Plzeň, Czech Republic

Abstract. This paper describes the **"ws-cr"** diarization system - the baseline system developed by the team from the New Technologies for the Information Society (NTIS) research center of the University of West Bohemia (team "ZCU-NTIS"), for the First DIHARD Speech Diarization Challenge. Our system follows the currently-standard approach of segmentation, i-vector extraction and clustering. Additionally, we use an ANN-based domain classifier, which categorizes each conversation into one of 10 domains (the 9 corpora from the development set, plus "other"). This classification determines the specific system configuration (expected number of speakers, stopping criterion, etc.). This "ws-cr" version of the system uses fixed-length segmentation (with overlap) and no resegmentation, and achieves a DER of 27.43% and an MI of 8.33 bits on the evaluation set (using gold segmentation).

1 Differences from our other two systems

This document describes the entire "ws-cr" system, including parts that are identical to our other two systems. For convenience, here we list the parts which are different:

Unlike the "MH-crR" system, both this and the "ws-crR" system use segmentation with fixed segment length and overlaps between neighboring segments (section 3.3) and do not apply a weighing during the accumulation process for i-vector extraction (section 3.4).

Compared to our other two systems, this system also does not include a standard resegmentation. There is only a limited version which resolves conflicting labels in the segment overlaps (section 3.6). Adult-child classification of SEEDLingS data is also not included, and as the system was only used for Track 1, there is no speech activity detection (section 3.2).

Finally, the total execution times (section 4.5) are different for each system.

2 Data resources

2.1 i-Vector extraction for segment representation

Data for UBM:

- LibriSpeech (http://www.openslr.org/12/, training sets only),

- DIHARD Challenge Development Data (LDC2018E31),
- Czech Speecon database (ELRA-S0298, child voices only),
- UK English Speecon database (ELRA-S0215, child voices only),
- US English Speecon database (ELRA-S0233, child voices only),
- TIMIT Acoustic-Phonetic Continuous Speech Corpus (LDC93S1),
- CSR-I (WSJ0) Complete (LDC93S6A),
- CSR-II (WSJ1) Complete (LDC94S13A),
- AMI Meeting Corpus (http://groups.inf.ed.ac.uk/ami/download/),
- RT-03 MDE Training Data Speech (LDC2004S08),
- Santa Barbara Corpus of Spoken American English Part II (LDC2003S06)

Data for FA model:

- Czech Speecon database (ELRA-S0298, child voices only),
- UK English Speecon database (ELRA-S0215, child voices only),
- US English Speecon database (ELRA-S0233, child voices only),
- TIMIT Acoustic-Phonetic Continuous Speech Corpus (LDC93S1),
- CSR-I (WSJ0) Complete (LDC93S6A),
- CSR-II (WSJ1) Complete (LDC94S13A),

2.2 i-Vector extraction for the domain classifier

Data for UBM (only 2-3 hours of data from each corpus):

- LibriSpeech (http://www.openslr.org/12/, training sets only),
- DIHARD Development Data (LDC2018E31),
- AMI Meeting Corpus (http://groups.inf.ed.ac.uk/ami/download/),
- RT-03 MDE Training Data Speech (LDC2004S08),
- Santa Barbara Corpus of Spoken American English Part II (LDC2003S06)

Data for FA model:

- DIHARD Development Data (LDC2018E31),
- LibriSpeech (http://www.openslr.org/12/, training sets only)

2.3 Domain classifier (ANN):

- DIHARD Development Data (LDC2018E31),
- LibriSpeech (http://www.openslr.org/12/, training sets only),
  (10 randomly chosen 10 min recordings were used as additional LibriVox data)

For the training process, we excluded 1 randomly selected recording from each domain for testing the performance of the network. The final network was trained on all data, but with the same number of epochs.

3 Detailed description of algorithm

Our system [1–3] follows an i-vector-based approach, as introduced in [4–6]: First, each recording is divided into short segments and i-vectors are extracted. Then, a clustering method is used in order to determine which parts of the signal were produced by the same speaker. For the DIHARD Challenge, we have also introduced a domain classifier that determines the source of each recording and selects the most suitable system configuration.

3.1 Feature extraction

We used Linear Frequency Cepstral Coefficients (LFCCs), Hamming window of length 25 ms with 10 ms shift. There are 40 triangular filter banks linearly spread across the frequency spectrum, and 25 LFCCs are extracted. The resultant 50-dimensional feature vector ($D_f = 50$) also includes delta coefficients.

3.2 Speech activity detection

This particular system was only used for Track 1 - information about speaker activity was received as gold speech segmentation.

3.3 Segmentation

Each recording is split into multiple individual speech regions by breaking it on any non-speech longer than 0.5 s. These speech regions are then split at regular intervals into segments with a length of 2 s and with a 1 s overlap between neighboring segments. If the remainder is shorter than 1 s, we extend it to 1 s by adding frames from the preceding segment.

3.4 Segment description

Each segment is represented by an i-vector derived from the supervector of accumulated statistics - zeroth and first statistical moments of data related to a UBM as a GMM with $M = 1024$ components. The dimensionality of this supervector is reduced by Factor Analysis (FA) [7, 8] into $D_w = 100$ and we have used conversation-dependent Principal Component Analysis (PCA) [6] to reduce the dimension further into 3 or 9 (depending on the specific data - see Tab. 1).

3.5 Clustering

Given i-vector representations of the extracted segments, we perform a clustering into sets of i-vectors describing different speakers. For clustering we mainly use agglomerative hierarchical clustering (AHC). The system starts with each i-vector in a separate cluster and then merges the closest pairs until it reaches a stopping point. The distance between two clusters is calculated as the average cosine distance between each pair of

i-vectors. The stopping condition is a combination of maximum merging distance and a minimum and a maximum number of clusters: First, we perform AHC by merging the closest pairs of clusters until the lowest distance exceeds a specific threshold. If the resulting number of clusters is not within the expected range, we adjust the stopping point so that we reach either the minimum or maximum allowed number of clusters. If we are certain there are only 2 speakers in the conversation, we use the k-means algorithm instead (also with cosine distance). All mentioned parameters were selected on a per-corpus basis using the development set (see section 3.7).

As this system uses overlapping segments (see section 3.3), some frames will receive two conflicting labels. We exclude such regions from the obtained clusters and resolve this using a resegmentation-like process (described in the next section).

3.6 Reclassification of segment overlaps

The "ws-cr" system does not use a full resegmentation. However, it employs a similar method to resolve conflicting labels in overlapping regions between neighboring segments:

We compute GMMs over the (non-conflicted) feature vectors, one GMM for each speaker cluster. Then the problematic regions are redistributed frame by frame according to the likelihoods of the GMMs, filtered by a Gaussian window (length 75 ms with shift 50 ms) to smooth the peaks in the likelihoods. The number of GMM components depends on the amount of data in each cluster (approximately 1 GMM component per 4 s of data, rounded down to the nearest power of 2) and ranges between 1 and 64.

3.7 Domain Classification

The domain classifier was implemented as a neural network in Keras, using TensorFlow. It was trained with one hidden layer (2048 neurons, tanh activation function) followed by 0.9 dropout and the output layer as softmax into 9 categories, batch size = 32, epochs = 25, categorical cross-entropy, "adam" optimizer. The remaining hyperparameters were left at default values.

The input to the classifier is a single i-vector calculated over the entire recording. It was extracted using a UBM with 512 components and a FA model with dimension 100. The network outputs the probability of each of the 9 corpora in the DIHARD development set.

Because of the presence of unseen corpora in the evaluation set, we have also added a threshold (= 0.5) on the output probability from the classifier and categorize lower-scoring recordings as "unknown domain".

The experimentally chosen parameters for each corpus in the development set are listed in Table 1 and were chosen as follows:

- The target number of speakers was observed directly from the development data (with the exception of VAST and SEEDLingS).
- PCA dimension reduction was selected from three options: reduction to dimension 3 or 9, or no reduction.

- The AHC stopping threshold was found as the setting with the lowest DER at the clustering stage - without overlap reclassification (tested in increments of 0.02).
- For VAST and SEEDLingS, lowest overall DER was achieved with only a single cluster in each recording (i.e. the system did not work well on these data).

Table 1. Experimentally chosen parameters for each corpus, system "ws-cr".

corpus	Clustering	No. spk	AHC Thresh.	PCA dim
SEEDLingS	-	1	-	-
SCOTUS	AHC	5-10	0.64	9
DCIEM	k-means	2	-	3
ADOS	k-means	2	-	3
YouthPoint	AHC	3-5	0.64	9
SLX	AHC	2-6	0.74	9
RT-04S	AHC	3-10	0.60	9
LibriVox	-	1	-	-
VAST	-	1	-	-
other	AHC	3	-	9

4 Hardware requirements

The main body of the system was implemented in Matlab. The code was not optimized with regards to execution time (e.g., intermediate results were saved to the disk).

The feature extractor, i-vector extractor and GMM adaptation were all in separate executables (C++), called from Matlab.

Domain classification was computed separately, using Keras with TensorFlow.

4.1 Training of UBM models for i-vector extraction:

Hardware:

- CPU 8-core Intel Xeon E5-2650v2 2.60 GHz
- GPU 2x nVidia Tesla K20, 5GB, 1000 GFLOPS
- 4 GB RAM
- 10 GB required storage

Total training time: approx. 96 hours

4.2 Training of FA models for i-vector extraction:

Hardware:

- CPU 8-core Intel Xeon E5-2650v2 2.60 GHz
- 22 GB RAM
- 10 GB required storage

Total training time: approx. 48 hours

4.3 Training of UBM and FA models for the domain classifier:

Hardware:

- CPU Intel(R) Core(TM) i7 cpu - 1 core used, 3.07GHz
- GPU NVIDIA GeForce 1080 Ti, 11 GB VRAM, 11,340 GFLOPS
- 32 GB RAM
- 5.5 GB required storage

Total training time: approx. 5 hours

4.4 Training of the domain classifier ANN:

Hardware:

- CPU Intel(R) Core(TM) i7 cpu - 1 core used, 3.07GHz
- GPU NVIDIA GeForce 1080 Ti, 11 GB VRAM, 11,340 GFLOPS
- 32 GB RAM
- 5.5 GB required storage

Implemented in Keras, using TensorFlow.
Total training time: approx. 5 minutes

4.5 Execution times to process an average 10 minute recording:

Hardware (main system):

- CPU Intel(R) Core(TM) i7 cpu - 4 cores used, 3.07GHz
- GPU NVIDIA GeForce 1080 Ti, 11 GB VRAM, 11,340 GFLOPS
- 32 GB RAM
- 80 MB required storage

Execution time:

- Domain classification: $\sim$30 s
- Main system: 110 s
- **Total time:** $\sim$140 s

5 Acknowledgements

The work was supported by the project no. P103/12/G084 of the Grant Agency of the Czech Republic. Access to computing and storage facilities (CESNET LM2015042) is greatly appreciated. The authors would also like to thank their friends and colleagues who provided recordings of their children as additional training data.

References

1. Z. Zajíc, M. Kunešová, and V. Radová, "Investigation of Segmentation in i-Vector Based Speaker Diarization of Telephone Speech," in *Specom*. Budapest: Springer, 2016, pp. 411–418.
2. M. Hrúz and Z. Zajíc, "Convolutional Neural Network for Speaker Change Detection in Telephone Speaker Diarization System," in *ICASSP*. New Orleans: IEEE, 2017, pp. 4945–4949.
3. Z. Zajíc, M. Hrúz, and L. Müller, "Speaker diarization using convolutional neural network for statistics accumulation refinement," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, Stockholm, 2017, pp. 3562–3566.
4. G. Sell and D. Garcia-Romero, "Speaker Diarization with PLDA I-vector Scoring and Unsupervised Calibration," in *IEEE Spoken Language Technology Workshop*, South Lake Tahoe, 2014, pp. 413–417.
5. M. Senoussaoui, P. Kenny, T. Stafylakis, and P. Dumouchel, "A Study of the Cosine Distance-Based Mean Shift for Telephone Speech Diarization," *Audio, Speech and Language Processing*, vol. 22, no. 1, pp. 217–227, 2014.
6. S. Shum, N. Dehak, E. Chuangsuwanich, D. Reynolds, and J. Glass, "Exploiting Intra-Conversation Variability for Speaker Diarization," in *Interspeech*, Florence, 2011, pp. 945–948.
7. P. Kenny and P. Dumouchel, "Experiments in Speaker Verification Using Factor Analysis Likelihood Ratios," in *Odyssey*, Toledo, 2004, pp. 219–226.
8. L. Machlica and Z. Zajíc, "Factor Analysis and Nuisance Attribute Projection Revisited," in *Interspeech*, vol. 2, Portland, 2012, pp. 1570–1573.