IberSPEECH 2018
21-23 November 2018, Barcelona, Spain

DNN-based Embeddings for Speaker Diarization in the AuDIaS-UAM System for the Albayzin 2018 IberSPEECH-RTVE Evaluation

Alicia Lozano-Diez, Beltran Labrador, Diego de Benito, Pablo Ramirez, Doroteo T. Toledano

AuDIaS - Audio, Data Intelligence and Speech Research Group,
Universidad Autonoma de Madrid (UAM), Madrid, Spain

alicia.lozano@uam.es

Abstract

This document describes the three systems submitted by the AuDIaS-UAM team for the Albayzin 2018 IberSPEECH-RTVE speaker diarization evaluation. Two of our systems (primary and contrastive 1 submissions) are based on *embeddings* which are a fixed length representation of a given audio segment obtained from a deep neural network (DNN) trained for speaker classification. The third system (contrastive 2) uses the classical i-vector as representation of the audio segments. The resulting embeddings or i-vectors are then grouped using Agglomerative Hierarchical Clustering (AHC) in order to obtain the diarization labels. The new DNN-embedding approach for speaker diarization has obtained a remarkable performance over the Albayzin development dataset, similar to the performance achieved with the well-known i-vector approach.

Index Terms: speaker diarization, embeddings, i-vectors, AHC

1. Introduction

The AuDIaS-UAM submission for the Speaker Diarization (SD) evaluation consisted of three different systems, two of them based on embeddings (also known as *x-vectors*) [1] extracted from a Deep Neural Network (DNN) trained for speaker classification, and a third one based on the classical total variability i-vector model [2].

Our systems are submitted for the closed-set condition since they are trained using the training and development datasets made available for this evaluation, briefly described in Section 2.

For all our systems, we extract frame-level features as described in Section 3. Then, using those features, we train either a DNN or an i-vector extractor in order to obtain a fixed-length representation of an audio segment (regardless its duration), as presented in Sections 4 and 5, respectively. These models are trained using a segmentation based on the reference labels (RTTM files) provided by the organizers for training and development purposes.

We kept three development files from RTVE dataset as held out set for diarization performance evaluation. For these recordings, in order to discard fragments where just music (without speech) is present, we developed a DNN-based music detector described in Section 6. Then, diarization labels are obtained by means of Agglomerative Hierarchical Clustering (AHC) performed over non-music segments. This last step is summarized in Section 7. Finally, in Section 8 we show results of the Audias-UAM submitted systems over our development dataset.

2. Training and Development Datasets

We used the three datasets provided by the organizers for this evaluation: Aragón Radio, 3/24 TV channel and RTVE 2018 (*dev2*) [3]. For training, we used all data from the first two databases and 8 files (out of 12) from RTVE 2018. This set was segmented according to the time alignments specified by the RTTM files.

In order to evaluate the speaker diarization performance of our systems, we used 3 files from the RTVE 2018 development set (approximately 4 hours) not used for training.

All the audio files were down-sampled to 16kHz.

3. Feature Extraction

All our systems are based on MFCC features extracted using Kaldi [4]. Each feature vector consists of 20 MFCCs (including C0), computed every 10 ms with a 25 ms "Povey" window (default in Kaldi, similar to Hamming window).

For the i-vector system, these 20-dimensional features are normalized using cepstral mean normalization over a 3 s sliding window, and augmented with their first and second derivatives (Δ and $\Delta\Delta$), providing a final 60-dimensional feature vector.

For the DNN-embedding systems, the *raw* 20-dimensional MFCC feature vectors are used to feed the network without applying channel compensation or adding temporal information. However, global zero-mean and unit-variance normalization is performed over the whole training set.

4. DNN-based Embedding Systems

Two of our submitted systems are based on DNN-based embeddings [1]. An embedding is a fixed-length representation of a given utterance or audio segment learned directly by a DNN. Typically, this DNN is trained for speaker classification.

In our case, we used an architecture based on Bidirectional Long Short-Term Memory (BLSTM) recurrent neural networks similar to the one used in [5] for language recognition, whose configuration was adjusted to the available data and the speaker diarization task. The architecture (sequence-summarizing DNN) used consists on a frame-level part composed of two BLSTM layers (with 128 cells each) and a fully-connected layer of 500 hidden units. Then, a pooling layer computes the mean and standard deviation over time to the outputs of the previous layer, followed by two fully connected layers (embeddings a and b, respectively) of 50 hidden units each and a softmax output layer with 3124 output units, working on an utterance-level basis. All the layers (except the output layer) use *sigmoid* non-linear activation. A graphical representation of the architecture is depicted in Figure 1.

The size of the output layer (3124) corresponds to the number of speakers considered in our training dataset. However, it should be pointed out that due to the lack of actual speaker identification labels and the segmentation based on the RTTM labels for diarization, each recording was considered to have

10.21437/IberSPEECH.2018-46

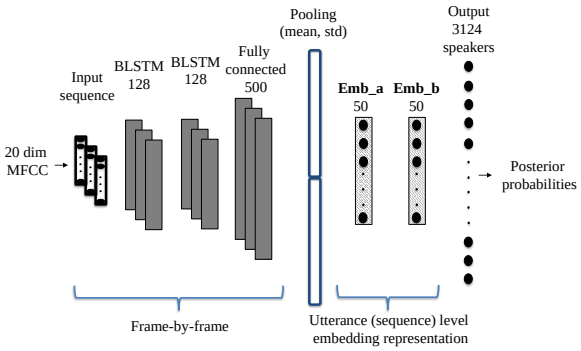

Figure 1: *Architecture used for the DNN-based embedding systems used as primary and contrastive 1 submissions to the speaker diarization evaluation.*

different speakers than the rest (which is usually not the case). Then, even though two segments might be labeled as spoken by different speakers, they could belong to the same person.

The DNN was trained using stochastic gradient descend to minimize the multi-class cross-entropy criteria for speaker classification. For training purposes, the network was fed with 3 s long sequences of 20-dimensional MFCC feature vectors.

After training, embeddings were extracted for each 3 s fragment of the development and test recordings, with a shift of 0.5 s. Each segment was forwarded through the network up to the first embedding layer (embedding a), providing a 50-dimensional embedding every 0.5 s (corresponding to 3 s sequences).

This system was implemented using Keras [6].

Embeddings obtained from this system were used for the **primary system** and the **contrastive system 1**, which differ in the clustering stage as described in Section 7.

5. I-vector System

As **contrastive system 2**, we used the classical total variability i-vector [2] modeling.

To develop this system, an UBM of 1024 Gaussian components was trained using the 60-dimensional MFCC+Δ+$\Delta\Delta$ features described in Section 3, and a 50-dimensional total variability subspace was derived from the Baum-Welch statistics of the training segments (obtained according to RTTM timestamps). The configuration was taken from previous speaker diarization systems developed in our research group.

After training, each development and test recording was processed in order to obtain a stream of i-vectors every 0.5 s (as with embeddings) with a sliding window of length 3 s.

This system was implemented using Kaldi [4].

The speaker diarization was performed using clustering on top of the resulting streams of i-vectors (see Section 7).

6. Music Detection

In order to discard segments where just music was present, we developed a music/speech classifier based on DNNs [7].

This system is trained using 150 h of audio from Google Audio Set [8], a dataset consisting of 10 seconds audio segments extracted from YouTube videos. Our architecture is composed of six bidimensional convolutional neural network (CNN) layers which operate on the Mel-spectrogram of the audio signals, followed by one LSTM layer and a final fully-connected layer prior to the output layer.

The output layer has 4 output units which correspond to the classification into music, speech, speech + music and none of them. For this evaluation we used just the probabilities of belonging to the music class in order to filter out segments that contain only music. This way, just the embeddings or i-vectors corresponding to speech (or speech with music) segments were used to perform the clustering stage.

In order to extract the probability of a segment to contain music, Mel-spectrograms corresponding to test recordings have been computed and split into a stream of 10 second segments to fit the input size of the music/speech classifier. The separation between consecutive segments is 0.5 s in order to identify each Mel-spectrogram with an embedding or i-vector in the stream.

7. Agglomerative Hierarchical Clustering

In order to obtain the speaker diarization labels, we used Agglomerative Hierarchical Clustering (AHC) over the resulting stream of either embeddings or i-vectors (depending on the system) for a given development or test recording. Embeddings or i-vectors corresponding to music segments according to our music detector were discarded previously to the clustering step. This stage was implemented in Python using the scikit-learn toolbox [9].

Thus, AHC is applied to the resulting sequence of vectors corresponding to a specific audio file. We used cosine distance for i-vectors and euclidean distance for embeddings. The number of clusters is controlled by the threshold of the distance to merge clusters, whose value was optimized on the development set. The linkage method used was the average of the vectors. This clustering was applied once for the contrastive systems.

However, for the primary system we applied first AHC over the whole set of vectors with a lower threshold to allow a bigger number of clusters, and then, a second AHC stage was applied to group the centroids of the previous clusters. This was done in order to help the clustering grouping speaker identities instead of vectors similar due to their closeness in time. The centroids were computed as the mean vector over all the points labeled as belonging to the same cluster in the first AHC stage.

Finally, for all the systems, we post-processed the clustering labels by filtering out clusters that grouped less than 10 s of audio. This was done in order to reduce false alarm in terms of clusters that do not group a different speaker but segments further than the chosen threshold.

8. Development Results

Table 1 shows the results obtained by our systems in our development set (three files from RTVE 2018 dev 2 partition).

In our development set, both systems based on embeddings obtained similar results, especially in terms of missed and false alarm speaker time. The difference in these two metrics with respect to the third system (based on i-vector) is also not significant, while the performance differs mainly due to the speaker error time. This might be related to the system settings and thresholds selected for the clustering, which would merge different speakers into one cluster or vice-versa.

Even though the i-vector system obtained better performance in our development dataset than the embedding-based systems, we submitted as primary system one of these systems due to the novelty of this technique with respect to the well-known i-vectors for speaker diarization and related tasks.

Table 1: *Performance of the Audias-UAM submission for the Albayzin IberSPEECH-RTVE speaker diarization evaluation over the development dataset (approximately 4 hours, 3 different recordings from 2 different shows. The performance is shown in % of scored speaker time.*

System	Missed	False Alarm	Speaker Error	DER
Embeddings + double AHC (primary)	2.2	1.2	14.6	18.02
Embeddings + simple AHC (contrastive 1)	2.1	1.2	15.3	18.65
i-vectors + simple AHC (contrastive 2)	2.9	1.3	12.9	17.22

9. Acknowledgements

This work was supported by project DSSL: Redes Profundas y Modelos de Subespacios para Detección y Seguimiento de Locutor Idioma y Enfermedades Degenerativas a partir de la Voz (TEC2015-68172-C2-1-P), funded by Ministerio de Economía y Competitividad, Spain and FEDER.

10. References

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

[2] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 19, no. 4, pp. 788–798, 2011. [Online]. Available: http://dx.doi.org/10.1109/TASL.2010.2064307

[3] "Albayzin evaluation: Iberspeech-rtve 2018 speaker diarization challenge," http://catedrartve.unizar.es/reto2018/EvalPlan-SpeakerDiarization-v1.3.pdf.

[4] 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 *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011, iEEE Catalog No.: CFP11SRW-USB.

[5] A. Lozano-Diez, O. Plchot, P. Matějka, and J. Gonzalez-Rodriguez, "Dnn based embeddings for language recognition," in *Proceedings of ICASSP*, April 2018.

[6] "Keras: The python deep learning library," https://keras.io/.

[7] D. de Benito Gorrón, "Detección de voz y música en un corpus a gran escala de eventos de audio," https://repositorio.uam.es/handle/10486/684843, June 2018.

[8] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.