INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Speaker-Corrupted Embeddings for Online Speaker Diarization

Omid Ghahabi, Volker Fischer

EML European Media Laboratory GmbH, Berliner Straße 45, 69120 Heidelberg, Germany
(omid.ghahabi | volker.fischer)@eml.org

Abstract

Speaker diarization is more challenging in presence of background noise or music, frequent speaker changes, and cross talks. In an online scenario, the decision should be made at time, given only the current short segment and the speakers detected in the past, which makes the task even harder. In this work, an online robust speaker diarization algorithm is proposed in which speech segments are represented by low dimensional vectors referred to as speaker-corrupted embeddings. The proposed speaker embedding network is a deep neural network which takes speaker-corrupted supervectors as input, uses variable ReLU (VReLU) as an activation function, and tries to discriminate the background speakers. Speaker corruption is performed by adding supervectors built by 20 speech frames from other speakers to the supervectors of a given speaker. It is shown that speaker corruption, VReLU, and input dropout increase the generalization power of the proposed network. To increase the robustness, the proposed embeddings are concatenated with LDA transformed supervectors. Experimental results on the Albayzin 2018 evaluation set show a competitive accuracy, more robustness, and much lower computational cost compared to typical offline algorithms.

Index Terms: Online Speaker Diarization, Speaker Embedding, Variable ReLU, Speaker Corruption

1. Introduction

Speaker diarization aims to determine who is speaking when in a multi-speaker conversation. The conversation could be, for example, in a broadcast show, a meeting, or a telephone call. Any prior knowledge regarding the speakers in the conversation could be a great help, but usually it is not available in real applications. The identity and the number of the speakers are typically unknown to the system which makes diarization more difficult compared to other common speaker recognition tasks.

Most of the research in speaker diarization has focused on offline applications where usually the whole conversation is first segmented to detect the speech and the speaker change points, and then the speech segments are clustered. Agglomerative hierarchical clustering with a distance measure like Bayesian information criterion (BIC) is often used [1]. Recently, i-vector based (e.g., [2,3]) or deep learning based (e.g., [4–6]) modifications are also applied. However, none of the offline techniques can be easily used in an online scenario where there is no access to the whole conversation at once and decision must be taken with an acceptable accuracy and low latency.

Several online approaches have been proposed over the last few years [7–17]. Most of these approaches are based on traditional GMM-UBM techniques (e.g., [7–9, 12, 14]). Usually, two gender-dependent UBMs are used as the seed models. Every incoming audio segment is compared with the current speaker models and labeled based on a predefined threshold. The speaker models are MAP adapted GMMs which are updated or created at runtime. It is shown in [14] that pre-enrollments of speakers in a meeting play an important role for an online system to work, otherwise the performance will be far from that of an offline system. A similar approach is proposed in [10] in which an offline recognition process is run in parallel using all data from time zero to the current processing segment as a help to the online recognition. In [15], with the help of an online automatic speech recognition system, words boundaries are expected to be potential speaker change points. Gaussians and BIC are used for change point detection and i-vectors are used for clustering. In [16], speaker embeddings are used rather than i-vector for speaker representation and support vector machines are used for clustering. Although the latency is 1 sec, the diarization error rate is almost doubled when compared to the offline system.

In this paper, we introduce an online speaker diarization algorithm which benefits from robust voice activity detection (VAD), adaptive speaker-dependent thresholds, and an efficient score normalization. Speech segments are represented by newly proposed low dimensional vectors referred to as speaker-corrupted embeddings. The proposed embeddings are alternatives to traditional i-vectors [18] and the recent x-vectors [19] with much lower computational cost and suitable for online processing. The proposed embedding network is a DNN taking advantage of input speaker-corrupted supervectors and efficient variable ReLU (VReLU) [20] as an activation function trying to discriminate the background speakers. Speaker corruption is performed by adding supervectors, built by only 20 speech frames randomly selected from other speakers, to the supervectors of a given speaker. Speaker corruption makes the training process harder and increases the generalization power of the network on unseen data. In the test phase, embeddings are extracted by only a single transformation of the input supervectors. Experimental results on the recent Albayzin evaluation challenge [21, 22] show a competitive accuracy, more robustness, and much lower computational cost compared to typical offline algorithms.

2. Proposed Online Speaker Diarization

Figure 1 summarizes the proposed online algorithm in which the input audio signal is processed segment by segment every 0.1 sec to decide if the current segment is speech (sp) or nonspeech (nsp) based on the zero order Baum-Welch statistics and the VAD algorithm proposed in [23]. If the short segment is detected as nonspeech, it will be discarded otherwise the Baum-Welch statistics are computed and accumulated over speech segments until the decision time is reached. Decision time is based on predefined maximum speech or successive nonspeech durations which are 2 sec and 0.6 sec in this work, respectively. Accumulated statistics are converted to a supervector which is further centralized by the UBM mean supervector. The resulting supervector is then converted to a lower dimensional speaker vector given a transformation matrix which can

Copyright © 2019 ISCA

http://dx.doi.org/10.21437/Interspeech.2019-2756

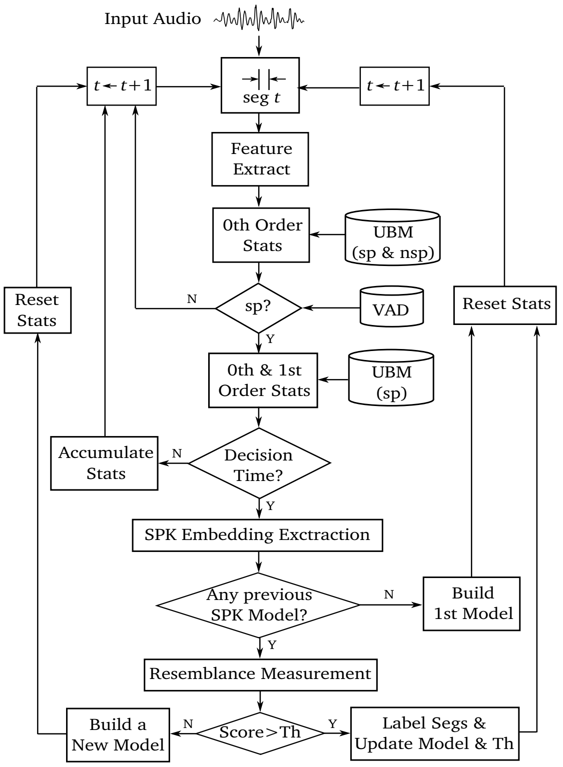

Figure 1: *Proposed online speaker diarization algorithm.*

be obtained by LDA, the proposed embedding network, or a combination of both. The speaker vector is compared with the current speaker models using cosine similarity. If the model with the highest similarity gives a score higher than the speaker-dependent threshold, all the segments corresponding to the accumulated statistics are labeled as the selected speaker and the speaker model and its threshold are updated. Otherwise, a new model will be created. Before a new model creation, the speech segment is divided into two halves. For each half, a speaker vector is created and compared with another. If the two halves are similar enough in terms of the speaker identity, the statistics are merged and the new model is created. Otherwise, each half is assigned to one of the current speaker models. In other words, a new speaker model is created only if two halves are similar enough. More technical details about every part of the algorithm are given as follows.

2.1. Voice Activity Detection

The VAD proposed in [23] is used in this work. The proposed VAD is a hybrid supervised/unsupervised model taking advantage of a large amount of unlabeled data to train a UBM and a small amount of labeled data to model the speech and non-speech classes with two very low dimensional vectors based on zero order Baum-Welch statistics obtained from the UBM. Given little amount of labeled speech and nonspeech feature vectors and the UBM, the zeroth order Baum-Welch statistics are computed for each class and saved as the VAD vectors. In the testing phase, the zeroth order statistics vector of an unknown short duration segment is first computed (ω) and the resemblance ratio score will be based on the cosine distance between ω and each VAD vector.

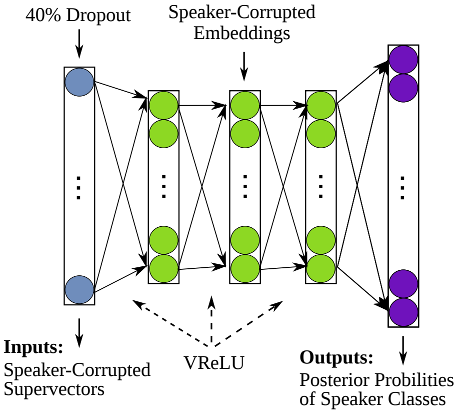
40% Dropout
Speaker-Corrupted Embeddings
Inputs:
Speaker-Corrupted Supervectors
VReLU
Outputs:
Posterior Probilities of Speaker Classes

Figure 2: *The network architecture of the proposed Speaker-Corrupted (SC) embeddings.*

2.2. Speaker-Corrupted Embeddings

The most well-known low dimensional vector representation for speaker recognition is i-vector [24] which works very well for long duration signals. However, i-vectors usually do not work very well for very short segments and are computationally costly and, therefore, prohibitive for real-time applications. Alternatively, the recent deep learning based vector representations, which are usually referred to as speaker embeddings, e.g., d-vector [25] and x-vector [26], have shown superior quality for short segments compared to i-vectors. Nevertheless, the current speaker embeddings are still computationally expensive for a fast online speaker diarization.

Both i-vector and x-vector extraction processes include two main steps, namely statistics computation and dimension reduction. Statistics are usually computed with a time-delay neural network and a background GMM (UBM) for x-vectors and i-vectors, respectively. Dimension reduction is performed with a feedforward neural network for x-vectors and with factor analysis for i-vectors. We propose in this section to use the lower computational part of each technique and combine them to create an efficient vector representation technique in terms of both accuracy and speed. Therefore, we use the Baum-Welch statistics to create supervectors and do the dimension reduction through a speaker discriminative neural network.

Figure 2 shows the proposed architecture for speaker embedding extraction. The network is composed of three hidden layers where the layer in the middle is found less sensitive and considered as the embedding layer. The activation function for the hidden layers is Variable ReLU (VReLU) [20] and for the output layer is softmax. The network is trained to discriminate the background speakers. The input supervectors are computed as follows,

$$\boldsymbol{s}^a = (\boldsymbol{\mu}_1^a, \boldsymbol{\mu}_2^a, ..., \boldsymbol{\mu}_M^a)^T \tag{1}$$

$$\boldsymbol{\mu}_i^a = \frac{\mathcal{N}_i(\boldsymbol{u}_{sp})}{\mathcal{N}_i(\boldsymbol{u}_{sp}) + r} \left[\frac{\mathcal{F}_i(\boldsymbol{u}_{sp})}{\mathcal{N}_i(\boldsymbol{u}_{sp})} - \boldsymbol{\mu}_i^{ubm} \right] \tag{2}$$

where $\boldsymbol{\mu}_i^a$ is the adapted mean vector for Gaussian i and speaker a, which is normalized with the corresponding mean vector from UBM ($\boldsymbol{\mu}_i^{ubm}$), r is a fixed relevance factor, and $\mathcal{N}_i(\boldsymbol{u}_{sp})$ and $\mathcal{F}_i(\boldsymbol{u}_{sp})$ are, respectively, zero and first order statistics for the speech feature vectors $\boldsymbol{u}_{sp}$.

In order to increase the generalization power of the network and the robustness on unseen data, we have proposed to train the network as described in the following.

Data Preparation. As the background data is automatically labeled, the speakers with too few and too much speech data are considered as non-reliable labeled speakers and are discarded from the training data. The remaining data is referred to as cleaned data in section 3. The available speech data for the remaining background speakers is chopped into segment sizes from 0.5 to 2 sec and for each segment a supervector is extracted as an input data. As a common problem in neural networks, if the amount of data for different classes is highly unbalanced, the network will be biased towards the majority, more frequent, classes and the performance of the network will go down. The common solution is to randomly discard some samples from the majority classes and just repeat the samples from the minority, less frequent, classes to keep the number of samples balanced across all classes. However, for the minority classes, we propose to randomly select 20 speech frames from the same speaker and add to the repeated chopped segments to be considered in statistic computation and make them slightly different than the original segments.

Speaker Corruption. In order to increase the generalization power of the network, we furthermore propose to confuse the network by speaker corruption of the input data. Speaker corruption is performed by adding low quality supervectors from other speakers to the supervectors of a given speaker in the training data. Low quality supervectors are built with only 20 speech frames and will have very low energy after UBM mean normalization, as in eq. 2, compared to the original supervectors. Choosing only 20 frames guarantees to have always less corruption data than the shortest segments (0.5 sec) used in training. Thus, the degree of corruption depends on the duration of the original segment (0.5 to 2 sec), the longer segment duration the less corruption.

Variable ReLU. The activation function used for hidden layers is VReLU proposed in [20] and computed as follows,

$$f(x) = \begin{cases} x & x > \tau \\ 0 & x \leq \tau \end{cases} \quad , \quad \tau \in N(0,1) \tag{3}$$

where the activations less than the threshold τ are zeroed out, rather than the fixed threshold zero in ReLU. Threshold τ is randomly selected from a normal distribution $N(0,1)$. It is shown in [20] that VReLU has higher generalization power and more compatibility with PLDA scoring compared to ReLU for unsupervised speaker embedding extraction. We will confirm this in section 3 for the proposed supervised embeddings as well.

Input Dropout. In order to avoid training the network with the same input data in each epoch, we randomly show only a different part of each input vector every time. In other words, in each minibatch, a percentage of each input supervector is dropped out.

Embedding Extraction. One of the interesting properties of VReLU is that it uses nonlinearity in training and linearity in testing. Although the prediction accuracy of the output classes will decrease in deeper networks compared to ReLU, it is not important for embedding extraction. This means that the connection matrix between the input and the first hidden layer can be multiplied by the connection matrix between the first and the second hidden layer to create a single transformation matrix from input supervectors to the proposed speaker embedding vectors referred to as speaker-corrupted (SC) embeddings. We can further reduce the dimensions of embeddings with an LDA transformation which gives some accuracy loss. However, our experiments showed that if the eigenvectors are computed only on either within or between class covariance matrices, the dimension of embedding vectors can be reduced from 300 down to 40 without any accuracy loss.

As supervectors are going to be extracted from very short segments, a big UBM is usually not required. In this work, both the UBM size and the feature vectors are very small leading to much lower dimensional supervectors compared to common speaker recognition tasks. Thus, an efficient alternative dimension reduction technique is to use directly LDA on supervectors. As it will be shown in section 3, LDA transformed supervectors give more stable decision thresholds but less accuracy compared to SC embeddings. To take advantage of both vectors, they can be combined either in the score or vector level. To keep the complexity and the computational cost of the speaker diarization algorithm the same for all the vectors, we concatenate length-normalized 260 LDA-transformed supervectors with length-normalized 40 dimensional embeddings computed as mentioned above. Speaker models will then be the average over the time of speaker vectors assigned to each detected speaker.

3. Experiments

Diarization Error Rate (DER) is used for the performance measurement in this work. DER includes the time assigned to wrong speakers (speaker error), missed and false alarm speech time. Equal Error Rates (EER) are used for the quality comparison of the proposed speaker embeddings. Given the reference speaker labels of a given audio recording, the speech segments of each speaker are chopped into short segments of 0.5 to 2 sec duration. Then all the short segments within an audio recording are compared to each other, resulting to true and false scores which are later used for EER computation per each recording. The final EER presented in the tables is the average EER obtained on all the development recordings.

3.1. Database and Setup

Three sets of data provided in Albayzin 2018 challenge are used for training and development of the diarization systems in this work. The first set is about 440 hours unlabeled broadcast news recordings. The second one is about 75 hours automatically labeled with another speaker diarization system. The last dataset is about 16 hours of human-revised labeled data. The final evaluation data set is also about 16 hours of recordings from other channels. This data is collected from RTVE2018 [22], Aragon Radio, and 3/24 TV channel databases. The details can be found in [21, 22].

Feature vectors are extracted every 10 msec with a 25 msec window. There are 16 dimensional MFCCs along with their deltas for VAD, and 30 dimensional static MFCCs for speaker vectors. Features are mean normalized with a 3 sec sliding window. UBM for both VAD and speaker vectors are GMMs with 64 Gaussian mixtures each.

3.2. Results

To monitor the training procedure of the proposed embedding network, we have used the average EER obtained on 2 sec chopped segments, as mentioned above, rather than the common validation error rate. The speakers used for EER computation are totally different than those used for training the network. Figure 3 (top) shows the effect of each training proposals mentioned in section 2.2 by comparing the EER variation over

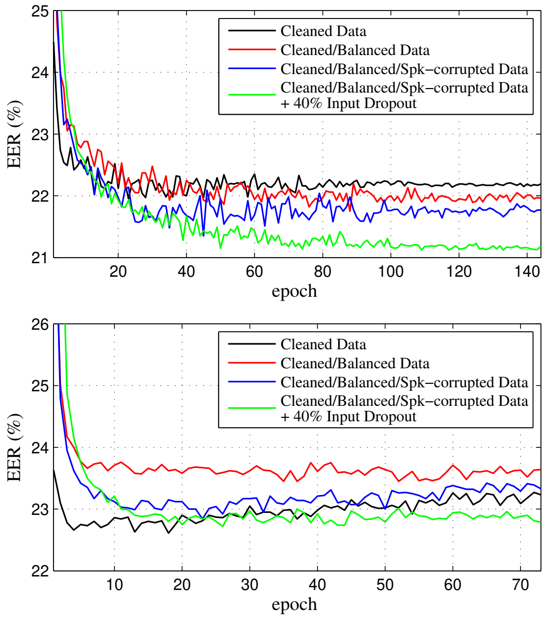

Figure 3: *The effect of data preparation, speaker corruption, and input dropout on the training of the proposed embedding network trained with* ***VReLU*** *(top) and* ***ReLU*** *(bottom).*

Table 1: *Speaker embedding quality comparison obtained on the dev2 partition of Albayzin 2018.*

Vector Representation	**Dimension**	**2 sec**		**0.5-2 sec**	
		EER	σ_{th}	EER	σ_{th}
(1) i-Vector	400	29.47	0.022	35.75	0.018
(2) i-Vector + LDA	300	26.21	0.038	31.04	0.036
(3) Supervector + LDA	300	25.08	0.027	33.12	0.016
(4) SC Embedding	300	21.16	0.086	26.85	0.058
Combined (3) & (4)	260+40	**20.82**	0.060	**26.66**	0.040

training epochs. Balancing the input data shows a small EER improvement with much fewer amount of data. Speaker corruption shows a noticeable EER reduction even from the beginning of training. Adding an input dropout of 40% shows a persistent further improvement.

Figure 3 (bottom) shows the same experiments but with a network trained with ReLU activation function. The results show a totally different behavior of the network. The network overfits at the very beginning of training. However, balancing the data helps to avoid overfiting but with a cost of performance reduction. This shows that the network trained with ReLU is very sensitive to the amount of data as expected. Although both speaker corruption and input dropout help to improve the performance, the network trained with VReLU still shows higher quality and better generalization.

Table 1 compares the EERs, obtained on 2 sec chopped segments and a pool of segments from 0.5 to 2 sec with a step size of 0.5 sec, for different speaker vectors. We have also computed the standard deviation of the thresholds corresponding to EERs (σ_{th}) over different audio recordings to see how the thresholds are stable by changing the channel and recording. Lower standard deviation is considered as more stable. Among all, the proposed SC embeddings show significantly higher accuracy but lower threshold stability. The combination of SC embeddings and LDA transformed supervectors slightly improves both EER and σ_{th} with the same computational cost.

Table 2: *DER comparison of the proposed low cost speaker embeddings obtained on the dev2 partition of Albayzin 2018.*

Vector Representation	**DER** (%)		
	La noche 24H	Millennium	Overall
Supervector + LDA	37.53	**8.50**	24.45
SC Embedding	29.06	19.99	24.61
Combined	**26.43**	16.10	**21.53**

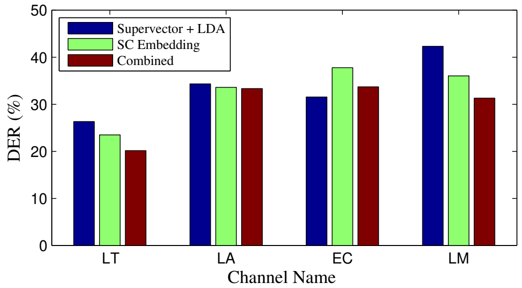

Figure 4: *DER comparison of the proposed speaker embeddings per channel obtained on eval set of Albayzin 2018 challenge.*

Table 3: *DER comparison of the proposed low cost speaker embeddings obtained on the eval partition of Albayzin 2018.*

Vector Representation	**Miss**	**FA**	**Speaker Error**	**Overall**
Supervector + LDA	1.8	2.5	26.0	30.31
SC Embedding	1.5	2.5	26.7	30.68
Combined	**1.5**	**2.5**	**23.7**	**27.69**

Table 2 compares the DER results for three proposed speaker embeddings obtained on dev set of Albayzin 2018 dataset. Both LDA transformed supervectors and SC embeddings give similar overall DER but SC embeddings show more balanced performance over two different channels. The combination of both improves the DER for both channels.

Table 3 and Fig. 4 show the same experiments as in Table 2 on the evaluation set, yielding the same conclusions. We participated in the challenge with only LDA transformed supervectors and the same low cost online diarization algorithm [27] and obtained very competitive or even better results compared to the other more expensive offline diarization techniques.

4. Conclusions

We have proposed an efficient online speaker diarization algorithm in which the speech segments are represented by the proposed speaker vectors referred to as speaker-corrupted (SC) embeddings. Speaker corruption is made by adding low quality supervectors, built from only 20 speech frames from other speakers, to the supervectors of a given speaker. The embedding network takes speaker-corrupted supervectors as inputs, uses VReLU as an activation function, and tries to discriminate the background speakers. It is shown that the proposed algorithm achieves competitive or even better DER results compared to other offline algorithms with much lower computational cost, $0.01 \times RT$ with a single-core CPU machine.

5. Acknowledgements

This work uses data from RTVE 2018 database. This set of data has been provided by RTVE Corporation to help develop Spanish-language speech technologies.

6. References

[1] S. E. Tranter and D. A. Reynolds, "An overview of automatic speaker diarization systems," *IEEE Transactions on audio, speech, and language processing*, vol. 14, no. 5, pp. 1557–1565, 2006.

[2] S. Madikeri, I. Himawan, P. Motlicek, and M. Ferras, "Integrating online i-vector extractor with information bottleneck based speaker diarization system," in *Proc. Interspeech*, 2015.

[3] I. Viñals, A. Ortega, J. Villalba, A. Miguel, and E. Lleida, "Domain adaptation of plda models in broadcast diarization by means of unsupervised speaker clustering," in *Proc. Interspeech*, 2017, pp. 2829–2833.

[4] S. H. Yella and A. Stolcke, "A comparison of neural network feature transforms for speaker diarization," in *Proc. Interspeech*, 2015.

[5] Z. Zajıc, M. Hrúz, and L. Müller, "Speaker diarization using convolutional neural network for statistics accumulation refinement," *Proc. Interspeech*, 2017.

[6] D. Garcia-Romero, D. Snyder, G. Sell, D. Povey, and A. McCree, "Speaker diarization using deep neural network embeddings," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 4930–4934.

[7] K. Markov and S. Nakamura, "Never-ending learning system for on-line speaker diarization," in *Proc. ASRU*. IEEE, 2007, pp. 699–704.

[8] K. Markov and S. Nakamura, "Improved novelty detection for online gmm based speaker diarization," in *Proc. Interspeech*, 2008.

[9] J. Geiger, F. Wallhoff, and G. Rigoll, "GMM-UBM based open-set online speaker diarization," in *Proc. Interspeech*, 2010.

[10] C. Vaquero, O. Vinyals, and G. Friedland, "A hybrid approach to online speaker diarization," in *Proc. Interspeech*, 2010.

[11] C. Breslin, K. Chin, M. J. Gales, and K. Knill, "Integrated online speaker clustering and adaptation," in *Proc. Interspeech*, 2011.

[12] G. Soldi, C. Beaugeant, and N. Evans, "Adaptive and online speaker diarization for meeting data," in *Proc. EUSIPCO*. IEEE, 2015, pp. 2112–2116.

[13] W. Zhu and J. Pelecanos, "Online speaker diarization using adapted i-vector transforms," in *Proc. ICASSP*. IEEE, 2016, pp. 5045–5049.

[14] G. Soldi, M. Todisco, H. Delgado, C. Beaugeant, and N. Evans, "Semi-supervised on-line speaker diarization for meeting data with incremental maximum a-posteriori adaptation," *Proc. Odyssey*, pp. 377–384, 2016.

[15] D. Dimitriadis and P. Fousek, "Developing on-line speaker diarization system," in *Proc. Interspeech*, 2017, pp. 2739–2743.

[16] G. Wisniewksi, H. Bredin, G. Gelly, and C. Barras, "Combining speaker turn embedding and incremental structure prediction for low-latency speaker diarization," *Proc. Interspeech*, pp. 3582–3586, 2017.

[17] J. Patino, R. Yin, H. Delgado, H. Bredin, A. Komaty, G. Wisniewski, C. Barras, N. Evans, and S. Marcel, "Low-latency speaker spotting with online diarization and detection," in *Proc. Odyssey*, vol. 2018, 2018, pp. 140–146.

[18] 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.

[19] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 1695–1699.

[20] O. Ghahabi and J. Hernando, "Restricted boltzmann machines for vector representation of speech in speaker recognition," *Computer Speech and Language*, vol. 47, pp. 16–29, 2018.

[21] A. Ortega, I. Vinals, A. Miguel, E. Lleida, V. Bazan, C. Perez, M. Zotano, and A. Prada, "Albayzin evaluation: IberSpeech-RTVE 2018 speaker diarization challenge," 2018, [Online]. Available: http://catedrartve.unizar.es/reto2018/EvalPlan-SpeakerDiarization-v1.3.pdf.

[22] E. Lleida, A. Ortega, A. Miguel, V. Bazan, C. Perez, M. Zotano, and A. Prada, "RTVE2018 database description," 2018, [Online]. Available: http://catedrartve.unizar.es/reto2018/RTVE2018DB.pdf.

[23] O. Ghahabi, W. Zhou, and V. Fischer, "A robust voice activity detection for real-time automatic speech recognition," in *Proc. ESSV*, 2018.

[24] N. Dehak, P. 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.

[25] E. Variani, X. Lei, E. McDermott, I. Lopez Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Proc. ICASSP*, May 2014, pp. 4052–4056.

[26] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," 2018.

[27] O. Ghahabi and V. Fischer, "EML submission to Albayzin 2018 speaker diarization challenge," *Proc. IberSPEECH 2018*, pp. 216–219, 2018.