applied sciences

MDPI

Article

Restricted Boltzmann Machine Vectors for Speaker Clustering and Tracking Tasks in TV Broadcast Shows †

Umair Khan *, Pooyan Safari and Javier Hernando

TALP Research Center, Department of Signal Theory and Communications, Universitat Politecnica de Catalunya—BarcelonaTech, 08034 Barcelona, Spain

* Correspondence: umair.khan@upc.edu; Tel.: +34-640-770-111

† This paper is an extended version of our paper published in IberSPEECH-2018.

Received: 21 May 2019; Accepted: 2 July 2019; Published: 9 July 2019

Abstract: Restricted Boltzmann Machines (RBMs) have shown success in both the front-end and backend of speaker verification systems. In this paper, we propose applying RBMs to the front-end for the tasks of speaker clustering and speaker tracking in TV broadcast shows. RBMs are trained to transform utterances into a vector based representation. Because of the lack of data for a test speaker, we propose RBM adaptation to a global model. First, the global model—which is referred to as universal RBM—is trained with all the available background data. Then an adapted RBM model is trained with the data of each test speaker. The visible to hidden weight matrices of the adapted models are concatenated along with the bias vectors and are whitened to generate the vector representation of speakers. These vectors, referred to as RBM vectors, were shown to preserve speaker-specific information and are used in the tasks of speaker clustering and speaker tracking. The evaluation was performed on the audio recordings of Catalan TV Broadcast shows. The experimental results show that our proposed speaker clustering system gained up to 12% relative improvement, in terms of Equal Impurity (EI), over the baseline system. On the other hand, in the task of speaker tracking, our system has a relative improvement of 11% and 7% compared to the baseline system using cosine and Probabilistic Linear Discriminant Analysis (PLDA) scoring, respectively.

Keywords: speaker tracking; speaker clustering; speaker segmentation; restricted boltzmann machine adaptation; agglomerative hierarchical clustering

1. Introduction

Deep learning has been successfully applied to various tasks of image and speech technologies in recent decades. Their success has influenced the research community to make use of these techniques in speaker recognition tasks [1–5]. Deep learning has been applied to extracting bottle neck features (BNF) and then compute Gaussian Mixture Models (GMM) posterior probabilities in a hybrid Deep Neural Network–Hidden Markov Model (DNN-HMM) model [6,7]. At the front end, deep learning is capable of learning deep features from acoustic features, which are used in several speaker recognition tasks [5,8–11]. Deep learning has also been applied to learning a vector representation of a speaker for speaker verification, such as in References [5,12–14]. There are some interesting works that address the performance loss on degraded speech condition and acoustic mismatch between enrollment and test phases of speaker recognition systems [15,16]. Also, there are several recent approaches to obtaining fast training, for example, the Extreme Learning Machine (ELM), which has been extremely efficient in representational learning and several other learning tasks [17–19].

Unsupervised deep learning architectures like Restricted Boltzmann Machines (RBMs), Deep Belief Networks (DBNs) and Deep Autoencoders have the ability of representational learning power.

A first attempt to use RBMs at the backend in a speaker verification task was made in Reference [20]. The authors put their efforts into the front end of a speaker verification system, in order to learn a compact and fixed dimensional speaker representation in the form of a speaker vector by means of RBM adaptation [21–24]. They also make use of DBNs in the i-vector/PLDA (Probabilistic Linear Discriminant Analysis) framework for speaker verification at the backend [25]. The vector representation of speakers in Reference [22], was referred to as an RBM vector. It has been shown that the RBM vectors can extract speaker specific information that can be competitive as compared to i-vector based speaker verification. This has led us to apply this kind of vector representation of speakers to the tasks of speaker clustering and speaker tracking.

Speaker clustering refers to the task of grouping speech segments in order to have segments from the same speaker in the same group. Ideally each group or cluster must contain speech segments that belong to the same speaker. On the other hand, utterances from the same speakers must not be distributed among multiple clusters. Several approaches to speaker clustering tasks exist, for example, cost optimization, sequential and Agglomerative Hierarchical Clustering (AHC) [26–29]. Some approaches rely on commonly used statistical speaker modeling like Gaussian Mixture Models (GMMs), while others use features extracted using Deep Neural Networks (DNNs). For example, in Reference [9], BN features extracted from different DNNs are used for speaker clustering using an AHC approach.

In certain applications, a person of interest (target) is tracked in an audio file, by using his voice characteristics. To identify *when the target speaker speaks* in the audio, is a speaker tracking task [30]. The main stages in speaker tracking are to determine the positions in the audio where the speaker changes occur, that is, speaker segmentation, and to then identify the speaker, that is, speaker identification. Based on these two stages, there can be a joint or a separate approach to a speaker tracking task. In the past, several approaches to speaker tracking were proposed based on different segmentation and speaker modeling strategies. In order to detect speaker changes, a fixed length window is slid over the audio and a distance metric is computed between consecutive windows. This distance is set to a threshold to decide whether there exists a speaker change. In Reference [31], speaker change detection is performed using a Generalized Likelihood Ratio [32,33]. In Reference [34] the Divergence Shape distance, described in Reference [35,36], is computed for speaker change detection. After this, the audio is segmented using the speaker change detection. The conventional GMMs are trained to represent the speakers and a speaker identification is performed in order to decide which segment belongs to which speaker among the set of given target speakers.

In this paper, we propose the use of RBM vectors [22] for the above mentioned tasks, that is, speaker clustering and speaker tracking. The RBM vector is extracted in several steps. First of all, a global or Universal RBM—referred to as URBM—is trained with all the available background data. Then, an adapted RBM model per test speaker is trained. In the case of speaker clustering, the test speakers are the segments that are to be clustered. In the case of speaker tracking, the test speakers are audio segments and target speakers. The visible to hidden weight matrices along with the visible and hidden bias vectors of these adapted RBMs are concatenated to generate RBM supervectors. The RBM supervectors are subjected to a Principal Component Analysis (PCA) whitening and dimensionality reduction to extract the desired RBM vectors.

For the speaker clustering task, we extract RBM vectors using the method described above for the test speakers. In this way, all the speaker segments that are to be clustered are represented by RBM vectors. Then we cluster these RBM vectors by applying a bottom-up AHC approach using cosine and Probabilistic Linear Discriminant Analysis (PLDA) scores. In Reference [24], we have concluded that the RBM vector representation of speakers is successful in the task of speaker clustering. In this paper, we investigate the same RBM vector representation of speakers in the task of speaker tracking.

For the speaker tracking task, we implement a two stage strategy. The first stage is based on speaker change detection by using Divergence Shape distance as in Reference [34]. The audio is segmented according to these speaker change points. In the second stage, the segments generated are identified against all the target speakers, in order to specify which segment belongs to which target speaker. The target speakers are first enrolled in the system. We represent all the segments and target speakers by RBM vectors. Then, the RBM vectors of all the segments are scored against the RBM vectors of all the target speakers using cosine and PLDA scoring. We have found that the RBM vector representation of speakers is successful in both these tasks as in speaker verification. The experimental results show that the RBM vector outperforms the conventional i-vectors based systems using both the cosine and PLDA scoring methods.

The rest of the paper is organized as follows: Section 2 explains the detailed procedure of the proposed vector representation of speakers by using RBMs; Section 3 contains a brief description of the speaker clustering system; Section 4 contains a detailed description of the fundamental stages of our speaker tracking system; Section 5 describes the experimental setup, the database used and how the experiments were carried out; the results obtained are discussed in Section 6; and finally, in Section 7, some conclusions are drawn as the findings of this paper.

2. RBM Vector Representation

In this paper, we propose the use of a compact, vector based representation of speakers using RBM adaptation for speaker tracking and speaker clustering tasks. Figure 1 shows a detailed block diagram of the proposed RBM vector extraction. First, a global model—referred to as Universal RBM (URBM)—is trained with a large amount of background data. The URBM is then adapted to the data of every test speaker and thus an RBM is trained per test speaker. The visible to hidden weight matrices of these adapted models are used to generate the desired vector representation for the corresponding speaker. These vector representations of speakers are further used in the above-mentioned tasks using cosine/PLDA scoring. The whole process of the vector representation of speakers has three main steps, namely URBM training, RBM adaptation and RBM vector extraction using PCA whitening with dimensionality reduction.

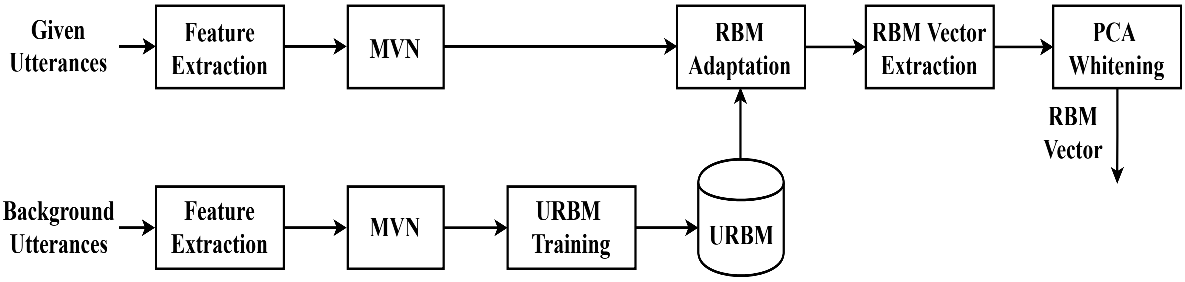


Figure 1. Block diagram showing different stages of the Resricted Botlzmann Machine (RBM) vector extraction

2.1. URBM Training

To extract the desired RBM vector, the first step is to train a global or universal model with a large amount of available background speakers' utterances. This global model is referred to as URBM, which is supposed to convey speaker-independent information. The URBM is trained as a single model with the features extracted from all the background speakers' data. For the real valued input features, we have used Gaussian real-valued units for the visible layer of the RBM [37]. The training is performed using the CD-1 algorithm [38,39] assuming that the inputs have zero mean and unit variance. Thus, the features are Mean Variance Normalized (MVN) before the RBM training. Finally, the universal model is trained with a large number of training samples generated from the feature vectors of the background speakers' utterances. This universal model is supposed to learn both speaker and session variabilities from the large background data [22].

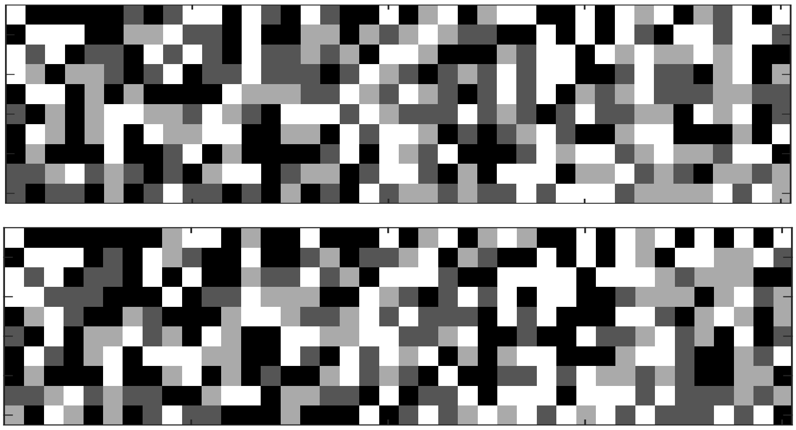
2.2. RBM Adaptation

After the URBM training, we perform speaker adaptation for every test speaker. The adapted RBM model is trained only with the data of the corresponding speaker, in order to capture speaker-specific information. In this step, the RBM model of the speaker segment is initialized with the parameters (weights and biases) of the URBM. In other words, the adaptation step drives the URBM model in a speaker-specific direction. This kind of adaptation technique is successfully applied in References [25,40–42]. The adaptation is also carried out by the CD-1 algorithm. As we have only one weight matrix in an RBM, all the information learned by an RBM is in the weight matrix and it is supposed to convey speaker-specific information of the corresponding speaker.

2.3. RBM Vector Extraction

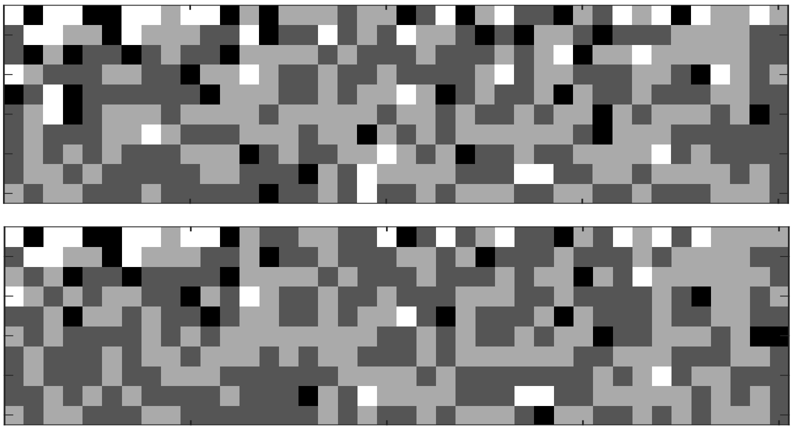
An RBM model is assigned to each test speaker after the adaptation step. The visible to hidden weight matrices along with their corresponding bias vectors of the adapted RBMs are concatenated in order to generate a higher dimensional speaker vector. These are referred to as RBM supervectors. After this, a PCA whitening with dimensionality reduction is applied to the RBM supervectors in order to generate the lower dimensional RBM vectors. The PCA whitening transforms the original data to the principal component space which de-correlates the data components. The PCA is trained with the RBM supervectors extracted from the background speakers' utterances and is applied to the RBM supervectors of the test speakers. All the RBM supervectors are mean-normalized before subjecting to PCA whitening and dimensionality reduction. The extracted RBM vectors are supposed to convey enough speaker-specific information, which can discriminate different speakers.

Figure 2 shows a visualization of a pair of RBM vectors (top and bottom) extracted from different utterances of two different speakers randomly selected from the test audios. From the Figure, it is clear that the two RBM vectors extracted for Speaker 1 look similar but are different from those extracted for Speaker 2. Similarly, the two RBM vectors extracted for Speaker 2 look similar but are different from those extracted for Speaker 1. In our previous work [22], it has been shown that the RBM vector extracted in this way is successful in learning speaker-specific information in a speaker verification task. Thus, we make an effort to make use of the RBM vector in the tasks of speaker clustering and speaker tracking.



(**a**) Speaker 1

Figure 2. *Cont.*



(**b**) Speaker 2

Figure 2. Examples of 400-dimensional RBM vectors. The figure shows two pairs of RBM vectors from the test audios. Each pair belong to the same speaker. We rearrange the RBM vectors in the form 10 × 40 for the convenience of visualization. The ordering of the RBM vector is the same for all.

3. Speaker Clustering

In order to evaluate the effect of RBM vectors in a speaker clustering task, we considered the conventional bottom-up AHC clustering system with the options of single and average linkages. We did not consider the model retraining approach because it is costly in terms of computations as compared to the linkage approaches to clustering [28]. The system starts with an initial number of clusters equal to the total number of speaker segments. Iteratively, the segments that are more likely to be from the same speaker are clustered together until a stopping criterion has reached. The stopping criterion can be thresholding the score in order to decide to merge clusters or it can be a desired (known) number of clusters achieved. The clustering algorithm is based on computing a distance/similarity matrix $M(X)$ between all the speakers' segments where X is the set of segments to be clustered. Hence, the RBM vectors of all the segments are extracted, the matrix $M(X)$ is computed by scoring all the RBM vectors against all. Thus, for N RBM vectors, the matrix $M(X)$ has dimensions $N \times N$. In every iteration, the segments with minimum/maximum distance/similarity scores are clustered together and the matrix $M(X)$ is updated. The corresponding rows and columns of the clustered segments are removed from $M(X)$ and a new row and column are added. The new row and column contain the distance scores between the new and old clusters. The new scores are computed according to the linkage algorithm used. For example, segments S_a and S_b are clustered in S_{ab}. Then the scores between new cluster (S_{ab}) and old segment (S_n) are computed as follows:

(a) Average Linkage:

$$s(S_{ab}, S_n) = \frac{1}{2}\{s(S_a, S_n) + s(S_b, S_n)\} \tag{1}$$

(b) Single Linkage:

$$s(S_{ab}, S_n) = max\{s(S_a, S_n), s(S_b, S_n)\} \tag{2}$$

where $s(S_{ab}, S_n)$ is the score between new cluster S_{ab} and old segment S_n while $s(S_a, S_n)$ is the score between old segments S_a and S_n.

In this way, the process is iterated until a stopping criterion is met. There are two methods to control the iterations: (1) to fix a threshold and (2) to add an additional information to the system about the desired (known) number of clusters. The system stops when this number is reached. In this work, we did not let the system know any desired number of clusters and we have used the thresholding method. We have tuned a threshold in order to see the performance of the system at different possible working points. The system performance is measured with respect to a ground truth cluster label.

4. Speaker Tracking

We extend our previous work in Reference [24] in order to investigate the effect of RBM vectors on a speaker tracking task. We implemented a two stage speaker tracking system, that is, speaker segmentation and speaker identification. Figure 3 shows the basic steps of speaker segmentation, RBM vector extraction and identification. First of all, the audio is segmented according to the speaker change points. The speaker change points are detected using *'the sliding window and searching for speaker change'* approach. A fixed length window is slid over the audio with a very small shift and speaker change is detected using some distance metric. We have used the Divergence Shape distance as a distance metric in this paper. The distance is thresholded in order to decide if the neighboring windows are spoken by the same speaker or whether there exists a speaker change. As a result of these speaker change points, the audio is segmented. In the next stage, a speaker identification of the target speakers against the segments is performed in order to know *'to which target speaker the corresponding segment belongs?'* All the target speakers and segments are transformed into a vector based representation by means of RBMs, that is, RBM vectors. These RBM vectors are scored using cosine and PLDA scoring methods. In the following sections, the two stages of our speaker tracking system are discussed in detail.

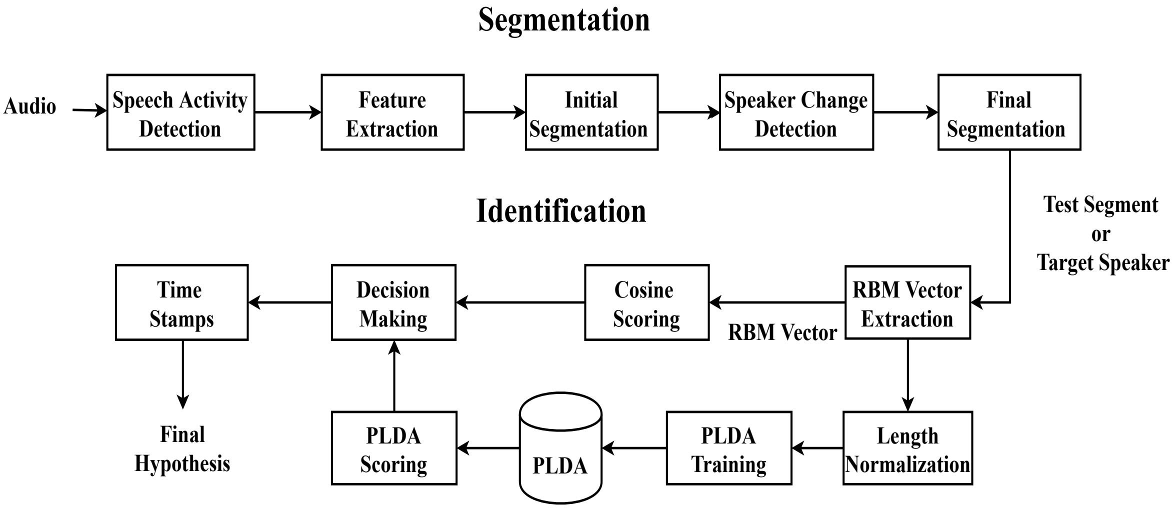


Figure 3. Architecture of Speaker Tracking using RBM vector Extraction.

4.1. Speaker Segmentation

As shown in Figure 3, first an energy-based Speech Activity Detection (SAD) is performed on the audio. Then, the speech parts are segmented into small segments of d seconds with an overlap of $(d - \Delta)$ seconds, where Δ is the shift. This is referred to as *initial segmentation* in the segmentation part of Figure 3. The segments generated in this step are reffered to as *small segments*. The shift Δ defines the resolution of speaker change detection. Then, Mel-Frequency Cepstral Coefficients (MFCC) features are extracted for every *small segment*. In order to detect speaker change points, the Divergence Shape distance is computed between every adjacent *small segments* and is thresholded. We compute the Divergence Shape distance as in References [34,36], using the following simplified expression:

$$D = \frac{1}{2} tr[(C_i - C_j)(C_j^{-1} - C_i^{-1})] \tag{3}$$

where tr is the trace function that sums the diagonal elements of a matrix, C_i is the covariance of the features from *small segment* S_i and C_j is the covariance of the features from *small segment* S_j. A speaker change point is marked if the distance at that point is greater than the distances at the two neighboring points (one before and one after) and a threshold at that point. For example, a speaker change point at *small segment* S_i occurs if:

$$D(i, i+1) > D(i, i+2) \tag{4}$$

and

$$D(i, i+1) > D(i-1, i) \tag{5}$$

and

$$D(i, i+1) > Threshold_i \tag{6}$$

where $D(i, i+1)$, $D(i, i+2)$ and $D(i-1, i)$ are the Divergence Shape distances of *small segment* S_i to S_{i+1}, S_i to S_{i+2} and S_{i-1} to S_i, respectively. $Threshold_i$ is an adaptive threshold which is computed for every *small segment* and is defined in [34] as:

$$Threshold_i = \frac{\alpha}{N} \sum_{n=0}^{N} D(i-n-1, i-n) \tag{7}$$

where α is a scaling factor and needs to be tuned experimentally. We have evaluated the segmentation with different values of α which we will discuss in Section 6. In Equation (7), N is the number of previous distances used for predicting the threshold. Once we detect the speaker change points by using this method, we segment the audio on these points. The segments generated will be used in the next step, that is, speaker identification. It is worth noting that we did not perform any refining algorithm for the speaker change points. Rather, we fixed the value of α so as to minimize the Miss Detection error in order not to miss a speaker change. This is because a False Alarm error can possibly be corrected in the speaker identification stage but a Miss Detection error cannot be corrected.

4.2. Speaker Identification

The second stage of our speaker tracking system performs a conventional speaker identification test on the segments and target speakers as shown in Figure 3. The goal is to answer *to which target speaker, the segments belong?* We propose the use of RBM vector representation for both the target speakers and segments generated in the segmentation stage. The MFCC features are extracted both for target speakers and segments. Then, RBM vectors are extracted and all the segments are tested against target speakers using cosine and PLDA scoring. Assume that $S_{Tm,Sn}$ represents the cosine/PLDA score for testing the target speaker T_m against the segment S_n. For a segment under test, first we select a potential candidate among all the target speakers. The target speaker with the maximum score is a potential candidate for the segment. Then, if the maximum score is greater than a threshold, the identity of that target speaker is assigned to that segment. Generally, the identity of the target T_m is assigned to the segment S_n according to:

$$Id_{S_n} = \arg\max_m (S_{Tm,Sn}) \quad \text{if} \quad max(S_{Tm,Sn}) > \lambda \tag{8}$$

where λ is a threshold to decide whether the segment under test does not belong to any of the target speakers. If the score is less than λ, the segment is not assigned to any of the target speakers. This is reflected as a Missed Speaker Time (MST) error for the target speaker which the segment actually belongs to. There are no speakers that should be rejected by the system because we consider all the speakers as possible target speakers. We have performed experiments with different values of λ in order to analyze the effect of the proposed RBM vectors at all possible working points.

5. Experimental Setup and Database

5.1. Database

The experiments are performed on the AGORA database, which contains audio recordings of 34 TV shows from Catalan broadcast TV3 [43] (in total 68 audios of approximately 38 min each). These audios contain segments from 871 adult Catalan and 157 adult Spanish speakers. For all the experiments in this work, we selected 38 audio files for testing and 30 audios are used as background data. The background data were used to train the Universal Background Model (UBM) and Total Variability (T) matrix for the baseline i-vector system. For the proposed system, the background data were used to train the URBM and PCA. We manually extracted 2631 speaker segments from the test audios, according to ground truth rich transcription. These segments were used in the speaker clustering experiments. In the testing audios, 414 different speakers appear which were used as target speakers for the tracking experiments. For an audio file, all the speakers are considered as possible target speakers. A priori knowledge is required to enroll the target speakers in the system. Thus, the target speakers are enrolled using i-vectors and RBM vector approaches for the baseline and proposed systems, respectively. The target speakers are enrolled with 30 s of utterances. These enrollment utterances of target speakers are manually selected from the corresponding audio file (in which they appear) according to the ground truth rich transcription. It is worth noting that each target speaker appears in at least one of the test segments.

5.2. Baseline and RBM Vector Setup

For all the experiments, 20 dimensional MFCC features were extracted, for both the baseline and proposed systems, using a Hamming window of 25 ms with 10 ms shift. A 512 component UBM was trained to extract i-vectors for the baseline system and the PLDA was trained with the background i-vectors. A more recent and competitive features could have been used, for example the BottleNeck Features (BNF). These features (either in the baseline or in the proposed approach) would require a huge amount of labeled background data (for example phonetic labels). On the other hand, MFCC features do not require labeled data for training our models. This is the strength of our proposed RBM vectors, which were trained in a completely unsupervised manner. The UBM training, i-vector extraction, i-vector testing and PLDA training were carried out using Alize, a free open source toolkit [44].

For the proposed system, more than 3000 speaker segments were extracted from the background audios according to the ground truth rich transcription. For each segment, the features of 4 neighboring frames were concatenated in order to generate 80-dimensional feature inputs to the RBMs. With a shift of one frame, we generated almost 10 million samples for the URBM training. All the RBMs used in this paper consisted of 80 visible and 400 hidden units. The URBM was trained for 200 epochs with a learning rate of 0.0005, weight decay of 0.0002 and a batch size of 100. All the adapted RBM models for the segments and target speakers were trained with 200 epochs with a learning rate of 0.005, weight decay of 0.000002 and a batch size of 64.

For the baseline i-vector system, the hyperparameters were set to the typical values that are commonly used in speaker recognition tasks. For the proposed RBM vector system, the set of hyperparameters, that is, the visible and hidden units in all the RBMs, the number of epochs and batch size for the URBM, and learning rate for the adapted RBM models were adopted from our previous work in Reference [22]. For the adapted RBM models, we used a higher value for the number of epochs and a slightly lower value for batch size because the segments were very short as compared to our previous work in Reference [22].

The PCA was trained with the background RBM supervectors and was applied to the background RBM supervectors and test RBM supervectors, as discussed in Section 2.3. Finally, fixed dimensional RBM vectors were extracted for the test speakers that were used in the speaker tracking and clustering experiments. Different dimensions for the RBM vectors were evaluated in the experiments which is discussed in Section 6.

5.3. Evaluation Metrics

The results of the speaker clustering system were evaluated in terms of Cluster Impurity (CI). CI measures the quality of a cluster, *to what extent a cluster contains segments from different speakers*. However, this metric has a trivial solution when there is only one segment per cluster. To deal with this, Speaker Impurity (SI) was measured at the same time. SI measures *to what extent a speaker is distributed among clusters*. There is always a trade-off between these two metrics [45]. CI and SI were plotted against each other in an Impurity Trade-off (IT) curve and an Equal Impurity (EI) point was marked as a working point.

We evaluated the results for speaker segmentation in terms of False Alarm Rate (FAR) and Miss Detection Rate (MDR), as discussed in Reference [46]. The overall speaker tracking system was evaluated in terms of False Alarm (FA) and Missed Speaker Time (MST). In this case, FA is the percentage of duration (in seconds) that is falsely accepted for a target speaker while MST is the percentage of duration (in seconds) that is falsely rejected for a target speaker.

6. Results

6.1. Speaker Clustering

Different lengths for RBM vectors, as well as for i-vectors, were evaluated using cosine scoring and the average linkage clustering algorithm. The results are shown in the second column of Table 1. From the Table, it can be observed that if the dimension is increased, the performance is improved, both in case of i-vectors and RBM vectors, in terms of Equal Impurity (EI). However, in the case of i-vectors, the best choice is 800 dimension. In case of RBM vectors, the 2000 dimensional RBM vectors perform better than the others. In this case, a relative improvement of 11% is achieved compared to 800 dimensional i-vectors. A further increase in the length of RBM vectors beyond 2000 degrades the performance in terms of EI.

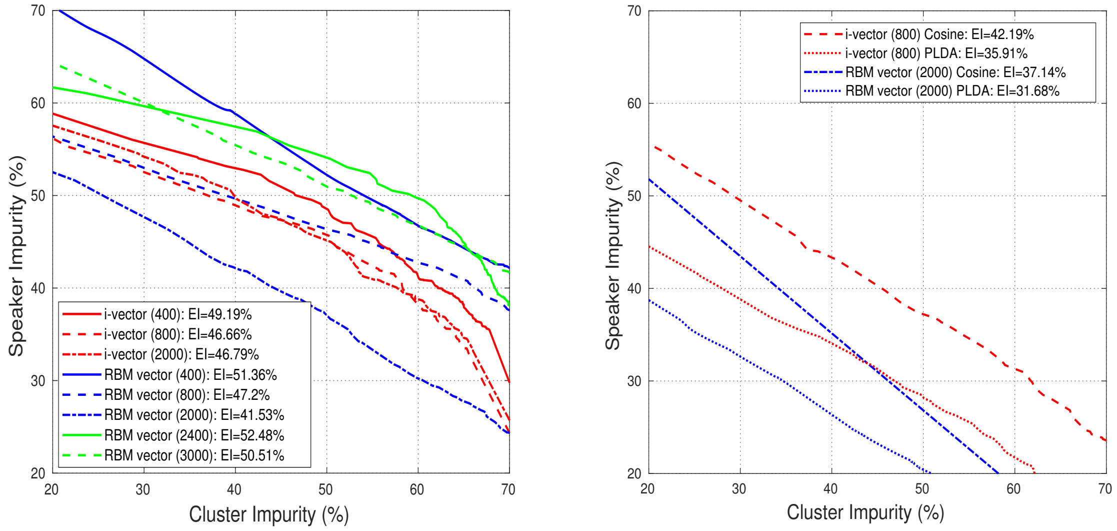
The third column of Table 1 compares the performance of the RBM vector with the baseline i-vectors in the case of the single linkage algorithm for clustering using cosine scoring. From the table it is seen that single linkage was a better choice for our experiments. In this case, a minimum EI of 37.14% is obtained with 2000 dimensional RBM vectors which has a relative improvement of 12% over 800 dimensional i-vectors.

Table 1. Comparison of speaker clustering results for the proposed RBM vectors with i-vectors, in terms of Equal Impurity (EI) in %. The dimensions of vectors are given in parenthesis. Each column shows EI in % for different scoring and linkage combinations.

Approach	EI% (Cosine Average)	EI% (Cosine Single)	EI% (PLDA Single)
i-vector (400)	49.19	46.26	36.16
i-vector (800)	46.66	42.19	35.91
i-vector (2000)	46.79	42.83	35.89
RBM vector (400)	51.36	39.66	37.36
RBM vector (800)	47.20	40.02	32.36
RBM vector (2000)	41.53	37.14	31.68

Finally, we evaluated the proposed system using PLDA scoring as well. The PLDA was trained using background RBM vectors for 15 iterations. The number of eigenvoices were set to 250, 450 and 500 for RBM vectors of dimensions 400, 800 and 2000, respectively. All the RBM vectors were subjected to length normalization prior to PLDA training. As per the previous results, we performed this experiment with the single linkage algorithm only. The results were compared with i-vectors in the fourth column of Table 1. It was observed that 800 and 2000 dimensional RBM vectors have a better EI compared to the respective similar dimensional i-vectors. In this case, the RBM vectors of dimension 2000 have a minimum EI of 31.68% which results in a relative improvement of 11% over the 800 dimensional i-vectors. However, in the case of 400 dimensions, the i-vectors outperform RBM vectors.

The Impurity Trade-off (IT) curves for the baseline, as well as the proposed system, are shown in Figure 4. Figure 4a shows the evaluation of different dimensions of i-vectors and RBM vectors in the average linkage clustering using cosine scoring. It can be seen that RBM vectors of length 2000 gives a better performance than 800 dimensional i-vectors at all working points. On the other hand, RBM vectors of dimensions 400, 800, 2400 and 3000 perform worse than i-vectors. It is observed that 400 and 800 dimensional RBM vectors could not capture enough information about the speaker while 2400 and 3000 dimensional RBM vectors include unnecessary information which degrades the performance.



(**a**) Length selection using cosine scoring with average linkage algorithm for clustering.

(**b**) Best length using cosine and PLDA scoring with single linkage algorithm for clustering.

Figure 4. Comparison of Impurity Trade-off (IT) curves for the proposed RBM vectors with i-vectors. Different dimensions of RBM vectors are evaluated using different scoring and linkage algorithms for clustering. The dimensions of i-vectors and RBM vectors are given in parenthesis.

In Figure 4b we show a comparison of 2000 dimensional RBM vectors with 800 dimensional i-vectors using both cosine and PLDA scoring with the single linkage algorithm for clustering. The choices of dimensions are based on the previous experiments as 2000 dimensional RBM vectors and 800 dimensional i-vectors give the best results with cosine scoring and average linkage. From Figure 4b, it can be seen that the RBM vectors perform better at all working points as compared to i-vectors using their respective cosine and PLDA scoring. However, at low Speaker Impurity regions, the RBM vector with cosine scoring outperforms the baseline i-vector with PLDA scoring. Overall, the 2000 dimensional RBM vector has a consistent improved performance compared to i-vectors.

6.2. *Speaker Tracking*

The application of RBM vectors was further extended to a speaker tracking task. For speaker change detection and segmentation, 20 MFCC features were extracted for all the *small segments* using a Hamming window of 25 ms with 10 ms shift. We performed segmentation using different sizes of small segments, that is, the d parameter discussed in Section 4.1 was equal to 2, 2.5 and 3 s. The value of Δ was set to 0.25 s. The speech parts smaller than d were not considered in these experiments and were simply discarded. Figure 5 shows the graph of FAR against MDR for different values of d and α. The results were computed, accepting a tolerance (collar) of ± 0.25 s in the position of detected speaker change points. We experimented with different values of d in order to see the behaviour at different working points, that is, d = 2, 2.5 and 3 s. Then, we experimented with different values of α and the results are plotted in Figure 5. From the Figure it is clear that the best choice for d is a 3 s window.

The MDR for this window is not very sensitive to alpha as compared to the other window sizes. This is because in our experiments, the segments less than the selected window size were discarded. Thus the segments have longer durations, which have strong boundaries with the neighbouring segments as compared to a window size of 2 and 2.5 s. A strong boundary is not very likely to be missed by the system. That is why, when we vary α, the MDR does not vary a lot and thus the MDR seems to be insensitive. On the other hand, if the window size is small, the segments have weak boundaries with the neighbour segments and are relatively more likely to be missed by the system.

Our actual working point is marked as a black circle which is obtained for $\alpha = 2$ (in Equation (7)). We performed the *final segmentation* at this point which has less Miss Detection (MDR) as compared to False Alarm (FAR). At this point, a FAR of 10% and MDR of 7.8% are achieved. There is a trade-off between the two metrics (FAR and MDR). One can decrease one of the metrics at the cost of increasing the other.

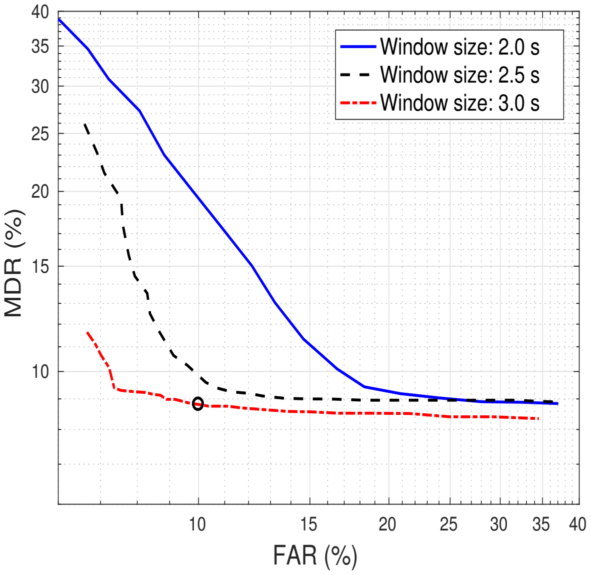


Figure 5. Speaker segmentation results in terms of False Alarm Rate (FAR) and Miss Detection Rate (MDR) in %. Results are obtained using different Window sizes (d) and a constant shift i.e., $\Delta = 0.25$ s. A collar of ±0.25 s is accepted around a speaker change point.

Table 2. Comparison of speaker tracking results for the proposed RBM vectors with 800 dimensional i-vectors, in terms of EER in %. The lengths of RBM vectors and i-vectors are given in parenthesis. Column 2 and 3 represents EER in % for cosine and PLDA Scoring respectively.

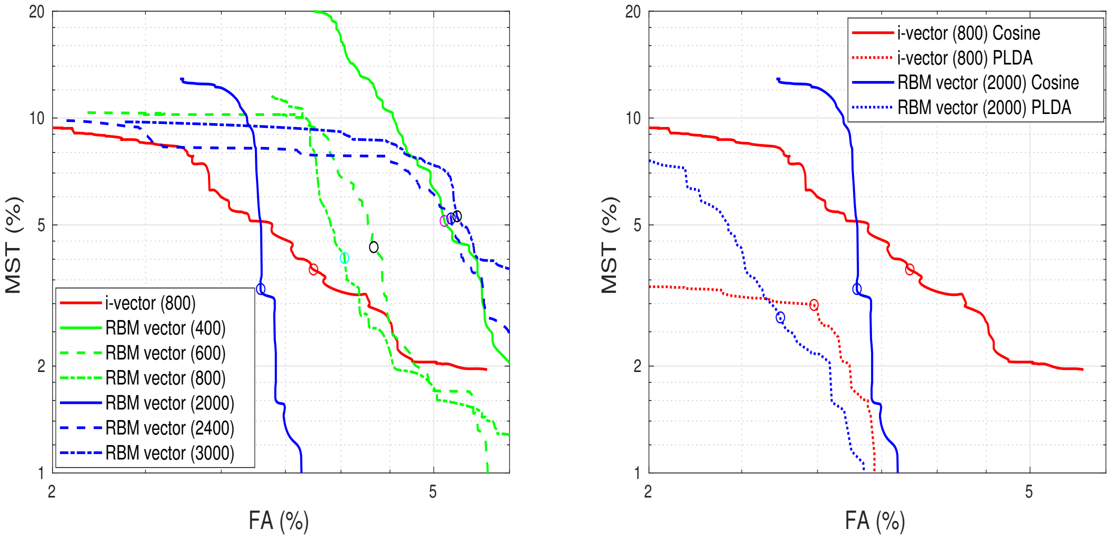
Approach	EER% (Cosine)	EER% (PLDA)
i-vector (800)	3.74	2.97
RBM vector (600)	4.33	3.57
RBM vector (800)	4.03	3.47
RBM vector (2000)	3.30	2.74

The segments generated in the speaker segmentation were then tested against the target speakers for the tracking task. Table 2 shows the results of speaker tracking for different lengths of RBM vector in terms of Equal Error Rate (EER). In this case EER was the coinciding point between FA and MST. The second column of Table 2 shows the comparison of RBM vector with the baseline i-vectors using cosine scoring. We fixed the length of i-vectors to 800 as a conclusion of the speaker clustering experiments. It is observed that, as the length of the RBM vector is increased, the performance is improved. The best EER of 3.30% was obtained using 2000 dimensional RBM vector, which gained a relative improvement of 11.76% as compared to the baseline 800 dimensional i-vectors. Increasing the dimensions of the RBM vectors does not affect the computational costs of training the models. The dimensions of RBM vectors are only controlled by the number of components while applying PCA to the RBM supervectors, as discussed in Section 2.3

The third column of Table 2 shows the comparison of the RBM vector with the baseline i-vectors using the PLDA scoring method. For the RBM vector/PLDA framework, the PLDA is trained using the background RBM vectors for 15 iterations. The number of eigenvoices are set to 350, 450 and

500 for RBM vectors of lengths 600, 800 and 2000 respectively. All the RBM vectors are subjected to length normalization prior to PLDA training. From the table, it is clear that the 2000 dimensional RBM vector/PLDA system outperforms the 800 dimensional i-vector/PLDA system by a relative improvement of 7.74%. In the case of PLDA post processing, increasing the dimensions of the RBM vectors will increase the computational costs of PLDA training. This is because the PLDA model is trained on higher dimensional background RBM vectors.

Figure 6 shows the comparison of Detection Error Trade-off (DET) curves for the baseline as well as the proposed system. These graphs are obtained by tuning the λ parameter in Equation (8). In Figure 6a we have evaluated different lengths of RBM vectors by comparing with i-vectors using cosine scoring. It can be observed that RBM vector of lengths 800 and 2000 give a better performance than the baseline i-vectors at low MST points only. An RBM vector of length 600 can be comparable with baseline i-vectors in this region. On the other hand, at low FA points the baseline i-vectors outperform RBM vectors of either length. However, at very few working points in low FA region, the RBM vector of length 2400 can be comparable with the baseline i-vectors.



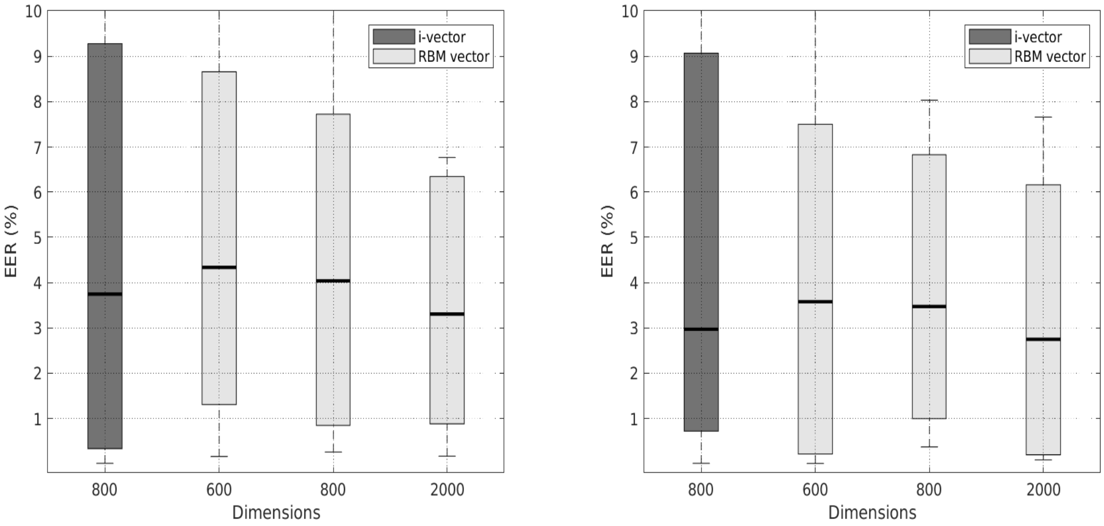
(**a**) Length selection using cosine scoring (**b**) Best length using cosine/PLDA scoring

Figure 6. Comparison of Detection Error Trade-off (DET) curves for the proposed RBM vectors with 800 dimensional i-vectors. Different lengths of RBM vectors are evaluated using cosine and PLDA scoring. The lengths of RBM vectors and i-vectors are given in parenthesis.

In Figure 6b we have shown a comparison of 2000 dimensional RBM vector (which gives the best results with the cosine scoring method) with baseline i-vectors using both cosine and PLDA scoring. From the figure, a similar kind of behavior is observed for RBM vectors using PLDA scoring as well. It can be seen that the 2000 dimensional RBM vector outperforms the baseline i-vectors in low MST regions using both cosine and PLDA scoring. However, in the low FA regions, the i-vector/PLDA framework still performs better which was also the case using the cosine scoring method.

The plots in Figure 6 are not very smooth and seem insensitive to λ. This is because the segments are not necessarily of the same duration. As the error (FA and MST) depends on the duration of segments, a false acceptance/rejection does not affect the error in a linear manner. Sometimes a certain value of lambda will falsely accept/reject a long segment which will highly affect the error. While in the case of a short segment a false acceptance/rejection will have a minimum reflection in the error.

We show the error variations of our experiments in Figure 7. The box plots in Figure 7 show the EER distribution of 38 test shows for the proposed RBM vector and i-vector based speaker tracking systems. Each box plot shows the minimum, lower quartile, mean, upper quartile, and maximum EER scores.



(**a**) EER variation using cosine scoring

(**b**) EER variation using PLDA scoring

Figure 7. EER variation comparison for the proposed RBM vectors with 800 dimensional i-vectors. Different lengths of RBM vectors are evaluated.

Figure 7a,b depict the box plots for different lengths of RBM vectors and 800 dimensional i-vectors using cosine and PLDA scoring, respectively. Figure 7a shows that RBM vectors reduce the EER variations as compared to i-vectors. It is seen that when we increased the length of RBM vectors, the EER variation was reduced further. The lowest EER variation is observed for 2000 dimensional RBM vector. Similarly, Figure 7b shows the same behaviour in EER variation using PLDA scoring. The EER variation was reduced in a similar manner for 800 and 2000 dimensional RBM vectors. However the mean EER was lower for the 200 dimensional RBM vector. Overall, the respective mean values of EER were lower for PLDA scoring as compared to cosine scoring.

7. Conclusions

In this paper, we have proposed the use of Restricted Boltzmann Machine (RBM) vectors for the tasks of speaker tracking and speaker clustering in TV broadcast shows. RBM is applied for learning a fixed dimensional vector representation of a speaker which is referred to as an RBM vector. First, a Universal RBM model is trained with a large amount of available background data. Then an adapted RBM model is trained per test speaker. The visible to hidden weight matrices along with the bias vectors of these adapted models are concatenated to generate RBM supervectors. The RBM supervectors are further subjected to a PCA whitening with dimensionality reduction to extract the desired RBM vectors. These RBM vectors are used in the tasks of speaker clustering and speaker tracking. For speaker clustering experiments, two linkage algorithms for an AHC approach are explored with RBM vectors scored using cosine and PLDA. Using cosine scoring, the performance of the proposed system is better for both the linkage algorithms as compared to i-vector based clustering. Overall, the single linkage algorithm with 2000 dimensional RBM vectors is the best choice for our experiments, using both cosine and PLDA scoring. For speaker tracking experiments, we performed speaker segmentation followed by a speaker identification. We proposed the use of RBM vectors for the speaker identification stage. In general, the proposed system is more effective in low MST regions. The experimental results have shown that, in terms of EER, the proposed system outperforms the baseline i-vectors system using both cosine and PLDA scoring methods. We conclude that the RBM vectors can be successfully used as a speaker representation in speaker clustering and speaker tracking tasks.

Author Contributions: Conceptualization, U.K., P.S. and J.H.; Funding acquisition, J.H.; Investigation, U.K.; Methodology, U.K., P.S. and J.H.; Project administration, J.H.; Resources, J.H.; Software, U.K. and P.S.; Supervision, J.H.; Writing—original draft, U.K.; Writing— review & editing, P.S. and J.H..

Funding: This work has been developed in the framework of DeepVoice Project (TEC2015-69266-P), funded by Spanish Ministry.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RBM	Restricted Boltzmann Machines
EI	Equal Impurity
PLDA	Probabilistic Linear Discriminant Analysis
GMM	Gaussian Mixture Models
HMM	Hidden Markov Models
DNN	Deep Neural Networks
DBN	Deep Belief Networks
BNF	Bottle Neck Features
AHC	Agglomerative Hierarchical Clustering
PCA	Principal Component Analysis
URBM	Universal Restricted Boltzmann Machines
MVN	Mean Variance Normalized
CD	Contrastive Divergence
SAD	Speech Activity Detection
MFCC	Mel-Frequency Cepstral Coefficients
UBM	Universal Background Model
TV	Total Variability
CI	Cluster Impurity
SI	Speaker Impurity
IT	Impurity Trade-off
FA	False Alarm
MST	Missed Speaker time
FAR	False Alarm Rate
MDR	Miss Detection Rate
DET	Detection Error Trade-off

References

1. Lei, Y.; Scheffer, N.; Ferrer, L.; McLaren, M. A novel scheme for speaker recognition using a phonetically-aware deep neural network. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 1695–1699.
2. Richardson, F.; Reynolds, D.; Dehak, N. Deep neural network approaches to speaker and language recognition. *IEEE Signal Process. Lett.* **2015**, *22*, 1671–1675. [CrossRef]
3. Chen, K.; Salman, A. Learning speaker-specific characteristics with a deep neural architecture. *IEEE Trans. Neural Netw.* **2011**, *22*, 1744–1756. [CrossRef] [PubMed]
4. Kenny, P.; Gupta, V.; Stafylakis, T.; Ouellet, P.; Alam, J. Deep neural networks for extracting baum-welch statistics for speaker recognition. *Proc. Odyssey*, 2014, pp. 293–298. Available online: https://www.isca-speech.org/archive/odyssey_2014/pdfs/28.pdf (accessed on 8 July 2019).
5. Liu, Y.; Qian, Y.; Chen, N.; Fu, T.; Zhang, Y.; Yu, K. Deep feature for text-dependent speaker verification. *Speech Commun.* **2015**, *73*, 1–13. [CrossRef]

6. Deng, L.; Yu, D. Deep learning: Methods and applications. *Found. Trends Signal Process.* **2014**, *7*, 197–387. [CrossRef]
7. Yamada, T.; Wang, L.; Kai, A. Improvement of distant-talking speaker identification using bottleneck features of DNN. *Interspeech*, **2013**, 3661–3664. Available online: https://www.isca-speech.org/archive/archive_papers/interspeech_2013/i13_3661.pdf (accessed on 8 July 2019).
8. Lee, H.; Pham, P.; Largman, Y.; Ng, A.Y. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc., 2009; pp. 1096–1104. Available online: http://papers.nips.cc/paper/3674-unsupervised-feature-learning-for-audio-classification-using-convolutional-deep-belief-networks.pdf (accessed on 8 July 2019).
9. Jorrín, J.; García, P.; Buera, L. DNN Bottleneck Features for Speaker Clustering. *Proc. Interspeech* **2017**, 1024–1028. [CrossRef]
10. Jati, A.; Georgiou, P. Speaker2Vec: Unsupervised Learning and Adaptation of a Speaker Manifold using Deep Neural Networks with an Evaluation on Speaker Segmentation. *Proc. Interspeech* **2017**, 3567–3571. [CrossRef]
11. Anna, S.; Lukáš, B.; Jan, C. Alternative Approaches to Neural Network Based Speaker Verification. *Proc. Interspeech* **2017**, 1572–1575. [CrossRef]
12. Variani, E.; Lei, X.; McDermott, E.; Moreno, I.L.; Gonzalez-Dominguez, J. Deep neural networks for small footprint text-dependent speaker verification. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014; pp. 4052–4056. Available online: https://ieeexplore-ieee-org.recursos.biblioteca.upc.edu/stamp/stamp.jsp?tp=&arnumber=6854363 (accessed on 8 July 2019)
13. Isik, Y.Z.; Erdogan, H.; Sarikaya, R. S-vector: A discriminative representation derived from i-vector for speaker verification. In Proceedings of the IEEE 23rd European Signal Processing Conference (EUSIPCO), Nice, France, 31 August–4 September 2015; pp. 2097–2101.
14. Bhattacharya, G.; Alam, J.; Kenny, P. Deep Speaker Embeddings for Short-Duration Speaker Verification. *Proc. Interspeech* **2017**, 1517–1521. [CrossRef]
15. Gong, Y. Speech recognition in noisy environments: A survey. *Speech Commun.* **1995**, *16*, 261–291. [CrossRef]
16. Siniscalchi, S.M.; Salerno, V.M. Adaptation to new microphones using artificial neural networks with trainable activation functions. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *28*, 1959–1965. [CrossRef]
17. Huang, G.; Huang, G.B.; Song, S.; You, K. Trends in extreme learning machines: A review. *Neural Netw.* **2015**, *61*, 32–48. [CrossRef] [PubMed]
18. Khoo, S.; Man, Z.; Cao, Z. Automatic han chinese folk song classification using extreme learning machines. In *Australasian Joint Conference on Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 49–60.
19. Salerno, V.; Rabbeni, G. An extreme learning machine approach to effective energy disaggregation. *Electronics* **2018**, *7*, 235. [CrossRef]
20. Senoussaoui, M.; Dehak, N.; Kenny, P.; Dehak, R.; Dumouchel, P. First attempt of boltzmann machines for speaker verification. In Proceedings of the Odyssey 2012—The Speaker and Language Recognition Workshop, 2012. Available online: https://www.isca-speech.org/archive/odyssey_2012/papers/od12_117.pdf (accessed on 8 July 2019)
21. Ghahabi, O.; Hernando, J. Restricted Boltzmann machines for vector representation of speech in speaker recognition. *Comput. Speech Lang.* **2018**, *47*, 16–29. [CrossRef]
22. Safari, P.; Ghahabi, O.; Hernando, J. From features to speaker vectors by means of restricted Boltzmann machine adaptation. In Proceedings of the ODYSSEY 2016—The Speaker and Language Recognition Workshop, 2016; pp. 366–371. Available online: https://www.isca-speech.org/archive/Odyssey_2016/pdfs/15.pdf (accessed on 8 July 2019).
23. Ghahabi, O.; Hernando, J. Restricted Boltzmann machine supervectors for speaker recognition. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015; pp. 4804–4808. Available online: https://ieeexplore-ieee-org.recursos.biblioteca.upc.edu/stamp/stamp.jsp?tp=&arnumber=7178883 (accessed on 8 July 2019).
24. Khan, U.; Safari, P.; Hernando, J. Restricted Boltzmann Machine Vectors for Speaker Clustering. In Proceedings of the IberSPEECH 2018, Barcelona, Spain, 21–23 November 2018; pp. 10–14, doi:10.21437/IberSPEECH.2018-3. [CrossRef]

25. Ghahabi, O.; Hernando, J. Deep Learning Backend for Single and Multisession i-Vector Speaker Recognition. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2017**, *25*, 807–817. [CrossRef]
26. Sayoud, H.; Ouamour, S. Speaker clustering of stereo audio documents based on sequential gathering process. *J. Inf. Hiding Multimedia Signal Process.* **2010**, *4*, 344–360.
27. Siegler, M.A.; Jain, U.; Raj, B.; Stern, R.M. Automatic segmentation, classification and clustering of broadcast news audio. In Proceedings of the DARPA Speech Recognition Workshop, 1997; pp. 97–99. Available online: https://pdfs.semanticscholar.org/219c/382f29b734d0be0bbf0426aab825b328b3c1.pdf (accessed on 8 July 2019).
28. Ghaemmaghami, H.; Dean, D.; Sridharan, S.; van Leeuwen, D.A. A study of speaker clustering for speaker attribution in large telephone conversation datasets. *Comput. Speech Lang.* **2016**, *40*, 23–45. [CrossRef]
29. Tranter, S.E.; Reynolds, D.A. An overview of automatic speaker diarization systems. *IEEE Trans. Audio Speech Lang. Process.* **2006**, *14*, 1557–1565. [CrossRef]
30. Luque, J. Speaker Diarization and Tracking in Multiple-Sensor Environments. Ph.D. Thesis, Department of Signal Theory and Communications, Universitat Politècnica de Catalunya, Barcelona, Spain, 2012.
31. Bonastre, J.F.; Delacourt, P.; Fredouille, C.; Merlin, T.; Wellekens, C. A speaker tracking system based on speaker turn detection for NIST evaluation. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Istanbul, Turkey, 5–9 June 2000; Volume 2, pp. II1177–II1180.
32. Gish, H.; Siu, M.H.; Rohlicek, R. Segregation of speakers for speech recognition and speaker identification. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Toronto, ON, Canada, 14–17 May 1991; pp. 873–876.
33. Gish, H.; Schmidt, M. Text-independent speaker identification. *IEEE Signal Process. Mag.* **1994**, *11*, 18–32. [CrossRef]
34. Lu, L.; Zhang, H.J. Speaker change detection and tracking in real-time news broadcasting analysis. In Proceedings of the Tenth ACM International Conference on Multimedia, Miami, FL, USA, 4–6 January 2002; pp. 602–610.
35. Lu, L.; Jiang, H.; Zhang, H. A robust audio classification and segmentation method. In Proceedings of the Ninth ACM International Conference on Multimedia, Ottawa, ON, Canada, 30 September–5 October 2001; pp. 203–211.
36. Campbell, J.P. Speaker recognition: A tutorial. *Proc. IEEE* **1997**, *85*, 1437–1462. [CrossRef]
37. Hinton, G.E. A practical guide to training restricted Boltzmann machines. In *Neural Networks: Tricks of the Trade*; Springer: Berlin/Heidelberg, Germnay, 2012; pp. 599–619.
38. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507. [CrossRef]
39. Hinton, G.E.; Osindero, S.; Teh, Y.W. A fast learning algorithm for deep belief nets. *Neural Comput.* **2006**, *18*, 1527–1554. [CrossRef] [PubMed]
40. Safari, P.; Ghahabi, O.; Hernando, J. Feature classification by means of deep belief networks for speaker recognition. In Proceedings of the 23rd European Signal Processing Conference (EUSIPCO), Aalborg, Denmark, 23–27 August 2015; pp. 2117–2121.
41. Ghahabi, O.; Hernando, J. Deep belief networks for i-vector based speaker recognition. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Adelaide, Australia, 19–22 April 2014; pp. 1700–1704.
42. Ghahabi, O.; Hernando, J. I-vector modeling with deep belief networks for multi-session speaker recognition. *Network* **2014**, *20*, 13.
43. Schulz, H.; Fonollosa, J.A.R. A Catalan broadcast conversational speech database. In Proceedings of the Joint SIG-IL/Microsoft Workshop on Speech and Language Technologies for Iberian Languages, Sao Carlos, Brazil, 7–11 September 2009; pp. 27–30.
44. Larcher, A.; Bonastre, J.F.; Fauve, B.G.B.; Lee, K.A.; Lévy, C.; Li, H.; Mason, J.S.D.; Parfait, J.Y. ALIZE 3.0-open source toolkit for state-of-the-art speaker recognition. *Interspeech* **2013**, 2768–2772. Available online: https://www.isca-speech.org/archive/archive_papers/interspeech_2013/i13_2768.pdf (accessed on 8 July 2019).
45. Van Leeuwen, D.A. Speaker inking in large data sets. In *Proceedings of the Speaker and Language Recognition Odyssey*; 2010; pp. 202–208, Available online: https://www.isca-speech.org/archive_open/archive_papers/odyssey_2010/papers/od10_035.pdf (accessed on 8 July 2019).

46. Kotti, M.; Moschou, V.; Kotropoulos, C. Speaker segmentation and clustering. *Signal Process.* **2008**, *88*, 1091–1124. [CrossRef]