IberSPEECH 2018
21-23 November 2018, Barcelona, Spain

Restricted Boltzmann Machine Vectors for Speaker Clustering

Umair Khan, Pooyan Safari and Javier Hernando

TALP Research Center, Department of Signal Theory and Communications
Universitat Politecnica de Catalunya - BarcelonaTech, Spain
{umair.khan, javier.hernando}@upc.edu, pooyan.safari@tsc.upc.edu

Abstract

Restricted Boltzmann Machines (RBMs) have been used both in the front-end and backend of speaker verification systems. In this work, we apply RBMs as a front-end in the context of speaker clustering. Speakers' utterances are transformed into a vector representation by means of RBMs. These vectors, referred to as RBM vectors, have shown to preserve speaker-specific information and are used for the task of speaker clustering. In this work, we perform the traditional bottom-up Agglomerative Hierarchical Clustering (AHC). Using the RBM vector representation of speakers, the performance of speaker clustering is improved. The evaluation has been performed on the audio recordings of Catalan TV Broadcast shows. The experimental results show that our proposed system outperforms the baseline i-vectors system in terms of Equal Impurity (EI). Using cosine scoring, a relative improvement of 11% and 12% are achieved for average and single linkage clustering algorithms respectively. Using PLDA scoring, the RBM vectors achieve a relative improvement of 11% compared to i-vectors for the single linkage algorithm.

Index Terms: Speaker Clustering, Restricted Boltzmann Machine Adaptation, Agglomerative Hierarchical Clustering.

1. Introduction

In recent years, deep learning architectures have shown their success in various areas of image processing, computer vision, speech recognition, machine translation and natural language processing. This fact has inspired the research community to make use of these techniques in speaker recognition tasks [1, 2, 3, 4, 5]. In speaker recognition tasks, deep learning architectures are used to extract bottle neck (BN) features and to compute GMMs posterior probabilities in a hybrid HMM-DNN model [6, 7].

Generative or 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 [8]. Efforts have been done by the authors, in order to learn a compact and fixed dimensional speaker representation in the form of speaker vector by using RBMs as a front-end [9]. They also make use of DBNs at the backend in the i-vector framework for speaker verification [10]. As a continuation to these works, a successful attempt was made in our previous work to apply RBMs as a front-end for learning a fixed dimensional speaker representation [11]. This vector representation of speaker was referred to as RBM vector. In [11] it has been shown that the RBM vector preserves speaker specific information and has shown competitive results as compared to the conventional i-vector based speaker verification systems. This has lead us to apply the RBM vector for learning speaker representation in the task of speaker clustering.

Speaker clustering refers to the task of grouping speech segments in order to have segments from same speaker in the same group. Ideally each group or cluster must contains speech segments that belong to the same speaker. On the other hand, utterances from same speakers must not be distributed among multiple clusters. Several approaches to speaker clustering task exist, for example cost optimization, sequential and Agglomerative Hierarchical Clustering [12, 13, 14, 15]. 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 [16], BN features extracted from different DNNs are used for speaker clustering. In this work we consider the use of RBMs for vector representation of speakers. We extend the use of RBM vector [11] in the context of speaker clustering. First, we extract RBM vectors for all the speaker utterances in the same way as in [11]. Then, we perform a bottom-up AHC clustering for all the RBM vectors using cosine or Probabilistic Linear Discriminant Analysis (PLDA) scores. We have found that the RBM vector representation of speakers is successful in task of speaker clustering as in speaker verification. The experimental results show that the RBM vector outperforms the conventional i-vectors based speaker clustering using both the cosine and PLDA scoring methods.

The rest of the paper is organized as follows. Section 2 explains the training of a global model referred to as Universal RBM (URBM), RBM adaptation for speaker utterances and RBM vector extraction followed by a Principal Components Analysis (PCA) whitening and dimensionality reduction. Section 3 contains a brief description of the speaker clustering system using RBM vectors. Section 4 is about the experimental setup, database, evaluation metrics and the experiments carried out. In section 5 the obtained results are depicted and finally some conclusions are drawn in section 6.

2. RBM Vector Representation

In this work, we propose the use of a new speaker representation using RBMs in the context of speaker clustering. Fig. 1 shows the basic block diagram of different stages in the RBM vector extraction process and its input to the clustering module. First of all a global model which is referred to as Universal RBM (URBM), is trained using the features extracted from a large amount of background speakers. Then the URBM is adapted to the features extracted from each speaker's segments that are to be clustered. These models are referred to as adapted RBMs. The visible to hidden connection weights of these adapted models are used to generate the RBM vector for the corresponding

This work has been developed in the framework of the Camomile Project (PCIN-2013-067) and the Spanish Project DeepVoice (TEC2015-69266-P).

10.21437/IberSPEECH.2018-3

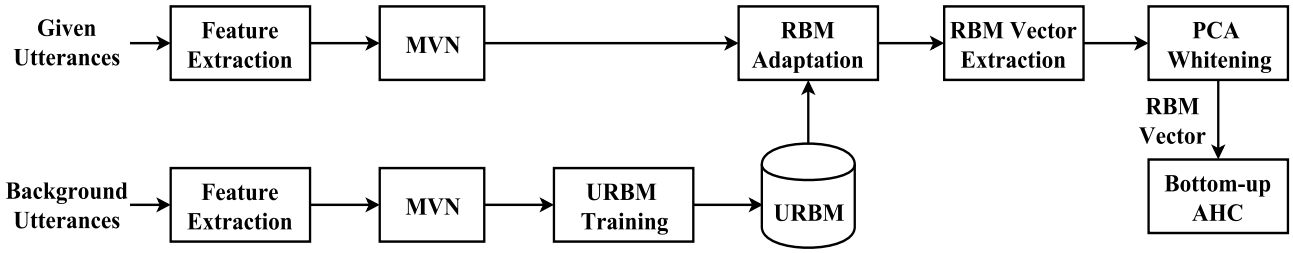

Figure 1: *Block diagram showing different stages of the RBM vector extraction and its input to Bottom-up AHC.*

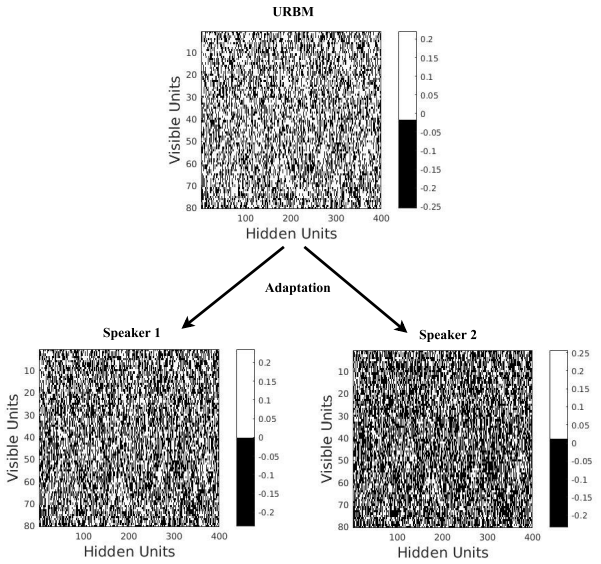

Figure 2: *Comparison of URBM and adapted RBMs visible-hidden weight matrices. The URBM weights are quite different from the adapted RBM ones which shows that these parameters convey speaker-specific information.*

speaker segment. Finally, the RBM vectors are further used for speaker clustering task using cosine and PLDA scoring. The whole process has three main stages namely URBM training, RBM adaptation and RBM vector extraction using PCA whitening with dimensionality reduction.

2.1. URBM Training

In order to generate the RBM vector, the first step is to train a global model with a large amount of background data. This is the 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. As, these features are real valued, we have used Gaussian real-valued units for the visible layer of the RBM. The RBM is trained using the CD-1 algorithm [17] which assumes that the inputs have zero mean and unit variance. Thus the features are Mean Variance Normalized (MVN) prior to the URBM training. Finally, we trained the URBM with a large amount of training samples generated from the background speakers' features. The URBM is supposed to learn both speaker and session variabilities from the background data [11].

2.2. RBM Adaptation

Once the URBM is trained, we perform speaker adaptation for every speaker's segment that has to be used for clustering task. The adapted RBM model is trained only with the data of the corresponding speaker's segment, in order to capture speaker-specific information. During adaptation the RBM model of the speaker segment is initialized with the parameters (weights and biases) of the URBM. This kind of adaptation technique is successfully applied in [18, 19]. The adaptation is also carried out using CD-1 algorithm. In other words, the adaptation step drives the URBM model in a speaker-specific direction. The weight matrix of the adapted models are supposed to convey speaker-specific information of the corresponding speaker. Fig. 2 shows the visualization of the connection weights of the URBM (at the top of the figure) and of two randomly selected speakers (Speaker 1 and Speaker 2, at the bottom of the figure). From the figure, it is clear that the URBM weights are driven in speaker-specific direction which can be discriminative.

2.3. RBM Vector Extraction

After the adaptation step, an RBM model is assigned to each speaker's segment. We concatenate the visible-hidden connection weights along with the bias vectors of the adapted speaker RBMs in order to generate a higher dimensional speaker vector, referred to as RBM supervector. As in our previous work [11], we apply a PCA whitening with dimensionality reduction to the RBM supervector in order to generate the lower dimensional RBM vector. The PCA whitening transforms the original data to the principal component space. This results in reducing the correlation between the data components. The PCA is trained with the background RBM supervectors and applied to the test RBM supervectors (used in clustering). All the RBM supervectors are mean-normalized prior to applying PCA. In our previous work [11], it has been shown that the RBM vector is successful in learning speaker-specific information in speaker verification task. Thus, we make use of the RBM vector in the context of speaker clustering.

3. Speaker Clustering

We have 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 [14]. The system starts with 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 contains 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:

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

(b) Single Linkage:

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

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) Fix a threshold, and (2) 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 labels. We will discuss evaluation metrics in section 4.

4. Experimental Setup and Database

The experiments were performed using the audios from AGORA database, which contains audio recordings of 34 TV shows of Catalan broadcast TV3 [20]. Each show comprises of two parts, i.e., *a* and *b*. So there are 68 audio files in total, of approximate length of 38 minutes each. These files contain segments from 871 adult Catalan and 157 adult Spanish speakers. For the clustering experiments in this work, we have selected 38 audio files for testing and the remaining 30 audios are used as a background data. The background data is used to train the Universal Background Model (UBM), Total Variability (T) matrix, URBM and PCA. From the testing audio files, we have manually extracted 2631 speaker segments according to ground truth rich transcription. These segments belong to 414 speakers that appears in the audios.

For both the baseline and proposed systems, 20 dimensional Mel-Frequency Cepstral Coefficients (MFCC) features are extracted using a Hamming window of 25 ms with 10 ms shift. For the baseline, a 512 components UBM is trained to extract i-vectors and the PLDA is trained with the background i-vectors, using Alize toolkit [21]. For the proposed system, more than 3000 speaker segments are extracted from the background shows according to the ground truth rich transcription. For each segment, we concatenate the features of 4 neighboring frames in order to generate 80-dimensional feature inputs to the RBMs. With a shift of one frame, we generate almost 10 million samples for the URBM training. The large amount of training samples will favor more efficient learning that will lead to more accurate URBM. All the RBMs used in this paper comprise 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 test speaker segments are trained with 200 epochs with a learning rate of 0.005, weight decay of 0.000002 and a batch size of 64. The PCA is trained with the background RBM supervectors as discussed in section 2.3. Finally, fixed dimensional RBM vectors are extracted for the speakers' segments and are used in the speaker clustering experiments. Different dimensions of the RBM vectors are evaluated which will be discussed in the results section.

There are several metrics to measure the performance of speaker clustering. For example cluster impurity (or conversely cluster purity), rand index, normalized mutual information (NMI) and F-measure as described in [22]. We have considered the Cluster Impurity (CI) measure in this work. 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) is measured at the same time. SI measures *to what extent a speaker is distributed among clusters*. There is a trade-off between CI and SI [23]. CI and SI are plotted against each other in an Impurity Trade-off (IT) curve and an Equal Impurity (EI) point is marked as working point.

5. Results

Different lengths for RBM vectors as well as for i-vectors are evaluated using cosine scoring and 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 case of i-vectors, the best choice is 800 dimension. In case of RBM vectors, the 2000 dimensional RBM vectors performs 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 RBM vector with the baseline i-vectors in case of single linkage algorithm for clustering using

Table 1: *Comparison of speaker clustering results for the proposed RBM vectors with i-vectors. The dimensions of vectors are given in parenthesis. Each column shows Equal Impurity (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**

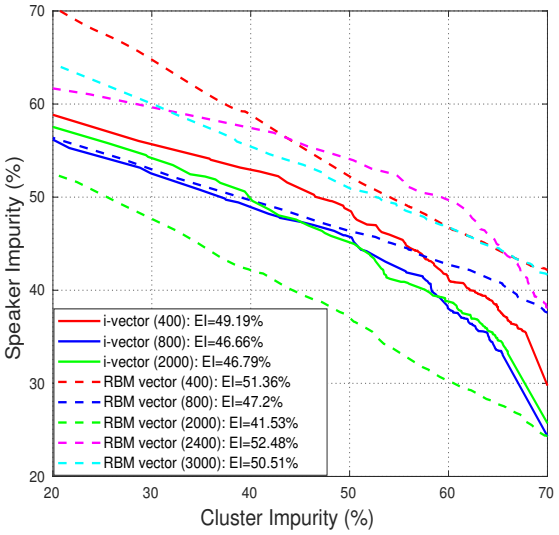

Figure 3: *Comparison of Impurity Trade-off (IT) curves for the proposed RBM vectors with i-vectors. Different dimensions of RBM vectors are evaluated using cosine scoring with average linkage algorithm for clustering. The dimensions of i-vectors and RBM vectors are given in parenthesis.*

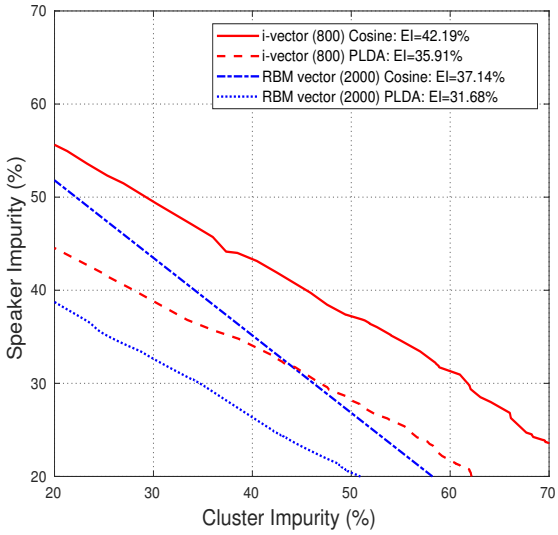

Figure 4: *Comparison of Impurity Trade-off (IT) curves for the proposed RBM vectors with i-vectors. Different dimensions of RBM vectors are evaluated using cosine and PLDA scoring with single linkage algorithm for clustering. The dimensions of i-vectors and RBM vectors are given in parenthesis.*

cosine scoring. From the table it is seen that, single linkage is 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. Finally, we evaluated the proposed system using PLDA scoring as well. The PLDA is trained using background RBM vectors for 15 iterations. The number of eigenvoices are set to 250, 450 and 500 for RBM vectors of dimensions 400, 800 and 2000 respectively. All the RBM vectors are subjected to length normalization prior to PLDA training. As per the previous results, we performed this experiment with single linkage algorithm only. The results are compared with i-vectors in the fourth column of Table 1. It is observed that 800 and 2000 dimensional RBM vectors has a better EI compared to the respective similar dimensional i-vectors. In this case, the RBM vectors of dimension 2000 has the minimum EI of 31.68% which results in a relative improvement of 11% over the 800 dimensional i-vectors. However, in 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 3 and 4. In Figure 3 we have shown 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 better performance than 800 dimensional i-vectors at all working points. On the other hand, RBM vectors of dimensions 400, 800, 2400 and 3000 performs 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. In Figure 4 we have shown a comparison of 2000 dimensional RBM vectors with 800 dimensional i-vectors using both cosine and PLDA scoring with 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 4, it can be seen that the RBM vectors performs 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. In overall, 2000 dimensional RBM vector has a consistent improved performance compared to i-vectors.

6. Conclusions

In this paper we proposed the use of Restricted Boltzmann Machines (RBMs) for speaker clustering task. RBMs are used to learn a fixed dimensional vector representation of speaker referred to as RBM vector. First, a Universal RBM is trained with a large amount of background data. Then an adapted RBM model per test speaker is trained. The visible-hidden weight matrices along with their bias vectors of these adapted RBMs are concatenated to generate RBM supervectors. These RBM supervectors are subjected to a PCA whitening and dimensionality reduction to extract RBM vectors. Two linkage algorithms for Agglomerative Hierarchical Clustering 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. In overall, single linkage algorithm with 2000 dimensional RBM vectors is the best choice for our experiments, using both cosine and PLDA scoring. We conclude that the RBM vectors can be successfully used as a speaker representation in a speaker clustering task. The best dimension for RBM vectors is found out to be 2000 which gives better performance over i-vectors as well as RBM vectors of other dimensions.

7. References

[1] F. Richardson, D. Reynolds, and N. Dehak, "Deep neural network approaches to speaker and language recognition," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1671–1675, 10 2015.

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

[3] P. Kenny, V. Gupta, T. Stafylakis, P. Ouellet, and J. Alam, "Deep neural networks for extracting baum-welch statistics for speaker recognition," in *Proc. Odyssey*, 2014, pp. 293–298.

[4] Y. Liu, Y. Qian, N. Chen, T. Fu, Y. Zhang, and K. Yu, "Deep feature for text-dependent speaker verification," *Speech Communication*, vol. 73, pp. 1–13, 10 2015.

[5] K. Chen and A. Salman, "Learning speaker-specific characteristics with a deep neural architecture," *IEEE Transactions on Neural Networks*, vol. 22, no. 11, pp. 1744–1756, 11 2011.

[6] L. Deng, D. Yu *et al.*, "Deep learning: methods and applications," *Foundations and Trends in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[7] T. Yamada, L. Wang, and A. Kai, "Improvement of distant-talking speaker identification using bottleneck features of DNN." in *Interspeech*, 2013, pp. 3661–3664.

[8] M. Senoussaoui, N. Dehak, P. Kenny, R. Dehak, and P. Dumouchel, "First attempt of boltzmann machines for speaker verification," in *Odyssey 2012-The Speaker and Language Recognition Workshop*, 2012.

[9] O. Ghahabi and J. Hernando, "Restricted boltzmann machines for vector representation of speech in speaker recognition," *Computer Speech & Language*, vol. 47, pp. 16–29, 1 2018.

[10] ——, "Deep learning backend for single and multisession i-vector speaker recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 4, pp. 807–817, 4 2017.

[11] P. Safari, O. Ghahabi, and J. Hernando, "From features to speaker vectors by means of restricted boltzmann machine adaptation," in *ODYSSEY 2016-The Speaker and Language Recognition Workshop*, 2016, pp. 366–371.

[12] H. Sayoud and S. Ouamour, "Speaker clustering of stereo audio documents based on sequential gathering process," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 4, pp. 344–360, 10 2010.

[13] M. A. Siegler, U. Jain, B. Raj, and R. M. Stern, "Automatic segmentation, classification and clustering of broadcast news audio," in *Proc. DARPA speech recognition workshop*, 1997, pp. 97–99.

[14] H. Ghaemmaghami, D. Dean, S. Sridharan, and D. A. van Leeuwen, "A study of speaker clustering for speaker attribution in large telephone conversation datasets," *Computer Speech & Language*, vol. 40, pp. 23–45, 11 2016.

[15] 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, 9 2006.

[16] J. Jorrín, P. García, and L. Buera, "DNN bottleneck features for speaker clustering," *Proc. Interspeech 2017*, pp. 1024–1028, 2017.

[17] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 6 2006.

[18] P. Safari, O. Ghahabi, and J. Hernando, "Feature classification by means of deep belief networks for speaker recognition," in *Signal Processing Conference (EUSIPCO), 2015 23rd European*. IEEE, 2015, pp. 2117–2121.

[19] O. Ghahabi and J. Hernando, "Deep belief networks for i-vector based speaker recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1700–1704.

[20] H. Schulz and J. A. R. Fonollosa, "A catalan broadcast conversational speech database," in *Joint SIG-IL/Microsoft Workshop on Speech and Language Technologies for Iberian Languages*, 2009, pp. 27–30.

[21] A. Larcher, J. F. Bonastre, B. G. B. Fauve, K. A. Lee, C. Lévy, H. Li, J. S. D. Mason, and J. Y. Parfait, "Alize 3.0-open source toolkit for state-of-the-art speaker recognition." in *Interspeech*, 2013, pp. 2768–2772.

[22] C. D. Manning, P. Raghavan, and H. Schütze, "Introduction to information retrieval," *Cambridge university press; 2008*, pp. 158–163.

[23] D. A. van Leeuwen, "Speaker linking in large data sets," *Proceedings of the Speaker and Language Recognition Odyssey*, pp. 202–208, 6 2010.