Speaker2Vec: Unsupervised Learning and Adaptation of a Speaker Manifold using Deep Neural Networks with an Evaluation on Speaker Segmentation

Arindam Jati[1], *Panayiotis Georgiou*[1]

[1]University of Southern California, Los Angeles, CA, USA

jati@usc.edu, georgiou@sipi.usc.edu

Abstract

This paper presents a novel approach, we term *Speaker2Vec*, to derive a speaker-characteristics manifold learned in an unsupervised manner. The proposed representation can be employed in different applications such as diarization, speaker identification or, as in our evaluation test case, speaker segmentation. Speaker2Vec exploits large amounts of unlabeled training data and the assumption of short-term active-speaker stationarity to derive a speaker embedding using Deep Neural Networks (DNN). We assume that temporally-near speech segments belong to the same speaker, and as such a joint representation connecting these nearby segments can encode their common information. Thus, this bottleneck representation will be capturing mainly speaker-specific information. Such training can take place in a completely unsupervised manner. For testing, our trained model generates the embeddings for the test audio, and applies a simple distance metric to detect speaker-change points. The paper also proposes a strategy for unsupervised adaptation of the DNN models to the application domain. The proposed method outperforms the state-of-the-art speaker segmentation algorithms and MFCC based baseline methods on four evaluation datasets, while it allows for further improvements by employing this embedding into supervised training methods.

Index Terms: unsupervised learning, deep neural networks, auto-encoder, manifold learning, speaker segmentation

1. Introduction

Learning speaker-specific characteristics is a very important requirement in many applications like speaker segmentation and clustering [1], speaker verification [2], and speaker recognition [3]. Speech features, like other high dimensional data, generally concentrate around a comparatively low dimensional representation or manifold [4], [5]. In recent years, many researchers have been working on finding these representations in speech to learn phoneme-level as well as speaker-specific characteristics [6], [7]. Speaker2Vec is designed to learn a speaker-characteristics manifold from unlabeled training data that even include multi-speaker audio streams. In this manifold, we expect speech samples of the same speaker to appear closer than they do in the raw feature space. Given any small segment (say 1 second) of speech, a trained Speaker2Vec model can find its latent "representation vector" or "embedding" which contains mostly speaker-specific information. In this paper, we have chosen speaker segmentation as a specific test case for evaluation, but we believe that the learned representations can also be used in many other applications.

Speaker segmentation (sometimes known as speaker change points detection) [1], [8] refers to the task of dividing an audio signal into multiple audio chunks such that each of them denotes a speaker homogeneous region. It has numerous applications in the fields of speaker diarization [9], [10], speaker tracking [11] and automatic speech recognition [12]. The state-of-the-art unsupervised speaker segmentation approaches are based on measuring the statistical distance between two consecutive windows in the audio signal. Some popular distance metrics are ΔBIC [13], KL or KL2 divergence [14], GLR [15] and Gaussian divergence (GD) [16]. These classical methods almost always use some low level acoustic features like MFCC [17] or PLP [18] for signal parameterization. In spite of the high distinctive capability of the distance metrics, sometimes MFCC or PLP fail to capture speaker characteristics especially in noisy scenarios or in audio streams containing similar sounding speakers.

Therefore, many researchers have investigated better ways of generating speaker-specific representations and consequently improving speaker segmentation. Deep neural networks [4] and deep auto-encoders [19] have been shown to perform quite satisfactorily for this task. The weights of the bottleneck or code layer [19] of an auto-encoder have been used as speaker embeddings [20] or representation vectors by many researchers for speaker segmentation, diarization and classification tasks. For example, [20] generated speaker embeddings using a DNN in supervised fashion by reducing the speaker misclassification error, and then employed those embeddings for speaker segmentation and clustering. A deep neural architecture consisting of two subnets was proposed in [7] for learning speaker characteristics using a loss function based on whether two input frames come from the same speaker or not. The model achieved promising performance for speaker segmentation task even when training and test sets were chosen from separate corpora. Yella and Stolcke [21] described multiple DNN architectures including auto-encoders to extract features for speaker diarization.

In spite of various novelties of the aforementioned methods, there are some limitations associated with them. Most of the methods need labeled training and validation data, which puts an upper bound to the amount of available training data in many scenarios. Consequently, these models might suffer from lack of robustness when the acoustic environment of the test audio is different from that of the training. Also, there is no way we can adapt these models to the test domain until we manually label some portion of the test audio.

The proposed Speaker2Vec framework tries to tackle these problems. We train a DNN on unlabeled data to learn a speaker-characteristics manifold, use the trained model to generate embeddings for the test audio, and use those embeddings to find the speaker change points. The main novelties of our work are the followings:

- The model doesn't require any labeled training data (no binary labeling of same/different speakers as used in [7] and [21], no multi-speaker labeling of different audio segments as used in [20]; and also no continuous labeling of speaker change points in the audio stream as used in [22]) for the learning phase.

- The proposed method is highly scalable. The DNN model can be expanded and trained with virtually unlimited amount of public domain audio data available from some sources like YouTube and LibriVox.
- We also propose a completely unsupervised domain adaptation technique which showed very promising results in our experiments.
- Any kind of voice activity detection (VAD) [23] information is not needed for training.

2. Methodology

We employ an auto-encoder to learn the speaker-characteristics manifold, but instead of reconstructing the input, we try to reconstruct a small window of speech from a temporally nearby window. Our hypothesis is that given the two windows are from the same speaker, the auto-encoder should be able to filter out all unnecessary features and capture the most common information between the two windows i.e. the speaker-characteristics and encode them in its embedding layer. This task remains simple for single-speaker audio streams, but seems to be difficult for unlabeled multi-speaker audio. We have the following motivation to move forward.

2.1. Unsupervised learning of speaker characteristics

2.1.1. Motivation: short-term speaker stationarity hypothesis

It is based on the simple notion that given a long audio stream containing multiple speakers, it is very unlikely that the speaker turns will occur very frequently (for example, every 1 second). So, if we consider pairs of two adjacent windows of small size, in most pairs, both windows will belong to a single speaker. There will be some pairs having windows from two different speakers (those actually contain the speaker change points), but the number of such pairs will probably be small compared to the total number of all pairs of adjacent windows. As we consider more and more data from various domains, the probability of getting more and more single-speaker window pairs increases. Now, we can take all the pairs to train an auto-encoder [4], [19] so that it learns the speaker manifold as described below.

2.1.2. Training framework

We use an auto-encoder having $(2K+1)$ hidden layers, where the $(K+1)^{th}$ layer is the embedding layer. Let's assume that we have extracted frames of MFCC feature vectors from an audio stream. We consider all pairs of adjacent windows $\mathbf{w_1}$ and $\mathbf{w_2}$, each of length d frames. Two consecutive pairs are separated by Δ frames as shown in Figure 1. Each pair becomes a training sample (input and output) for our auto-encoder. For every pair, $\mathbf{w_1}$ goes to the input layer of the auto-encoder, and $\mathbf{w_2}$ goes to the output. The auto-encoder tries to reconstruct $\mathbf{w_2}$ from $\mathbf{w_1}$ by minimizing a loss function

$$L(g(f(\mathbf{w_1})), \mathbf{w_2}) \tag{1}$$

where $f(.)$ is an encoder function which produces the embedding layer $\mathbf{h} = f(\mathbf{w_1})$, $g(.)$ is a decoder function which produces the output $\mathbf{o} = g(\mathbf{h})$, $L(.)$ is a scalar loss function such as mean squared error. This framework enables us to exploit longer context to capture speaker characteristics, and compress them into a lower dimensional vector representation. Different values of d and Δ employed in our experiments are reported in Section 3.3.

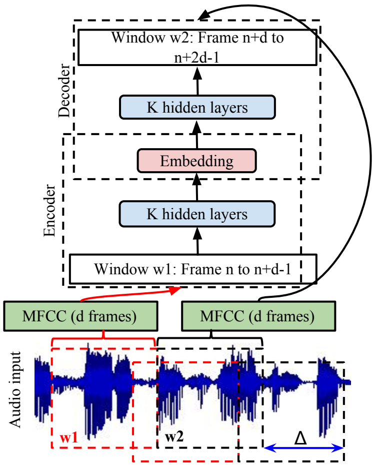

Figure 1: *Training framework and DNN architecture.*

2.2. Evaluation: Speaker segmentation

We use speaker segmentation as the task for evaluating Speaker2Vec. We just need the encoder part of the trained model for it. A window of d frames is traversed along the test audio stream with a shift of 1 frame. Every window is applied to the DNN model to generate an embedding $\mathbf{h}$ for that particular window. Now, these embeddings are used for segmentation instead of the original MFCC features. The segmentation algorithm (similar to the one used in [7]) can be summarized as follows.

1. Obtain a divergence curve by measuring the KL divergence between two adjacent windows and sliding them over the embeddings.
2. Normalize the divergence curve and apply a low pass filter to smooth it.
3. Detect the peaks of the divergence curve by using threshold T_{KL} (varies between 0 and 1).

We keep the segmentation algorithm very simple (single pass, no refinement, and asymmetric KL divergence [14]) to verify the distinctive ability of the speaker embeddings.

2.3. Transfer learning [24]: Unsupervised domain adaptation

We use the following two-pass algorithm for adapting a trained DNN model to the test data in a completely unsupervised way.

1. Find the speaker change points by the trained DNN model as described in Section 2.2.
2. Get all possible speaker homogeneous regions.
3. Retrain the same DNN again on these homogeneous segments of speech.

There might be some errors made by the model in step 1, but we only care about speaker homogeneous regions. So, we over-segment the test audio in the first step (by setting a particular value of T_{KL} as discussed in Section 3) so that we detect as many speaker change points as we can. We don't use segments smaller than $2d$ frames for adaptation.

3. Experiments

3.1. Training datasets

The first two columns of Table 1 describe the training datasets. We will refer to the trained Speaker2Vec models by the names of their training datasets. The TED-LIUM data is expected to fit well for training since each audio file contains speech mainly from a single speaker. To validate our hypothesis of short-term speaker stationarity more strongly, we have used a comparatively larger amount of various types of audio from YouTube. The reason for choosing YouTube is two-fold. First, we can get a virtually unlimited supply of data. Second, we can have speech samples from diverse conditions. For example, our YouTube datasets have varying acoustic environments including single-speaker monologues, multi-speaker discussions, movies, speech with background music, both in-studio and out-of-studio conversations etc. It also has audio from different languages including English, Spanish, Hindi and Telugu. We have used TED-LIUM development dataset [25] to stop the training before the model overfits to the training data (for all training sets).

3.2. DNN architectures

We have used two auto-encoder architectures by using $K = 2$ and 3 (Figure 1). The smaller network has been separately trained on TED-LIUM and YouTube datasets, while the bigger one has been trained on YouTubeLarge dataset. The number of neuron units in different layers from input to output are $[4000 \rightarrow 2000 \rightarrow 40 \rightarrow 2000 \rightarrow 4000]$ and $[4000 \rightarrow 6000 \rightarrow 2000 \rightarrow 40 \rightarrow 2000 \rightarrow 6000 \rightarrow 4000]$ for the smaller and bigger networks respectively. The length of the embedding layer has been set to 40, which makes it equivalent to the adopted MFCC dimension (discussed in Section 3.3). We have used ReLU activations for the hidden layers, and linear activations for the output layer. Logarithmic mean squared error has been used as error metric $L(.)$. The last two columns of Table 1 show number of hidden layers and total number of parameters for the encoder parts of the DNN models.

3.3. Experimental setting

We have adopted 40 dimensional high definition MFCC features extracted from 40 mel-spaced filters over a 25ms hamming window with a shift of 10ms using Kaldi toolkit [26]. We have used $d = 100$ frames (1s) for all training scenarios. This makes the size of input and output layers of the DNN models to be 4000. We have chosen $\Delta = 50$ frames (0.5s) for TED-LIUM, and $\Delta = 200$ frames (2s) for YouTube datasets (we could get enough training samples even with this increased Δ for YouTube datasets, and this completely removed overlap of samples). For segmentation, we have used two adjacent windows, each of size 1s, and slid them with a shift of 1 frame to achieve maximum resolution. For the first step of domain adaptation, T_{KL} has been experimentally set to 0.5. This threshold over-segments the test audio so that we get as many pure homogeneous segments as we can.

4. Results and discussions

4.1. Distinctive capability of the speaker embeddings over MFCC

Three datasets from very different acoustic conditions have been chosen to evaluate the performance of the proposed method. The first one has been artificially generated by taking 200 random utterances (extracted using the timestamps provided in the transcriptions) from TED-LIUM evaluation data. The second one is the NIST RT-06 conference meetings data (we have used the *hsum* audio files) [31]. The third set has been built by randomly picking 12 sessions (2.07 hours of audio) from the Couples Therapy Corpus [32], which has spontaneous conversations between husband and wife. The low SNR, poor performance of forced alignment systems [33], and our own requirements have inspired us to use this for evaluation.

Table 1: *Training datasets and corresponding DNN architectures employed.*

Dataset	Size (hours)	#Hidden layers (K+1)	#Parameters in DNN model
TED-LIUM	118	3	24M
YouTube	911	3	24M
YouTubeLarge	1953	4	52M

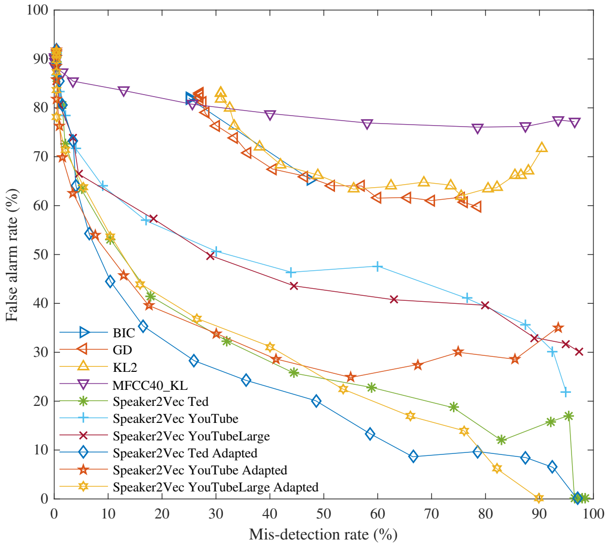

Figure 2: *ROC curves obtained by different algorithms on TED-LIUM evaluation data.*

Table 2 shows the results in terms of equal error rates (EER) as defined in [27]. Each tuple denotes mis-detection rate (MDR) and false alarm rate (FAR). Four baseline methods have been chosen. The first three use 13 dimensional MFCC features and apply BIC [13], Gaussian divergence (GD) [16] and KL2 [14] metrics respectively. The fourth algorithm uses the 40 dimensional MFCC features we use in our method, and directly applies the same KL divergence we have adopted. We have used forgiveness tolerance [8] of both $\pm 0.5s$ and $\pm 0.25s$ for RT-06 data to see the performance of our model as it decreases. For all other datasets, $\pm 0.5s$ is used similar to other works [8], [29].

We can see from Table 2 that the embeddings consistently perform much better than MFCC. The last row shows the absolute improvements in mean EER over the best baseline methods obtained by different Speaker2Vec models averaged across all test datasets. The TED-LIUM model, the best among unadapted models, achieves a 9.92% absolute improvement. The unsupervised adaptation on test data gives some boost in the performance. We obtain 11.72% and 11.14% absolute improvements

Table 2: *(Row 1-6) Comparison of the proposed method with some baseline methods (which use MFCC) on different datasets. EER is reported in (MDR, FAR) format. The best two EERs are in* ***boldface****. (Row 7) The last row shows the absolute improvements in mean EER (mean(MDR, FAR)) over the best baseline method obtained by Speaker2Vec models averaged across all datasets.*

Evaluation dataset	Different distance metrics with MFCC features				Speaker2Vec models			Adapted Speaker2Vec models		
	BIC with MFCC13	GD with MFCC13	KL2 with MFCC13	KL with MFCC40	Tedlium	YouTube	YouTube large	Tedlium	YouTube	YouTube large
TEDLIUM test	(47.50, 65.35)	(60.00, 61.54)	(62.50, 63.94)	(78.50, 76.05)	(32.00, 32.32)	(44.00, 46.38)	(44.50, 43.58)	**(26.00, 28.26)**	**(30.00, 33.87)**	(40.00, 31.19)
Couples Therapy	(49.11, 44.00)	(50.79, 44.09)	(51.92, 44.24)	(57.83, 57.95)	(42.39, 42.83)	(42.95, 42.36)	(43.96, 44.88)	**(42.45, 42.09)**	**(42.28, 42.36)**	(43.68, 44.00)
RT-06 (tol. ±0.25s)	(75.35, 69.61)	(76.51, 70.71)	(76.51, 69.61)	(75.32, 75.38)	(64.74, 64.41)	(65.33, 63.54)	(64.08, 65.21)	**(63.00, 63.31)**	**(62.43, 61.94)**	(63.95, 62.99)
RT-06 (tol. ±0.5s)	(51.71, 50.84)	(52.65, 48.57)	(53.47, 48.47)	(53.47, 48.47)	(46.31, 47.81)	(47.59, 47.05)	(51.72, 48.34)	(47.43, 45.80)	**(44.69, 45.42)**	**(44.92, 45.87)**
Mean improvement w.r.t. best baseline					9.92	6.62	5.73	**11.72**	**11.14**	9.44

Table 3: *Comparison of the proposed method with some state-of-the-art papers along with the characteristics (duration and actual number of change points) of the artificial dialogs they created from TIMIT dataset for evaluation. The best two results are in* ***boldface****.*

Method	Delacourt et al. [27]	Kotti et al. [28]	Kotti et al. [29]	Chen et al. [7]	Ma et al. [30]	Speaker2Vec models			Adapted Speaker2Vec models		
						Tedlium	YouTube	YouTube large	Tedlium	YouTube	YouTube large
Duration (mins)	–	6.42	60	16.67	160	45.22					
# change points	60	136	935	250	3200	1010					
Unsupervised?	Yes	Yes	Yes	No	No	Yes					
F1 score	–	0.73	0.78	0.74	0.79	0.82	0.82	0.83	**0.86**	**0.85**	**0.85**
EER	(15.6, 28.2)	(30.5, 21.8)	(5.1, 28.9)	(19, 25)	(16.9, 18.8)	(18.51, 18.19)	(17.62, 17.33)	(17.23, 17.77)	**(14.75, 14.99)**	**(14.95, 15.03)**	(15.74, 15.54)

in mean EER for the adapted TED-LIUM and YouTube models respectively. Figure 2 shows the ROC curves [34] obtained by different algorithms on TED-LIUM test dataset. One important point is that even though the TED-LIUM model performs best for the TED-LIUM test data (which is expected), the YouTube models give competitive results, and they perform much better than the baselines.

4.2. Comparison with other state-of-the-art works

To compare the performance of the proposed method with more sophisticated speaker segmentation algorithms, we have adopted TIMIT dataset [35]. We have created ten artificial dialogs by randomly choosing speakers from the dataset. We have removed the inter-speaker silence regions similar to [29] to make the problem more challenging. Table 3 tabulates the characteristics of the artificial data created from TIMIT corpus for evaluation in different papers, along with their learning strategies, F1 scores [29], and EERs. We achieve 6% and 7% absolute increments in F1 score over the best published work [30] with adapted YouTube and TED-LIUM models respectively.

5. Conclusions and future directions

In this paper, we have proposed a novel method for learning a speaker-characteristics manifold in an unsupervised manner. We have also proposed a strategy for unsupervised domain adaptation. The Speaker2Vec embeddings performed quite well in speaker segmentation, evaluated on different datasets from varying acoustic environments, even with a very simple distance metric. One major benefit is that these embeddings can be used in any speaker segmentation or diarization system just by replacing MFCCs with the embeddings.

We have three major future directions from here. We are working towards extending this work to diarization, where we will use the speaker change points generated by the proposed system, and do a clustering on top of that. We will be applying the embeddings for speaker recognition tasks [3] as well. We can also use the proposed approach in some different domain where scarcity of large amount of labeled data is a major issue. We already have some promising results in learning behavior manifolds from unlabeled audio [36]. Given the easy availability of large amount of unlabeled data, unsupervised deep learning like this can give even better results than supervised methods trained on some specific datasets.

6. Acknowledgements

The U.S. Army Medical Research Acquisition Activity, 820 Chandler Street, Fort Detrick MD 21702- 5014 is the awarding and administering acquisition office. This work was supported by the Office of the Assistant Secretary of Defense for Health Affairs through the Psychological Health and Traumatic Brain Injury Research Program under Award No. W81XWH-15-1-0632. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the Department of Defense.

7. References

[1] M. Kotti, V. Moschou, and C. Kotropoulos, "Speaker segmentation and clustering," *Signal processing*, vol. 88, no. 5, pp. 1091–1124, 2008.

[2] K. Chen, "Towards better making a decision in speaker verification," *Pattern Recognition*, vol. 36, no. 2, pp. 329–346, 2003.

[3] J. P. Campbell, "Speaker recognition: A tutorial," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1437–1462, 1997.

[4] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[5] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[6] H. Lee, P. Pham, Y. Largman, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Advances in neural information processing systems*, 2009, pp. 1096–1104.

[7] 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, 2011.

[8] S.-S. Cheng, H.-M. Wang, and H.-C. Fu, "Bic-based speaker segmentation using divide-and-conquer strategies with application to speaker diarization," *IEEE transactions on audio, speech, and language processing*, vol. 18, no. 1, pp. 141–157, 2010.

[9] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker diarization: A review of recent research," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 356–370, 2012.

[10] 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.

[11] J.-F. Bonastre, P. Delacourt, C. Fredouille, T. Merlin, and C. Wellekens, "A speaker tracking system based on speaker turn detection for nist evaluation," in *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, vol. 2. IEEE, 2000, pp. II1177–II1180.

[12] P. Woodland, M. Gales, D. Pye, and S. Young, "The development of the 1996 htk broadcast news transcription system," in *DARPA speech recognition workshop*. Morgan Kaufmann Pub, 1997, pp. 73–78.

[13] S. Chen and P. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the bayesian information criterion," in *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, vol. 8. Virginia, USA, 1998, pp. 127–132.

[14] 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*, vol. 1997, 1997.

[15] R. Gangadharaiah, B. Narayanaswamy, and N. Balakrishnan, "A novel method for two-speaker segmentation." in *Proc. of ICSLP, Jeju, Korea*, 2004, pp. 2337–2340.

[16] C. Barras, X. Zhu, S. Meignier, and J.-L. Gauvain, "Multistage speaker diarization of broadcast news," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1505–1512, 2006.

[17] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE transactions on acoustics, speech, and signal processing*, vol. 28, no. 4, pp. 357–366, 1980.

[18] H. Hermansky, "Perceptual linear predictive (plp) analysis of speech," *the Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[19] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[20] M. Rouvier, P.-M. Bousquet, and B. Favre, "Speaker diarization through speaker embeddings," in *Signal Processing Conference (EUSIPCO), 2015 23rd European*. IEEE, 2015, pp. 2082–2086.

[21] S. H. Yella and A. Stolcke, "A comparison of neural network feature transforms for speaker diarization." in *INTERSPEECH*, 2015, pp. 3026–3030.

[22] V. Gupta, "Speaker change point detection using deep neural nets," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4420–4424.

[23] J. Ramirez, J. M. Górriz, and J. C. Segura, *Voice activity detection. fundamentals and speech recognition system robustness*. IN-TECH Open Access Publisher NewYork, 2007.

[24] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[25] A. Rousseau, P. Deléglise, and Y. Esteve, "Ted-lium: an automatic speech recognition dedicated corpus." in *LREC*, 2012, pp. 125–129.

[26] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011, iEEE Catalog No.: CFP11SRW-USB.

[27] P. Delacourt and C. J. Wellekens, "Distbic: A speaker-based segmentation for audio data indexing," *Speech communication*, vol. 32, no. 1, pp. 111–126, 2000.

[28] M. Kotti, E. Benetos, and C. Kotropoulos, "Automatic speaker change detection with the bayesian information criterion using mpeg-7 features and a fusion scheme," in *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*. IEEE, 2006, pp. 4–pp.

[29] ——, "Computationally efficient and robust bic-based speaker segmentation," *IEEE transactions on audio, speech, and language processing*, vol. 16, no. 5, pp. 920–933, 2008.

[30] Y. Ma and C.-C. Bao, "Sparse dnn-based speaker segmentation using side information," *Electronics Letters*, vol. 51, no. 8, pp. 651–653, 2015.

[31] "Spring 2006 (rt-06s) rich transcription meeting recognition evaluation plan," http://www.itl.nist.gov/iad/mig/tests/rt/2006-spring/docs/rt06s-meeting-eval-plan-V2.pdf.

[32] A. Christensen, D. C. Atkins, S. Berns, J. Wheeler, D. H. Baucom, and L. E. Simpson, "Traditional versus integrative behavioral couple therapy for significantly and chronically distressed married couples." *Journal of consulting and clinical psychology*, vol. 72, no. 2, p. 176, 2004.

[33] M. Black, A. Katsamanis, C.-C. Lee, A. C. Lammert, B. R. Baucom, A. Christensen, P. G. Georgiou, and S. S. Narayanan, "Automatic classification of married couples' behavior using audio features." in *INTERSPEECH*, 2010, pp. 2030–2033.

[34] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.

[35] "Timit acoustic-phonetic continuous speech corpus," https://catalog.ldc.upenn.edu/LDC93S1, lDC Catalog No. LDC93S1.

[36] H. Li, B. Baucom, and P. Georgiou, "Unsupervised latent behavior manifold learning from acoustic features: Audio2behavior," in *Proceedings of IEEE International Conference on Audio, Speech and Signal Processing (ICASSP)*, March 2017.