Vocal Tract Length Perturbation for Text-Dependent Speaker Verification with Autoregressive Prediction Coding

Sarkar, Achintya; Tan, Zheng-Hua

Published in:
I E E E Signal Processing Letters

DOI (link to publication from Publisher):
10.1109/LSP.2021.3055180

Publication date:
2021

Document Version
Accepted author manuscript, peer reviewed version

Citation for published version (APA):
Sarkar, A., & Tan, Z.-H. (2021). Vocal Tract Length Perturbation for Text-Dependent Speaker Verification with Autoregressive Prediction Coding. *I E E E Signal Processing Letters*, *28*, 364-368. Article 9339931. https://doi.org/10.1109/LSP.2021.3055180

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2021.3055180, IEEE Signal Processing Letters

Vocal Tract Length Perturbation for Text-Dependent Speaker Verification with Autoregressive Prediction Coding

Achintya kr. Sarkar, Zheng-Hua Tan, *Senior Member, IEEE*

Abstract—**In this letter, we propose a vocal tract length (VTL) perturbation method for text-dependent speaker verification (TD-SV), in which a set of TD-SV systems are trained, one for each VTL factor, and score-level fusion is applied to make a final decision. Next, we explore the bottleneck (BN) feature extracted by training deep neural networks with a self-supervised learning objective, autoregressive predictive coding (APC), for TD-SV and comapre it with the well-studied speaker-discriminant BN feature. The proposed VTL method is then applied to APC and speaker-discriminant BN features. In the end, we combine the VTL perturbation systems trained on MFCC and the two BN features in the score domain. Experiments are performed on the RedDots challenge 2016 database of TD-SV using short utterances with Gaussian mixture model-universal background model and i-vector techniques. Results show the proposed methods significantly outperform the baselines.**

Index Terms: VTL factor, Autoregressive prediction coding, GMM-UBM, I-vector, Text-dependent speaker verification

I. Introduction

Speaker verification (SV) is the task of verifying a person using his/her speech signal, which can be either text-independent (TI) or text-dependent (TD). In a TI-SV system, speakers are free to speak any text content during the system enrollment and test phases. On the contrary, TD-SV constrains a speaker to speak a particular pass-phrase from a predefined list in both enrollment and test phases. As TD-SV maintains the matched phonetic contents between training and test phases, it is superior to TI-SV for SV with short utterances and ideal for real-world applications.

An important factor that characterises speech signals is vocal tract length (VTL), which varies from person to person and is of importance to deal with in speech systems including speaker recognition and automatic speech recognition (ASR). For example, vocal tract length normalization (VTLN) [1], [2], [3] is widely used in ASR to reduce the effect of speaker variability due to the difference in VTL among speakers. In [4], VTL warped features are used for ASR as a data augmentation technique.

The use of VTL information in speaker recognition is rather limited with few applications in *text-independent* context only, e.g., speaker identification [5] and verification [6]. In [5], VTL warped features are used for building a Gaussian mixture model-universal background model (GMM-UBM) called VTLN-UBM for speaker identification. Then speaker models are derived from the GMM-UBM using the enrolment data of each speaker. During test, feature vectors of a test utterance is scored against the models of registered speakers to identify the best matching speaker. Only training data are warped, but enrollment and test data are the original cepstral features without warping. In [6], [7], different GMM-UBM models are trained based on the VTL factors of speakers in training data; each model corresponds to one VTL factor value. During enrolment, speaker models are derived from the VTL factor-specific GMM-UBM that matches their VTL factor values by using enrolment data. During test, the feature vectors of the test utterance are scored against the claimant speaker model and the corresponding GMM-UBM. All training, enrollment and test data are unwarped.

In summary, VTL warped features have been used for TI speaker identification and the VTL concept has been exploited for TI-SV. To the best of our knowledge, no prior work exploits VTL warped features for SV. Furthermore, no prior work applies the VTL concept for TD-SV. As VTL is an important factor characterising speakers, *we propose a VTL perturbation method for TD-SV.* Specifically, the proposed method trains a set of TD-SV systems, one for each VTL factor, and decision level fusion is applied to combine the systems to make final SV decision. Note that the VTLN-UBM method in [5] is not suitable for SV as VTLN-UBM is trained with warped features whereas targets are derived with unwarped features during adaptation. Therefore a test unwarped feature will mismatch with the VTLN-UBM. Paper [6] extends the concept of gender dependent SV further by grouping the speaker based on their VTL values, which can be used in conjunction with the proposed method.

Recently, self-supervised learning has attracted significant attention in the community. A notable method of such is based on autoregressive prediction coding (APC), and it has shown to be able to learn powerful speech representations, which perform extremely well for several downstream tasks including ASR and speaker identification [8], [9]. In APC, a deep neural network (DNN) encoder is trained with the objective function to predict future frames. The output from a selected hidden layer is considered as the APC feature. It is shown in [8] that the APC feature significantly outperforms the widely-used cepstral feature. However, *the performance of APC on TD-SV has not been established yet*. Besides, we compare it to a bottleneck (BN) feature trained in a supervised fashion. In the literature, there are a number of BN features

A. K. Sarkar is with Indian Institute of Information Technology, Sri City, India (E-mail: sarkar.achintya@gmail.com). Z.-H. Tan is with Department of Electronic Systems, Aalborg University, Denmark (E-mails: zt@es.aau.dk).

such as those trained to discriminate speakers [10], senones [11], a combination of them [10], or tri-phone state [10], time-contrastive-learning (TCL) and phone discriminate [12]. In this work, we choose the BN feature discriminating speakers (Spk-BN) for comparison.

To investigate joint effect of BN features and VTL perturbation on TD-SV, we apply the proposed VTL method to the BN features including both APC-BN and Spk-BN. In the end, we fuse the systems based on MFCC, APC-BN and Spk-BN in the score domain.

We conducted experiments on the RedDots dataset of TD-SV using short utterances [13] and show encouraging results under both GMM-UBM and i-vector modeling frameworks. The choice of these modelling methods is based on the comparison in [14], which shows that GMM-UBM remains a better choice for TD-SV using short utterances than other methods. Although x-vector [15] has shown promising results for TI-SV, it is not successful so far for TD-SV possibly due to limited training data [16].

II. Features and TD-SV modeling methods

A. Features

1) Spk-BN feature: During training, Mel-frequency cepstral coefficients (MFCCs) are fed to a DNN to discriminate speakers with coss-entropy objective function. When using it, the output of a particular hidden layer of the DNN for a given speech segment at frame-level is projected onto the low dimensional space using principal component analysis (PCA) to get Spk-BN feature [12].

2) APC-BN feature: During training, MFCC feature vectors are fed to a DNN encoder with a objective function to minimize the $L1$ loss between the predicted sequence $(\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N)$ and the target sequence $(\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots \boldsymbol{t}_N)$ by right-shifting the input sequence $(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)$ of n time steps:

$$\sum_{i=1}^{N-n} |\boldsymbol{t}_i - \boldsymbol{y}_i|, \quad \boldsymbol{t}_i = \boldsymbol{x}_{i+n}. \tag{1}$$

When using it, the output from a particular hidden layer of the DNN for a given utterance is extracted at frame-level to get the APC feature of 512 dimensions, which is then used for text-independent speaker identification [8] and and verification [9]. In this work, the 512-dimensional APC feature is projected onto the low dimensional space using PCA to get APC-BN feature of the same dimension as Spk-BN.

B. Speaker verification methods

1) GMM-UBM technique: A GMM-UBM [17] is trained using data from many non-target speakers. Speakers are then represented by their models derived from the GMM-UBM using the enrollment data of the particular speaker with maximum a posteriori (MAP) adaptation. During test, the feature vectors of a test utterance $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ are scored against the claimant model λ_r (obtained in the enrollment phase) and GMM-UBM λ_{ubm}, respectively. Finally, a log-likelihood ratio (LLR) value $\Lambda(\boldsymbol{X})$ is calculated using scores between the claimant and GMM-UBM models. The LLR value is used to accept or reject the claimant.

$$\Lambda(\mathbf{X}) = \frac{1}{N}\big[\log p(\mathbf{X}|\lambda_r) - \log p(\mathbf{X}|\lambda_{ubm})\big] \tag{2}$$

2) i-vector: In this technique [18], a speech signal is represented by a vector called *i-vector*. An i-vector is obtained by decomposing the speaker and channel dependent GMM super-vector $\boldsymbol{M}$ as

$$\boldsymbol{M} = \boldsymbol{m} + \boldsymbol{T}\boldsymbol{w} \tag{3}$$

where $\boldsymbol{m}$ denotes the speaker-independent GMM super-vector and $\boldsymbol{w}$ is called an i-vector. $\boldsymbol{T}$ is the total variability space (T-space) on a subspace of $\boldsymbol{m}$, where speaker and channel information is assume to be dense. During the training phase, each target is represented by an average i-vector computed over his/her training utterance-wise (or session-wise) i-vectors. During test, the i-vector of the test utterance is scored against the claimant specific i-vector (obtained during enrolment) with probabilistic linear discriminant analysis (PLDA) [19].

III. Proposed VTL perturbation for TD-SV

A. Concept of VTL

In this concept [20], the spectra for a spoken content between the two speakers (A and B) are matched by scaling the frequency axis of the signal spectra,

$$S_A(f) = S_B(\alpha f) \tag{4}$$

where, the α is called VTL (or VTL warp) factor. Fig. 1 shows the smoothed spectra of vowel /eh/ uttered by a male and a female speaker. To match the formants of the male speaker to those of the female speaker (as reference), it is required to scale the frequency axis, i.e., to expand the spectra of the male speaker. Based on the psychological structure of male and female speakers, α generally lies in the range $[0.80, 1.20]$ with steps of 0.02. To maintain the bandwidth of original signal (f) with warped frequency spectra (f_w), the warped frequency is expressed as a piece-wise linear [20],

$$f_w = (\alpha f) = \begin{cases} \alpha f, & 0 \leq f \leq f_0 \\ \frac{f_{max} - \alpha f_0}{f_{max} - f_0}(f - f_0) + \alpha f_0, & f_0 \leq f \leq f_{max} \end{cases} \tag{5}$$

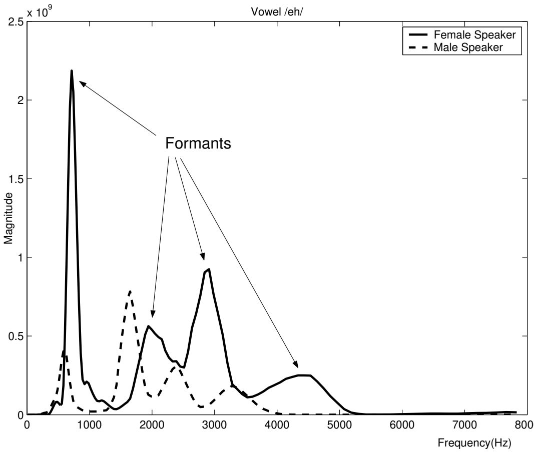

Fig. 1. *The spectra of vowel /eh/ for male and female speakers [6].*

where f_{max} and f_0 denote the signal maximum bandwidth and empirical upper boundary frequency up to α linearly warped, respectively.

B. Proposed method

Conventional TD-SV systems use MFCC features with $\alpha = 1.0$. In the proposed method, MFCC features are extracted for different values of VTL factors from 0.80 to 1.20 with increment value of 0.02 by warping the frequency axis of spectra for a given utterance as per [20]. This gives 21 sets of cepstral features (incl. $\alpha = 1.0$) for a given speech utterance and the sets of features are used for the development of TD-SV systems, separately, i.e., for training DNN, GMM-UBM, T-space and PLDA, leading to one system per VTL factor.

When APC-BN and Spk-BN features are used, DNN models for BN feature extraction are trained using cepstral features warped for each VTL factor. Afterward, the cepstral feature of an utterance for a particular VTL factor is fed to the corresponding DNN for Spk-BN and APC-BN. It can be expressed as

$$\boldsymbol{x}_\alpha^{BN} = f_{DNN_\alpha}(\boldsymbol{x_\alpha}), \quad \alpha\epsilon\,[0.80, 1.20] \tag{6}$$

where $\boldsymbol{x}_\alpha^{BN}$ denotes the Spk-BN or APC-BN feature extracted from MFCC with VTL factor α for a given speech utterance x using the DNN corresponding to VTL factor α. The extracted features are used for building the TD-SV. It gives 21 TD-SV systems for Spk-BN and APC-BN, respectively. Finally, scores of the different VTL feature based TD-SV systems are fused with equal importance and used for decision making,

$$Score_{fusion} = \frac{1}{N}\sum_{i=1}^{N} Score_i \tag{7}$$

This is compared to systems use MFCC, Spk-BN and APC-BN features for ($\alpha = 1.0$).

IV. Experimental setup

Experiments for TD-SV are conducted on the *m-part-01* task (for male speakers) of the RedDots database as per protocol [13]. There are 320 target models to train, each having three utterances/speech samples. Each utterance is approximately 2 seconds duration on average. There are four types of trials for system evaluation shown in Table I. More details can be found in [13].

- *Genuine trial* is when a target speaker claims the identity during test by speaking a pass-phrase included in the enrollment.
- *Target-wrong* is when a target speaker claims the identity during test by speaking a pass-phrase that is not included in the enrollment.
- *Imposter-correct* is when an imposter falsely claims as a genuine speaker with pronouncing a pass-phrase included in the enrollment.
- *Imposter-wrong* is when an imposter falsely claims as a genuine speaker with pronouncing a pass-phrase that is not included in the enrollment.

TABLE I
Number of trials in evaluation set.

# of	# of trials in Non-target type		
Genuine trials	Target -wrong	Imposter -correct	Imposter -wrong
3242	29178	120086	1080774

The RSR2015 database [21] is used to train the total variability space and PLDA in the i-vector system and to train DNNs for Spk-BN and APC-BN feature extraction. The database consists of 157 male and 143 female speakers and of 72764 utterances over 27 pass-phrases after excluding the pass-phrases common with the RedDots database.

MFCC feature vectors of 57 dimension (i.e., 19 static and their $\Delta, \Delta\Delta$) with RASTA filtering [22] are extracted from speech signals using a $25ms$ hamming window and a $10ms$ frames shift for different values of α. The upper boundary frequency required in VTL [20] is set 85% of highest frequency in the speech signal. HTK toolkit [23] is used for feature extraction. An open-source robust voice activity detection (rVAD)[1] [24] is applied to discard the less energized frames. The selected frames are then normalized to fit zero mean and unit variance at the utterance level.

A GMM-UBM with 512 mixtures having diagonal covariance matrices is trained using 6300 speech files from the TIMIT database [25] consisting of 438 males and 192 females. This data set is also used for training low dimensional PCA space in BN feature extraction. In MAP adaptation, 3 iterations with the value of relevance factor 10, are considered. In PLDA, the utterances of the same pass-phrase from a particular speaker are treated as an individual speaker. It gives 8100 classes (4239 males and 3861 females) in PLDA. Speaker and channel factors are kept full in PLDA, i.e. equal to the dimension of i-vector (400). Before PLDA, i-vectors are normalized with spherical normalization [26].

For Spk-BN, TensorFlow [27] is used for training the DNNs with the following settings: variable batch sizes from 256 to 1024, variable learning rates from 0.8 to 0.08, and 30 training epochs. Seven layers of feed-forward networks with a sigmoid activation function are used. The input layer considers the context window of 11 frames (i.e. 5 frames left, current frame, 5 frames right). Each hidden layer consists of 1024 neurons. The number of nodes at output is equal to the number of speakers, i.e., 300 in DNN training data. In Spk-BN, the frame-level output from the fourth hidden layer of DNNs is projected onto 57 dimensional space to align with the dimension of the MFCCs feature for a fair comparison as per [12].

Deep APC features are extracted as per [8]. DNN encoder consists of 3 hidden layers GRU with ReLU activation function, the output layer is linear, the batch size is 32, and the learning rate is 0.001, and $n = 5$ as in Eq 1 for APC (which gives the best performance in [8]). Output from the last hidden layer of DNN at frame-level is projected onto 57 dimensional space to get APC-BN feature. System performance is measured in terms of equal error rate (EER)

[1] https://github.com/zhenghuatan/rVAD

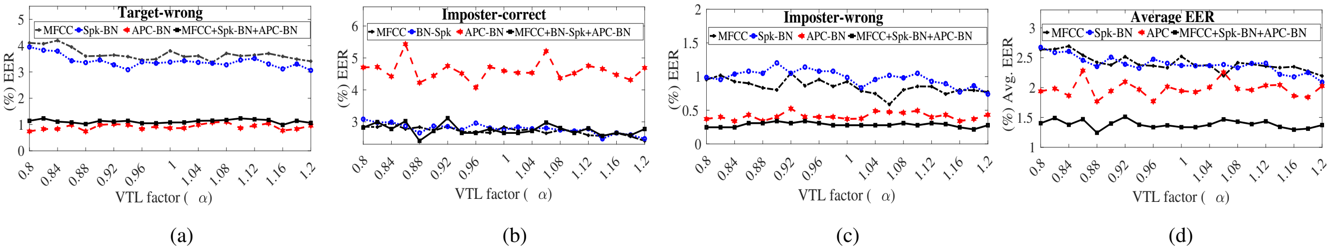

Fig. 2. *Comparison of TD-SV performance between MFCC, Spk-BN and APC-BN features and their score-level fusion across different values of* α *and different trials on the RedDots database with the GMM-UBM technique:* **(a)** *Target-wrong* **(b)** *Imposter-wrong* **(c)** *Imposter-wrong and* **(d)** *Average EER over (a), (b) and (c).*

and minimum detection cost function (minDCF) as per the 2008 SRE [28].

V. Results and Discussions

This section presents experimental results on TD-SV.

A. Results under the GMM-UBM framework

Fig. 2 shows TD-SV results of MFCC, Spk-BN and APC-BN features and their score-level fusion for different VTL factors (α) on the RedDots database. We observe that APC-BN yields lowest EER values for Target-wrong and Imposter-wrong over all values of α. However, it fails in Imposter-correct as compared to MFCC and Spk-BN. This can be explain by the fact that the APC objective is to predict the content of future frames and thus the APC-BN feature is sensitive to the linguistic content. On average, APC-BN significantly outperforms other features. Fusion of three feature systems further reduces the EERs.

Table II presents the results of the proposed VTL perturbation method and its combination with APC-BN and Spk-BN features. It is seen that the proposed method significantly reduces the average EERs for all features. Fusion of the feature-dependent systems further improves the performance; the EER reduces from 2.52% of the baseline MFCC to 1.18% of the fused system. Similar observation is seen in terms of minDCF. Note that the average EER for MFCC is 2.52% as compared to 3.19% in [12], and the improvement is due to the use of rVAD [24] in this work instead of an energy based one.

B. Results under the i-vector framework

Table III presents the corresponding results using the i-vector technique, and we can see similar observations to those of GMM-UBM.

VI. Conclusion

In this letter, we proposed a VTL perturbation method for TD-SV and experimentally demonstrated its effectiveness. We further explored the performance of APC-BN for TD-SV in comparison to the cepstral feature and the speaker-discriminant BN feature, and APC-BN showed superior performance. The VTL perturbation method was then applied to BN features, which improved the performance further. In the end, fusion of three feature-dependent systems performs the best.

TABLE II

TD-SV performance of the proposed VTL perturbation method and its combination with APC-BN and Spk-BN features on the RedDots database in terms of %EER/(MinDCF× 100) with GMM-UBM.

System	Non-target type			Avg.
	Target-wrong	Imposter correct	Imposter-wrong	EER/ MinDCF
Baseline:				
MFCC (α=1.0)	3.80/1.38	2.83/1.19	0.92/0.28	2.52/0.95
Spk-BN (α=1.0)	3.37/1.34	2.74/1.27	0.98/0.28	2.36/0.96
APC-BN (α=1.0)	0.86/0.29	4.59/2.03	0.37/0.09	1.94/0.81
Proposed:				
MFCC (all α)	3.06/1.23	2.19/1.00	0.52/0.14	1.92/0.79
Spk-BN (all α)	2.52/1.03	**2.03/0.89**	0.49/0.13	1.68/0.68
APC-BN (all α)	**0.64/0.19**	3.80/1.71	0.27/0.05	1.57/0.65
Score fusion:				
MFCC + Spk-BN (α=1.0)	3.26/1.22	2.51/1.09	0.77/0.19	2.18/0.84
MFCC + APC-BN (α=1.0)	0.95/0.35	3.08/1.43	0.30/0.07	1.44/0.62
MFCC + [Spk + APC]-BN (α=1.0)	1.07/0.38	2.65/1.26	0.27/0.07	1.33/0.57
MFCC + Spk-BN(all α)	2.74/1.10	1.91/0.91	0.46/0.11	1.70/0.71
MFCC + APC-BN (all α)	0.85/0.25	2.83/1.30	**0.24/0.04**	1.31/0.53
MFCC + [Spk + APC]-BN (all α)	0.92/0.30	2.37/1.12	**0.24/0.04**	**1.18/0.49**

TABLE III

TD-SV performance of the proposed VTL perturbation methdo and its combination with APC-BN and Spk-BN features on the RedDots database in terms of [%EER/(MinDCF× 100)]) with i-vector.

System	Non-target type			Avg.
	Target-wrong	Imposter correct	Imposter-wrong	EER/ MinDCF
Baseline:				
MFCC ($\alpha = 1.0$)	6.07/2.60	4.07/1.81	1.17/0.46	3.77/1.62
Spk-BN ($\alpha = 1.0$)	6.39/2.78	4.87/1.98	1.60/0.63	4.29/1.80
APC-BN (α=1.0)	2.07/0.65	7.97/3.22	0.77/0.28	3.60/1.38
Proposed:				
MFCC (all α)	4.88/2.10	2.80/1.23	0.67/0.25	2.78/1.19
Spk-BN (all α)	4.07/1.85	2.96/1.28	0.60/0.23	2.54/1.12
APC-BN (all α)	1.10/0.33	6.07/2.18	0.46/0.09	2.54/0.87
Score fusion:				
MFCC + Spk-BN (α=1.0)	3.26/1.22	**2.51/1.09**	0.77/0.19	2.18/0.84
MFCC + APC-BN (α=1.0)	1.69/0.58	5.27/2.12	0.49/0.13	2.48/0.94
MFCC + [Spk + APC]-BN (α=1.0)	1.56/0.58	4.10/1.68	0.43/0.11	2.03/0.79
MFCC + Spk-BN (all α)	4.27/1.88	2.71/1.16	0.57/0.21	2.51/1.08
MFCC + APC-BN (all α)	**0.98/0.35**	4.03/1.47	**0.33/0.06**	1.78/0.63
MFCC+ [Spk + APC]-BN (all α)	1.07/0.36	3.28/1.23	**0.33/0.06**	**1.56/0.55**

References

[1] R. Serizel and D. Giuliani, "Vocal tract length normalisation approaches to dnn-based children's and adults' speech recognition," in *IEEE Spoken Language Technology Workshop (SLT)*, 2014, pp. 135–140.

[2] S. Umesh, "Studies on inter-speaker variability in speech and its application in automatic speech recognition," *Sadhana*, vol. 36, pp. 853–883, 2011.

[3] P. T. Akhil, S. P. Rath, S. Umesh, and D. R. Sanand, "A computationally efficient approach to warp factor estimation in vtln using em algorithm and sufficient satistics," in *Proc. of Interspeech*, 2008, pp. 1713–1716.

[4] N. Jaitly and G. E. Hinton, "Vocal tract length perturbation vtlp improves speech recognition," in *Proc. of International Conference on Machine Learning (ICML)*, 2013.

[5] A. K. Sarkar, S. Umesh, and S. P. Rath, "Text-independent speaker identification using vocal tract length normalization for building universal background model," in *Prof. of Interspeech*, 2009, pp. 2331–2334.

[6] A. K. Sarkar and S. Umesh, "Investigation of speaker-clustered ubms based on vocal tract lengths and mllr matrices for speaker verification," in *Proc. of Odyssey Speaker and Language Recognition Workshop*, 2010, pp. 286–293.

[7] W. Q. Zhang, Y. Shan, and J. Liu, "Multiple background models for speaker verification," in *Proc. of Odyssey Speaker and Language Recognition Workshop*, 2010, pp. 47–51.

[8] Y.-A. Chung and J. Glass, "Generative pre-training for speech with auto-regressive predictive coding," in *Proc of IEEE Int. Conf. Acoust. Speech Signal Processing (ICASSP)*, 2020, pp. 3497–3501.

[9] Y. Chung, W.-N. Hsu, H. Tang, and J. Glass, "An unsupervised autoregressive model for speech representation learning," in *Proc. Interspeech*, 2019, pp. 146–150.

[10] 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, 2015.

[11] M. McLaren, Y. Lei, and L. Ferrer, "Advances in deep neural network approaches to speaker recognition," in *Proc. of IEEE Int. Conf. Acoust. Speech Signal Processing (ICASSP)*, 2015, pp. 4814–4818.

[12] A. K. Sarkar, Z.-H. Tan, H. Tang, S. Shon, and J. R. Glass, "Time-contrastive learning based deep bottleneck features for text-dependent speaker verification," *IEEE/ACM Trans. Audio, Speech & Language Processing*, vol. 27, no. 8, pp. 1267–1279, 2019.

[13] "The reddots challenge: Towards characterizing speakers from short utterances," https://sites.google.com/site/thereddotsproject/reddots-challenge.

[14] M. Sahidullah, A. K. Sarkar, V. Vestman, X. Liu, R. Serizel, T. Kinnunen, Z.-H. Tan, and E. Vincent, "UIAI system for short-duration speaker verification challenge 2020," in *Accepted at Spoken Language Technology (SLT) Workshop*, 2021.

[15] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: robust dnn embeddings for speaker recognition," in *Proc. of IEEE Int. Conf. Acoust. Speech Signal Processing (ICASSP)*, 2018, pp. 5329–5333.

[16] Hossein Zeinali, Kong Aik Lee, Jahangir Alam, and Lukas Burget, "Short-duration speaker verification (sdsv) challenge 2020: the challenge evaluation plan," *arXiv preprint arXiv:1912.06311*, 2019.

[17] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, pp. 19–41, 2000.

[18] N. Dehak, P. Kenny, R. Dehak, P. Ouellet, and P. Dumouchel, "Front-end factor analysis for speaker verification," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 19, pp. 788–798, 2011.

[19] M. Senoussaoui et al., "Mixture of plda models in i-vector space for gender-independent speaker recognition," in *Proc. of Interspeech*, 2011, pp. 25–28.

[20] L. Lee and R. Rose, "Frequency warping approach to speaker normalization," *IEEE Trans. on Speech and Audio Processing*, vol. 6, pp. 49–59, 1998.

[21] A. Larcher, K. A. Lee, B. Ma, and H. Li, "Text-dependent speaker verification: classifiers, databases and rsr2015," *Speech Communucation*, vol. 60, pp. 56–77, 2014.

[22] H. Hermanksy and N. Morgan, "Rasta processing of speech," *IEEE Trans. on Speech and Audio Processing*, vol. 2, pp. 578–589, 1994.

[23] S. Young, D. Kershaw, J. Odell, V. Valtchev, P. Woodland, and et al., "HTK book," *Copyright 2001-2006 Cambridge University Engineering Department*.

[24] Z.-H. Tan, A. K. Sarkar, and N. Dehak, "rVAD: an unsupervised segment-based robust voice activity detection method," *Computer Speech & Language*, vol. 59, pp. 1–21, 2020.

[25] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue, "TIMIT acoustic-phonetic continuous speech corpus ldc93s1," 1993, Web Download. Philadelphia: Linguistic Data Consortium.

[26] P. M. Bousquet et al., "Variance-spectra based normalization for i-vector standard and probabilistic linear discriminant anal ysis," in *Proc. of Odyssey Speaker and Language Recognition Workshop*, 2012, pp. 157–164.

[27] M. Abadi et al., "TensorFlow: large-scale machine learning on heterogeneous systems," 2015, Software available from tensorflow.org.

[28] "https://www.nist.gov/itl/iad/mig/2008-nist-speaker-recognition-evaluation-results," .