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

Phonetically-aware embeddings, Wide Residual Networks with Time-Delay Neural Networks and Self Attention models for the 2018 NIST Speaker Recognition Evaluation

Ignacio Viñals, Dayana Ribas, Victoria Mingote, Jorge Llombart, Pablo Gimeno, Antonio Miguel, Alfonso Ortega, Eduardo Lleida

ViVoLAB, Aragón Institute for Engineering Research (I3A)
University of Zaragoza, Spain
{ivinalsb,dribas,vmingote,jllombg,pablogj,amiguel,ortega,lleida}@unizar.es

Abstract

Very often, speaker recognition systems do not take into account phonetic information explicitly. In order to gain insight along this line of research, we have studied the use of phonetic information in the embedding extraction process for automatic speaker verification systems in two different ways: on the one hand using the well-known i-vector paradigm and, on the other hand, using Wide Residual Networks (WRN) with Time Delay Neural Networks (TDNN) and Self-Attention Mechanisms. The phonetic information is provided by a WRN with TDNN using 1D convolutional layers specifically trained for this purpose. These two approaches along with the widely used x-vector system based on the Kaldi toolkit were submitted to the 2018 NIST speaker recognition evaluation. As back-end, these representations used a standard PLDA classifier with ad-hoc configurations for each system and in-domain adaptation. The results obtained in the NIST SRE 2018 show that our methods are very promising and it is worth continuing to work on them to improve their performance.

Index Terms: NIST-SRE, speaker verification, Wide Residual Networks, Time-Delay Neural Networks, phonetically-aware embeddings

1. Introduction

The way we identify persons by their voice is heavily dependent on the phonetic content of the utterance. In addition to this, the way phonemes are produced are very speaker dependent [1, 2]. Nevertheless, Automatic Speaker Verification (ASV) usually do not consider phonetic information explicitly. To exploit this discriminant information, we propose here two different ways of extracting phonetically-aware speaker representations. First, we present a way of using phonetic information inside the well-known i-vector paradigm. Secondly, Wide Residual Network (WRN) with Time-Delay Neural Networks and Self-Attention Mechanisms are used to extract the speaker embedding.

This study is presented in the framework of the 2018 NIST Speaker Recognition Evaluation. Traditionally focused on conversational telephone speech (CTS) in English, recorded through a Public Switched Telephone Network (PSTN), this NIST-SRE 2018 [3] includes new types of audio: Voice over IP (VoIP) data and Audio from Video (AfV), collected “in the wild”. Furthermore, the challenges in this NIST-SRE 2018 lie on the multiple environments that the submitted systems have to deal with. While past CTS databases, except the 2016 edition [4], were mainly collected in North-America with English as the main language, audio data for this evaluation was recorded in Arabic Tunisian language through the Tunisian PSTN. This way, a strong language, and channel mismatch must be taken into account. Regarding the AfV speech, acquired in the wild, it is likely to contain a wide mixture of acoustic distortions e.g. background environmental noise, reverberation, as well as challenging speech artifacts such as children’s voices, laughs, sung speech, among others. The complexity in AfV audio increases when utterances can contain more than one speaker but no speaker turn marks are provided. The evaluation consisted in two conditions: a closed-set condition with training data limited to previous evaluations corpora plus a few databases to learn the AfV data, and an open-set condition in which the teams are allowed to use any data, public or not, to improve their performance.

In SRE18, the ViVoLab team worked on developing systems able to integrate phonetic information in the embedding extraction process. In order to better characterize speakers, our team developed three speaker recognition subsystems based on the extraction of an embedded representation per utterance, followed by a score estimation by means of a Probabilistic Linear Discriminant Analysis (PLDA) back-end.

The methods to extract speaker representations are:

- A system based on i-vectors [5], in which a phoneme-dependent background model is responsible for providing phonetic awareness in the extraction process.
- A solution based on the Wide Residual Network (WRN) architecture [6, 7] with Time-Delay Neural Networks (TDNN) [8] incorporating a Self-Attention Mechanism with exploitation of the phonetic knowledge.
- A system based on the well known x-vector embedding [9], obtained by means of the widely used Kaldi toolkit.

From now on, Section 2 provides a description of the methods we propose to extract the phonetic information. Section 3 describes the systems that were used to obtain the speaker embedings. Section 4 presents the results of the work and finally, Section 5 concludes the paper.

2. Extraction of the Phonetic information

The phonetic information in the systems submitted by our team to the NIST-SRE 2018 is obtained by means of the architecture depicted in Fig. 1. This network extract the phonetic information from an utterance in two different ways. The first one, used by the i-vector extractor, is provided in terms of posterior probabilities for each phonetic class in a frame by frame basis. On the other hand, the WRN-TDNN architecture with Self-Attention mechanisms use the phonetic information in the form of phonetic embeddings instead of posterior probabilities.

http://dx.doi.org/10.21437/Interspeech.2019-2417

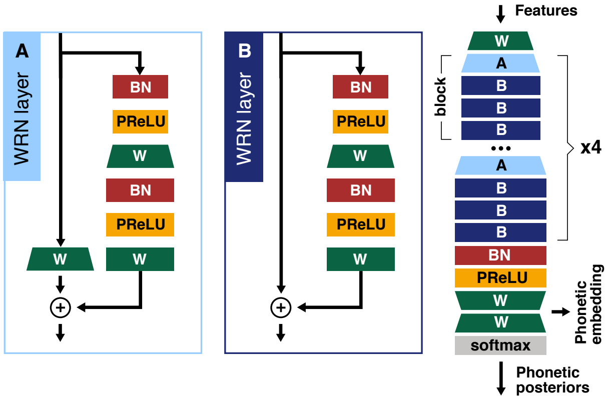

Figure 1: *WRN architecture used in the phoneme estimation. Left: WRN block type A; center: WRN block type B; right: Full WRN with phonetic posteriors and embeddings as output*

The proposed architecture combines Wide Residual Networks (WRN) and Time Delay Neural Networks (TDNN). This network makes use of 1D convolutions instead of the original 2D kernels. It is composed of four WRN blocks and, as nonlinearities, we use Parametric Rectified Linear Units (PReLU) [10]. Then, to produce the phonetic classes, the output of the fourth WRN block is connected to a Batch Normalization (BN) layer [11], a PReLU, a position-wise fully connected layer, and a softmax which finally produces the posterior probabilities. The phonetic embeddings are obtained from the input of the last linear layer of the network.

In order to increase the temporal context of the convolutional layers, we use a TDNN which is also equivalent to a convolutional 1D layer with dilation. Therefore, the configuration of our implementation (-9,-6,-3,0,+3,+6,+9) is equivalent to a 1D CNN with kernel size 7 and dilation 3. This approach can be also seen as the concatenation of shifted activation channels before each block, similarly to what is done in Shifted Delta Cepstral (SDC) feature extraction [12, 13].

The WRN-TDNN is trained using Fisher and Switchboard datasets aligned to phoneme labels using the Kaldi toolkit standard procedure for an HMM-GMM [14]. We have used the AdamW algorithm [15, 16] and data augmentation at training, by adding random room impulse responses [17], additive noises [18], and random scaling of the time axis at the feature level. The proposed system uses the log filter-bank outputs (size 32) plus the log energy as acoustic features.

3. Extraction of Speaker Embeddings

The primary ViVoLab system is the result of fusing three different pipelines, each one of them based on its own type of embedded speaker representation. Each pipeline implies its own front-end and extraction conditions, as well as its own backend.

3.1. Phonetic I-vectors

3.1.1. Acoustic Features

For this method, 20 Mel Frequency Cepstral Coefficients (MFCC) including C0 (C0-C19) over a 25 ms hamming window every 10 ms (15 ms overlap), and first and second order derivatives are computed over the feature vector sequence. Voice Activity Detection (VAD) is performed computing the Long-Term Spectral Divergence (LTSD) of the signal every 10 ms, and comparing it against a threshold as in [19, 20]. After frame selection, features are short-time Gaussianized with a 3-second window as in [21].

3.2. Phonetic i-vector extraction

In this method we propose the use of a phoneme dependent and Gender Independent (GI) Universal Background Models (UBM). The total number of Gaussian components of the model is 2496, 64 Gaussians per phoneme, considering 39 possible phonemes. These set of phoneme dependent GMMs were trained by means of Maximum Likelihood (ML) iterations. The posterior of each component is controlled by the phonetic information provided by the system described in 2, making a limited number of Gaussians responsible for the generation of a frame. This solution still maintains the flexibility of GMMs, by the use of multiple components to explain the different pronunciations of each sound. For this model, a cohort of the CTS data contained in SRE 2004-2010 databases and Switchboard were used.

On top of this UBM, we used a 600 dimension i-vector extractor trained on a selection of recordings contained in SRE 2004-2010, Switchboard, Fisher, and MIXER6 corpus. Utterances are reassured to contain at least 100 seconds of pure speech. Centering, whitening [22] and length normalization [23] were applied. When dealing with CTS data in Arabic Tunisian, adaptation in centering-whitening was carried out. Whilst the centering task is trained with SRE18 unlabeled subset, the whitening matrix has its training done according to SRE2004-2010, MIXER6, and Switchboard, reassuring to have at least 8 utterances per speaker.

3.3. WRN and Self attention models for the Extraction of speaker embeddings

3.3.1. Acoustic Features

The front end for this system uses the same window size and overlap as the one described in 3.1.1, but in this case, two log-filter bank outputs of sizes 24 and 32 are used, which are concatenated, plus the log energy, attempting to capture different resolutions.

3.3.2. Embedding extraction using WRN and Self Attention Models

In this system we also use a DNN to extract speaker embeddings as vectors from an audio utterance. The proposed architecture is composed of two different parts. The first part is similar to the phonetic decoder network described in Section 2. It makes use of two standard convolutional layers and two WRN-TDNN blocks with a context: (-6,-3,0,+3,+6), equivalent to a conv 1D with a kernel size 5 and dilation 3. The phonetic embeddings are concatenated across the channel axis to the network activation before each WRN-TDNN block and before the second part. This way, the integration of the phonetic information improves the performance of the attention mechanism. In general, the first part of the network can be seen as a deep feature extraction step conditioned to the phonetic information.

The second part of the network is inspired by the successful method for translation and text related tasks: “Attention is all you need” [24]. We have adapted some of the ideas of the encoder network in that work to the task of ASV by making an analogy with the i-vector extraction process.

The first step is to obtain Q attention weights for each frame

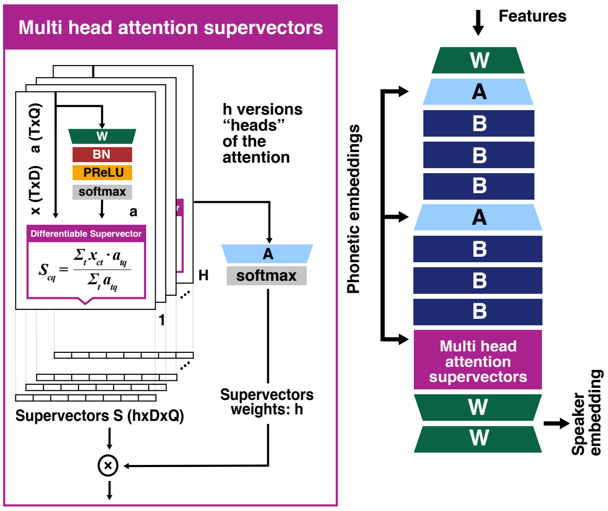

Figure 2: *Architecture used for embedding extraction based on attention models.*

by applying a linear projection, a PReLu and a softmax to the input of size $T \times D$, with D the number of channels provided by the previous part of the network. This can be seen as the equivalent of classifying all the frames to a number of classes in an unsupervised way . For example, we could consider them analogous to phonetic classes or Gaussians in a mixture, since we do not have a label for them. In the paper [24] this role is played by the dot product of key and query variables.

Next, we repeat this weight attention mechanism a number of times H, similarly to what is done in [24] with the multiple heads. The multiple heads in our case allow the network to focus on different groups of unsupervised classes. Then, we apply the H versions of the attention weights (of size $T \times Q$) to the original input sequence of size $(T \times D)$ using a matrix multiplication along the time axis and we divide by the count in each class to obtain the equivalent to H supervectors of size $Q \times D$ [25]. Finally, we reduce the H dimension to have a single supervector by computing a projection plus non-linearity and softmax in the H axis so that this term can act again as attention weights for the supervectors. Using matrix multiplication we reduce the head axis and we obtain a single supervector. Finally we use a small residual layer, similarly to [24] to produce the final output. In this work $H = 128$ and $Q = 8$.

Unlike the method presented in [24], the attention is distributed along the unsupervised class axis and not along the time axis. In addition to this, in previous proposals the attention mechanism was repeated several times, e.g. six times in the encoder but we have applied it twice, once to average the signal by meaningful groups, and the second to reduce the heads.

The network is trained using previous SRE data, Switchboard, and MIXER6 to classify speakers in the train set using the same optimizer and data augmentation as the phonetic networks. Embeddings were extracted before the last linear layer. See the following figure 2 for a graphical representation of the architecture used.

3.4. X-vector embeddings

The x-vector extraction from the speech was performed through the Kaldi recipe[1] and the available pre-trained model[2]. In the following, a brief description of the configuration used for the DNN x-vector extractor [9] is provided. Notice that, the extracted representations are also centered, whitened, and length-normalized, prior to the evaluation back-end.

Speech parametrization consists of 23-dimensional MFCC (C0-C22) over frames of 25ms every 10ms. Then, a short-time cepstral mean subtraction is applied over a 3-second sliding window. Finally, an energy-based VAD is used for dropping the non-speech frames. The x-vector extractor was built with a 7 hidden layer DNN (first 5 hidden layers operate at frame-level, last 2 operate at segment-level) and the non-linearities are Rectified Linear Units (ReLU) to discriminate among speakers. The statistics pooling layer between the frame-level and segment-level layers accumulates all frame-level outputs from the 5^{th} layer and computes the mean and standard deviation over all frames for an input segment. The trained model is used to obtain x-vectors, which are extracted from the concatenating the values at two affine layers [26], the 6^{th} and 7^{th} layers (512 + 512-dimensional x-vectors).

3.5. Back-ends

Each type of embedding is evaluated by its corresponding back-end using a gender independent standard simplified PLDA [27, 22]. This model is trained using excerpts from SRE2004-2010, Switchboard, and MIXER6. In order to guarantee proper within-speaker variability modeling, at least eight utterances per speaker were reassured during training.

All the speaker representations were post-processed for dimensionality reduction using Linear Discriminant Analysis (LDA), centered, whitened and length normalized [23]. To reduce the missmatch between training and eval dataset, the centering was carried out using the SRE18 unlabeled subset.

To deal with channel and language mismatch, the data used to evaluate the CTS condition, by means of the PLDA, was in-domain adapted using the SRE18 unlabeled subset. The adaptation considers each utterance to come from a single speaker, rather than performing unsupervised clustering.

3.6. Score normalization, calibration and fusion

We normalized the PLDA scores by adaptative S-normalization as follow:

$$s' = \frac{s - \mu_t}{\sigma_t} + \frac{s - \mu_e}{\sigma_e} \qquad (1)$$

where the mean μ_t and standard deviation σ_t, are computed on the scores of the cohort versus the test segments; and μ_e and σ_e are computed on the scores of the enrollment segments versus the cohort. For CTS type of audio, we used SRE18 unlabeled set as cohort, and for AfV type of audio we employed SITW. The score normalization was adaptive, i.e. the normalization parameters (μ_t, σ_t,μ_e and σ_e) are calculated only considering the 25% most restrictive scores.

ViVoLab's submission included scores from three different systems: a primary proposal fusing scores from the three systems describe above, and two contrastive systems, namely the x-vector based system as contrastive 1, and the phonetic i-vector system as contrastive 2.

[1] https://github.com/kaldi-asr/kaldi/tree/master/egs/sre16/v2
[2] http://kaldi-asr.org/models.html

Table 1: *Comparative results between a phonetic i-vector system and a traditional i-vector system on the evaluation set.*

System	EER (%)	minC	actC
CMN2			
No phonetic info	11.98	0.713	0.715
With Phonetic info	11.12	0.676	0.678
VAST			
No phonetic info	15.87	0.662	0.724
With Phonetic info	17.85	0.622	0.657
Overall results for both CMN2 & VAST			
No phonetic info	-	-	0.719
With Phonetic info	-	-	0.667

Table 2: *Results on the evaluation set for CMN2 (PSTN & VoIP), and VAST (AfV) using the NIST scoring tool.*

System	EER (%)	minC	actC
CMN2			
Primary	7.63	0.512	0.512
Contrastive 1	8.60	0.595	0.598
Contrastive 2	10.74	0.671	0.674
VAST			
Primary	14.60	0.574	0.631
Contrastive 1	15.24	0.708	0.716
Contrastive 2	18.41	0.605	0.649
Overall results for both CMN2 & VAST			
Primary	-	-	0.572
Contrastive 1	-	-	0.657
Contrastive 2	-	-	0.662

Calibration and fusion of the three systems were trained using the development SRE18 trial list using linear logistic regression. Both calibration and fusion are gender independent. Among the three different subsets in SRE18, namely PSTN, VoIP, and AfV. Both, PSTN and VoIP, were jointly calibrated and fused despite scored in two different evaluation points. They were adjusted to work at VoIP calibration point, the more restrictive one. Regarding VAST data, the low amount of trials for development made us not rely on the official operating point, tuning in a more relaxed condition ($P_{target} = 0.1$) in order to obtain a more reliable estimation of the error rates.

4. Results

In order to illustrate the influence of the phonetic information on the results of the ASV system, we conducted a limited experiment considering the i-vector approach. As we can see on Table 1, except for the EER in the AfV subset, the phonetic i-vectors outperforms traditional i-vectors for all the considered metrics. The overall relative improvement is around 8$\%$, and reaches up to 10$\%$ for the actual cost in the VAST subset.

As mentioned before, the final submission of the Vivolab Team for the NIST SRE 2018 evaluation consisted on three different systems: A fusion of the three different embeddings described in section 3.3 (primary), a system based on the x-vector embeddings (contrastive 1), and a system based on phonetic i-vectors (contrastive 2). Table 2 presents the performance of the submitted systems on the evaluation sets CMN2, which includes the PSTN & VoIP, and VAST, the AfV data.

According to results, the x-vector based system seems to work slightly better than the phonetic i-vector system in terms of the overall performance. The inclusion of the WRN-TDNN phonetic embeddings also improve the global performance of the system. Nevertheless, if we analyze in detail the results of the different systems for the different data subsets in eval, we can see that the x-vector system obtains lower cost for the CMN2 subset while the phonetic i-vector systems outperform the x-vector systems for the VAST subset. On the other hand, the phonetic i-vector contribution is quite relevant when fused with the x-vector and the attention embedding, obtaining the best result in CTS type of audio. The other way, note that the improvement due to fusion is not that significant when i-vectors performance overcome x-vectors as we can see, for the VAST dataset. Furthermore, for the AfV data, the calibration solution worked relatively well despite the fact that the operating point we calibrated for is far from the operating point proposed in the evaluation.

5. Conclusions

This paper presented the ViVoLab submission to the NIST-SRE 2018. As primary submission, a fusion of three ASV systems was presented including i-vector system based on phonetic modeling, x-vector system following the Kaldi recipe, and a WRN-TDNN solution based on self attention models and phonetic embeddings.

For CTS data, results showed better performance for the x-vector based system than the phonetic i-vector technology. Note that this sort of data is closer to training conditions than the alternative AfV. Nevertheless, when moving to AfV, the phonetic i-vector based system actually worked better than the x-vector embeddings.

It is also noticeable the usefulness of score fusion in this context. All embeddings representation actually were complemented together for a more substantial contribution to the system performance. Although, the fusion achieved a much more significant improvement when performance was led by x-vectors rather than i-vectors.

6. Acknowledgements

This work has been supported by the Spanish Ministry of Economy and Competitiveness and the European Social Fund through the project TIN2017-85854-C4-1-R, Government of Aragón (Reference Group T36_17R) and co-financed with Feder 2014-2020 "Building Europe from Aragón". We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. This material is based upon work supported by Google Cloud.

7. References

[1] I. Viñals, A. Ortega, A. Miguel, and E. Lleida, "Phonetic variability influence on short utterances in speaker verification," in *Proc. IBERSPEECH 2018*, 2018, pp. 6–9.

[2] Y. Liu, L. He, J. Liu, and M. T. Johnson, "Speaker embedding extraction with phonetic information," in *Proc. Interspeech 2018*, 2018, pp. 2247–2251. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1226

[3] S. O. Sadjad, C. S. Greenberg, D. A. R. E. Singer, L. P. Mason, and J. Hernandez-Cordero, "The 2018 NIST speaker recognition evaluation," in *Proc. INTERSPEECH 2019 (submitted)*, 2019.

[4] S. O. Sadjad, T. Kheyrkhah, A. Tong, C. S. Greenberg, D. A. Reynolds, E. Singer, L. P. Mason, and J. Hernandez-Cordero, "The 2016 NIST speaker recognition evaluation," in *Proc. INTERSPEECH 2017*, 2017, pp. 1353–1357.

[5] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end Factor Analysis for Speaker Verification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[7] S. Zagoruyko and N. Komodakis, "Wide residual networks," *CoRR*, vol. abs/1605.07146, 2016. [Online]. Available: http://arxiv.org/abs/1605.07146

[8] A. H. Waibel, T. Hanazawa, G. E. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 37, no. 3, pp. 328–339, 1989. [Online]. Available: https://doi.org/10.1109/29.21701

[9] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep Neural Network Embeddings for Text-Independent Speaker Verification," in *Proceedings of the annual conference of the International Speech Communication Association, Interspeech 2017*, 2017, pp. 999–1003.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[11] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

[12] J. R. Calvo, R. Fernández, and G. Hernández, "Application of shifted delta cepstral features in speaker verification," in *Eighth Annual Conference of the International Speech Communication Association*, 2007.

[13] D. Ribas and J. Calvo, "Speaker verification with shifted delta cepstral features: its pseudo-prosodic behavior," in *SLTECH-2009*, 2009.

[14] D. Povey, A. Ghoshal, G. Boulianne, N. Goel, M. Hannemann, Y. Qian, P. Schwarz, and G. Stemmer, "The kaldi speech recognition toolkit," in *In IEEE 2011 workshop*, 2011.

[15] D. P. Kingma and J. L. Ba, "Adam: Amethod for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

[16] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in adam," *arXiv preprint arXiv:1711.05101*, 2017.

[17] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.

[18] D. Snyder, G. Chen, and D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," 2015, arXiv:1510.08484v1.

[19] J. Ramirez, J. C. Segura, C. Benitez, A. de la Torre, and A. Rubio, "Voice activity detection with noise reduction and long-term spectral divergence estimation," in *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing, ICASSP 2004*, vol. 2, 2004, pp. 1093–1096.

[20] ——, "Efficient voice activity detection algorithms using long-term speech information," *Speech Communication*, vol. 42, no. 3, pp. 271 – 287, 2004.

[21] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *Proceedings of Odyssey 2001. The Speaker and Language Recognition Workshop*, 2004.

[22] J. Villalba and E. Lleida, "Handling i-vectors from different recording conditions using multi-channel simplified plda in speaker recognition," in *Proceedings of IEEE International Conference on Acoustic, Speech and Signal Processing, ICASSP 2013*, 2013.

[23] D. Garcia-Romero and C. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Proceedings of the annual conference of the International Speech Communication Association, Interspeech 2011*, 2011, pp. 249–252.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[25] V. Mingote, A. Miguel, A. Ortega, and E. Lleida, "Differentiable supervector extraction for encoding speaker and phrase information in text dependent speaker verification," in *Proc. IberSPEECH 2018*, 2018, pp. 1–5. [Online]. Available: http://dx.doi.org/10.21437/IberSPEECH.2018-1

[26] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proceedings of IEEE International Conference on Acoustic, Speech and Signal Processing, ICASSP 2018*, 2018.

[27] S. J. D. Prince and J. H. Elder, "Probabilistic Linear Discriminant Analysis for Inferences About Identity," in *Proceedings of the IEEE International Conference on Computer Vision*, 2007.