Interspeech 2018
2-6 September 2018, Hyderabad

An Unsupervised Neural Prediction Framework for Learning Speaker Embeddings using Recurrent Neural Networks

Arindam Jati, Panayiotis Georgiou

University of Southern California, Los Angeles, CA, USA

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

Abstract

This paper presents an unsupervised training framework for learning a speaker-specific embedding using a Neural Predictive Coding (NPC) technique. We employ a Recurrent Neural Network (RNN) trained on unlabeled audio with multiple and unknown speaker change points. We assume short-term speaker stationarity and hence that speech frames in close temporal proximity originated from a single speaker. In contrast, two random short speech segments from different audio streams are assumed to originate from two different speakers. Based on this hypothesis, a binary classification scenario of predicting whether an input pair of short speech segments comes from the same speaker or not, is developed. An RNN based deep siamese network is trained and the resulting embeddings, extracted from a hidden layer representation of the network, are employed as speaker embeddings. The experimental results on speaker change points detection show the efficacy of the proposed method to learn short-term speaker-specific features. We also show the consistency of these features via a simple statistics-based utterance-level speaker classification task. The proposed method outperforms the MFCC baseline for speaker change detection, and both MFCC and i-vector baselines for speaker classification.

Index Terms: unsupervised learning, recurrent neural networks, speaker segmentation, speaker classification, text-independent speaker recognition

1. Introduction

Speech signal conveys a multitude of different information including phonetic details, speaker and channel characteristics, short-term emotions and sentiments, and even long-term behavioral cues [1]. Extracting speaker-specific characteristics from speech plays a vital role in numerous applications like speaker recognition [2], speech recognition [3], and speaker segmentation (or speaker change points detection) and diarization [4].

Generally, short-term [2] acoustic features, like MFCC [5] and PLP [6], encode diverse information and are *not* constrained to retain only speaker-specific characteristics. Yet, they are the foundations of many state-of-the-art speaker segmentation, diarization and recognition systems. The essential trick in majority of these applications is to employ the short-term features and exploit temporal context to create speaker models [2]. One widely used method is to pose some mathematical assumption on the probability distribution of short-term features (*e.g.,* Gaussian) and, based on that, derive fixed dimensionality speaker-specific feature vectors from utterances of variable durations. For example, Reynolds *et al.* [7] proposed a speaker verification technique by training a GMM-UBM [7] and utilizing the concatenated means of the MAP adapted (on a speaker's data) GMM (known as GMM supervector [2]) as fixed dimensional speaker-dependent vector. Based on this foundation, a range of factor analysis methods [8, 9] came into the picture for separating speaker- and channel-dependent latent variabilities from the audio to obtain better speaker models. Later, Dehak *et al.*. proposed i-vectors [10] by employing a single Total Variability Matrix for modeling both channel and speaker variabilities. Although i-vectors are still considered state-of-the-art features, the GMM assumption [2] and their deterioration in performance for mismatched train and test utterance durations [11] are two major downsides of i-vectors as reported in literature.

Recent advancements in Deep Neural Network (DNN) [12] research have attracted speech scientists to utilize the distinctive ability of DNNs to learn and extract speaker-specific features from audio. The most common trend is to use some loss function that discriminates between speakers and extract one or more *meaningful* hidden layer representations, generally known as "speaker embeddings", which are then used as speaker-specific features. For example, [13] trained a speaker classification network and utilized the speaker embeddings for diarization. [14] proposed a supervised training scheme through comparing if two input speech frames were originated from the same speaker or not. Garcia *et al.* [15] adopted a similar approach but their DNN had a temporal pooling layer to handle variable length utterances. Snyder *et al.* [16, 17] achieved state-of-the-art performance in speaker verification with a DNN trained for a speaker classification task. Recent work [18] compared different architectures deploying Convolutional Neural Networks (CNNs) and RNNs for speaker verification using triplet loss.

The common drawback of the above methods is that they all need labeled data for supervised learning. This might inhibit the performance of these methods in case of scarcity of available labeled data, and in that scenario, these methods might fail to perform well if the test environment is quite different from the training conditions. This creates the need for developing scalable unsupervised methods for learning speaker embeddings. Some work [19, 20] has been done in the past for clustering speakers' space using DNNs, and deploying the clustered space as DNN-UBM to replace conventional GMM-UBM, but they are only applicable for speaker recognition and they do not learn short-term speaker-specific features. Lee *et al.* [21] trained a DNN for unsupervised speaker classification, but they trained on TIMIT dataset [22] where the training utterances do not have multiple speakers, and only in-domain evaluation was done.

Recently we proposed Speaker2Vec [23], where a dense DNN was trained in unsupervised setting to learn speaker-specific characteristics. In our recent followup work [24] we proposed the general idea of Neural Predictive Coding (NPC). It assumes two temporally close short speech segments to belong to a single speaker, and thus a unique transformation that can encode both the segments, could project them to a high dimensional manifold where they come in closer proximity to each other than they do in the original MFCC feature space. This unique transformation is learned through training the DNN. In this paper, we build on this idea by utilizing the temporal memory of a RNN to learn speaker embeddings from unlabeled data.

10.21437/Interspeech.2018-1363

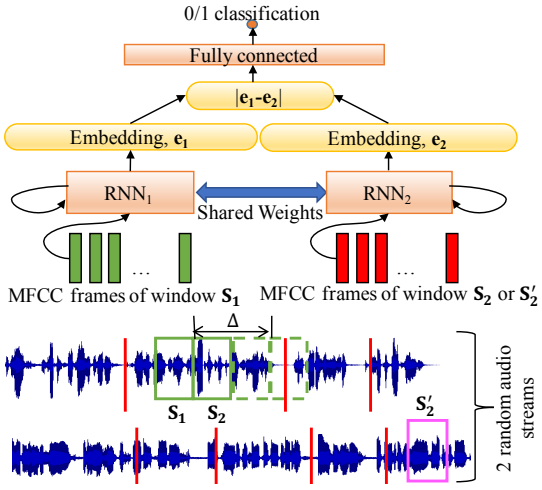

Figure 1: *The NPC-RNN unsupervised training framework as explained in Section 2.*

The contributions of this work are the following: *a*) It involves unsupervised training, and therefore it can be trained on real life data where the audio streams have multiple speakers with unknown speaker change points. This makes it highly scalable. *b*) It learns short-term speaker embeddings that can be useful for applications like speaker segmentation (Section 4.2). Moreover, simple statistics of the embeddings over the whole utterance can produce utterance-level fixed dimensional embedding useful for applications like speaker identification (Section 4.4). *c*) The Gated Recurrent Units (GRUs) [25] exploits the sequential information in MFCC frames, and helps us learn the speaker embeddings faster than CNNs employed in [24] even with fewer parameters. *d*) The deployment of a siamese network (Section 2.2) reduces the number of parameters compared to the dense architecture in [23]. Moreover, it paves a direct way of applying the model for speaker segmentation and does not need fitting any distribution and additional divergence computation between two segments (Section 4.2).

2. Methodology

NPC [24] is inspired from the idea of Linear Predictive Coding [26]. In LPC, the future value of a signal is modeled as a linear function (expressed by the filter coefficients) of the past values. In NPC, the future speech frames are described by a nonlinear function of the past frames, and the function is learned through training the DNN on large amounts of unlabeled speech.

2.1. The unsupervised training scheme

The first question that comes up in mind is how this local encoding of information is useful for learning speaker embeddings. Here we hypothesize short-term active speaker stationarity [23, 24]. This is based on the simple observation that in a long, natural conversation any random pair of consecutive short speech segments is highly-likely to belong to the same speaker. In other words, speaker changes generally do not occur very fast in realistic interactions. Although there are some pairs that contain speech from two different speakers, we assume that in a large and diverse dataset this probability is low. On the other hand, if we randomly choose two short speech segments from two different audio streams (*e.g.,* two random videos from YouTube), then the probability that those belong to the same speaker is also extremely small. In this way, we can create two sets of samples from our unlabeled dataset. The first set contains "genuine pairs" [27] ($(\mathbf{S_1}, \mathbf{S_2})$ in Fig. 1): two consecutive short speech segments, probabilistically originating from a single speaker. The second set contains "impostor pairs" [27] ($(\mathbf{S_1}, \mathbf{S_2'})$ in Fig. 1): two short speech segments randomly chosen from two random audio streams in the dataset, probabilistically originating from two different speakers. These samples are applied to a siamese neural network [27] for binary classification. The training framework is explained in Fig. 1 along with the sketch of the NPC-RNN siamese network. There are two major benefits of training in this discriminative style (instead of our previous approach [23] of encoding one segment and decoding it towards obtaining the next segment). First, the model encounters negative samples (impostor pairs) as well as positive samples (genuine pairs). Second, we are not optimizing for exact reconstruction but to teach the DNN if the two segments originated from the same speaker.

2.2. The NPC-RNN siamese network

The siamese network [27, 28, 29] has two identical twin networks (RNN_1 and RNN_2 in Fig. 1) whose weights are shared. It is generally trained in genuine/impostor pair classification scenario using a discriminative energy function between the outputs of the two twin networks. In Fig. 1, each RNN block denotes a multi-layered GRU network. We use GRU instead of LSTM because they require fewer parameters, but are generally found to achieve similar performance in several applications [30]. All the d input frames of the first segment ($\mathbf{S_1}$) are provided as a temporal sequence of vectors to the GRU. The last (temporally) hidden state of the last layer is connected to the D dimensional "embedding layer" (to produce transformation $\mathbf{e_1}$) through a fully connected network. The second segment ($\mathbf{S_2}$ or $\mathbf{S_2'}$), also having d frames, is transformed similarly to embedding $\mathbf{e_2}$. Note that the genuine pair of windows ($\mathbf{S_1}$, $\mathbf{S_2}$) is moved by Δ frames to generate other genuine pairs from that audio stream. The $\mathbf{L_1}$ distance vector between $\mathbf{e_1}$ and $\mathbf{e_2}$ is then calculated as the absolute differences between the elements (inspired from [29]):

$$\mathbf{L_1} = |\mathbf{e_1} - \mathbf{e_2}| \tag{1}$$

So, $\mathbf{L_1}$ is also D-dimensional. Now, $\mathbf{L_1}$ is connected to the final single output through a fully connected layer. The final output has sigmoid nonlinearity to predict whether the input pair is genuine (0) or impostor (1). If the input pair of segments is $(\mathbf{S_1}, \mathbf{S_2})$, the the final output of the model, $p(\mathbf{S_1}, \mathbf{S_2})$ denotes the probability of $\mathbf{S_1}$ and $\mathbf{S_2}$ to be from different speakers. A binary cross-entropy loss is used for optimization.

2.3. Evaluating NPC-RNN model

The NPC-RNN model can be utilized in two ways.

2.3.1. Speaker change points detection

First, we can use the full model, and compare two input segments (each of duration $1s$) to predict the probability of whether they are from the same speaker or not. This can be useful for applications like speaker comparison [14] and speaker change point detection (Section 4.2). We can move a sliding window pair $(\mathbf{S_1}(t), \mathbf{S_2}(t))$ centered at time t over the audio stream, and get a probability curve $p(\mathbf{S_1}(t), \mathbf{S_2}(t))$. The higher the probability at any time, the higher the chance that it corresponds to a speaker change point. This speaker segmentation method also eliminates the Gaussianity assumption generally used in segmentation algorithms.

2.3.2. Extracting NPC-RNN speaker embeddings

The second application utilizes only one of the siamese twins, up to the embedding layer, to extract a D dimensional embedding from any $1s$ speech segment. We can use a $1s$ moving window over the test audio stream and move it by 1 frame to create a sequence of embeddings. This can be employed in appli-

cations like utterance-level speaker classification (Section 4.4). For example, statistical functionals of the embeddings over the whole utterance can be used as a fixed length representation of a variable length utterance, or higher-layers of machine learning methods can employ these as features.

3. Experimental Setup

3.1. Features and model parameters

We use 40 dimensional high definition MFCC features computed with a sliding window of 25*ms* width and 10*ms* shift using the Kaldi toolbox [31]. We use the training segment size, $d = 1s = 100$ MFCC frames, and the shift, $\Delta = 2s = 200$ frames. Each of the siamese twins (RNN_1 or RNN_2) has 3 GRU layers with 200 hidden units in every layer. The embedding dimension, $D = 512$. Thus, the last hidden state of the last layer of GRU is attached to the embedding layer through a 200×512 fully connected layer. The model has one batch normalization layer after the embedding layer, *i.e.,* before going to the sigmoid output. The entire model has around 732k trainable parameters. We train the model on two NVIDIA K40 GPUs. We employ RMSProp optimizer [32] with a learning rate of 10^{-4} and l_2 regularization factor of 10^{-6}.

3.2. Training datasets

We train our model on two different datasets. The first one is trained on the TED-LIUM train dataset [33]. This was originally developed for speech recognition purposes. It has 666 speakers, but we rejected 19 speakers which are also present in TED-LIUM dev (7 speakers) and test (11 speakers) sets. After that we end up with a dataset of around 100 hours of speech. We create around 358k samples from that, considering both the genuine and impostor pairs. Since, every TED-LIUM session mainly contains one speaker, this dataset does not accurately validate our short-term speaker stationarity hypothesis. Yet, we use this dataset to compare in-domain and out-of-domain training scenarios (Section 4.1). We have also collected random videos from YouTube to prepare a more realistic unlabeled dataset, YoUSCTube, having approximately 584 hours of audio, resulting in 2.1M training samples. The YoUSCTube dataset is prepared in an unsupervised way starting from some initial random video, and then randomly selecting multiple videos from the list of automated-YouTube recommendations, and continue the process for as long as we like. In this case we did that until we collected the 584 hours of unique material. A brief inspection of the dataset has revealed that it contains both clean and noisy, single- and multi-speaker speech from diverse languages and environments. For both the datasets we have created equal number of genuine and impostor pairs.

3.3. Validation and test datasets

We use the TED-LIUM dev set as the validation set during training for model selection. The model with the best validation accuracy is selected. We utilized the utterance start and end timings provided in the transcripts so that the validation is error free. Here, we should keep in mind that the training sets are noisy because of our assumption of short-term speaker stationarity. The TED-LIUM test set (no speaker overlap with training or validation sets) is employed for testing the NPC-RNN embeddings in different applications as described in Section 4.

4. Results and Discussions

The NPC-RNN models are evaluated on speaker change point detection and utterance-level speaker classification tasks. The performance on the speaker change point detection task relates to the usefulness of the NPC-RNN embeddings as frame level features. The statistics-based speaker classification experiment validates the consistency of the embeddings over the whole utterance. In both cases, these are meant to show the relative effectiveness of the proposed embeddings and can provide better results through higher-layers of trainable machine learning systems.

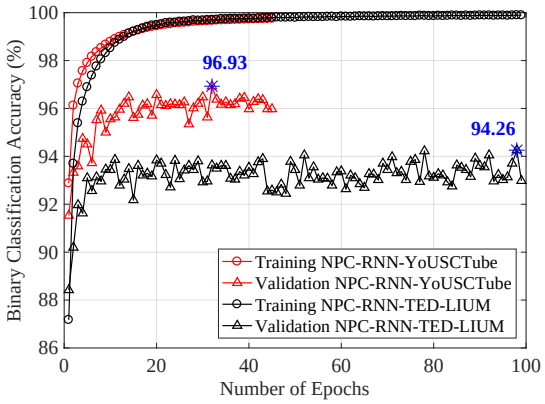

Figure 2: *NPC-RNN training and validation accuracies for YoUSCTube and TED-LIUM datasets. The asterisks denote the best obtained validation accuracies.*

4.1. In-domain and out-of-domain (OOD) training

In Fig. 2, we compare the training and validation accuracies for the two training datasets: YoUSCTube and TED-LIUM. From Section 3.3 we can infer that TED-LIUM encounters in-domain training, while YoUSCTube training is out-of-domain. Moreover, as explained in Section 3.2, YoUSCTube dataset is more realistic to validate the short-term speaker stationarity hypothesis. As we can see in Fig. 2, in both datasets, we reach almost 100% training accuracy. We get maximum validation accuracies of 94.26% and 96.93% for training on TED-LIUM and YoUSCTube datasets respectively, even though the latter one employs an out-of-domain validation set. We believe that the high validation accuracy on YoUSCTube supports both the short-term stationarity hypothesis and the benefits of unsupervised training on large amounts of diverse and unlabeled data. In the subsequent sections, we will only report results obtained using the NPC-RNN-YoUSCTube model.

4.2. Frame-level: Speaker change point detection

We apply the NPC-RNN-YoUSCTube model for the speaker change point detection application. We create an artificial dialog using randomly chosen audio segments of duration between $1s$ to $3s$ from the TED-LIUM test dataset. It has a total of 200 speaker change points. The baseline is the widely used speaker change detection algorithm [34] based on BIC metric. Table 1 shows the results. We adopt two evaluation metrics. The first one is the standard F1 score based on the harmonic mean of precision and recall [35]. The second one involves newly introduced coverage and purity metrics [35]. Higher values are better for both the metrics.

The BIC based algorithm requires a predefined window length for estimating the parameters of a Gaussian distribution. This window length (w in Table 1) has been varied in our experiments, and the different cases are shown in Table 1. In each case, we vary the BIC threshold and report the best F1 score and corresponding coverage and purity values.

For, the NPC-RNN the window size is always $1s$, same as the training segment size. We use a threshold value, p_τ. If the probability value at a certain time, $p(\mathbf{S_1}(t), \mathbf{S_2}(t)) > p_\tau$, then t is predicted to be a speaker change point. We vary p_τ and report the one with best F1 score, and the corresponding coverage

Table 1: *Speaker change detection results on a test dialog having 200 speaker change points*

Metric	BIC ($w = 0.5s$)	BIC ($w = 1s$)	BIC ($w = 2s$)	BIC ($w = 2.5s$)	BIC ($w = 3s$)	NPC-RNN
F1 Score	0.58	0.74	0.70	0.70	0.68	**0.85**
(Coverage, Purity)	(0.81, 0.73)	(0.87, 0.78)	(0.92, 0.75)	(0.91, 0.74)	(0.92, 0.73)	**(0.88, 0.86)**

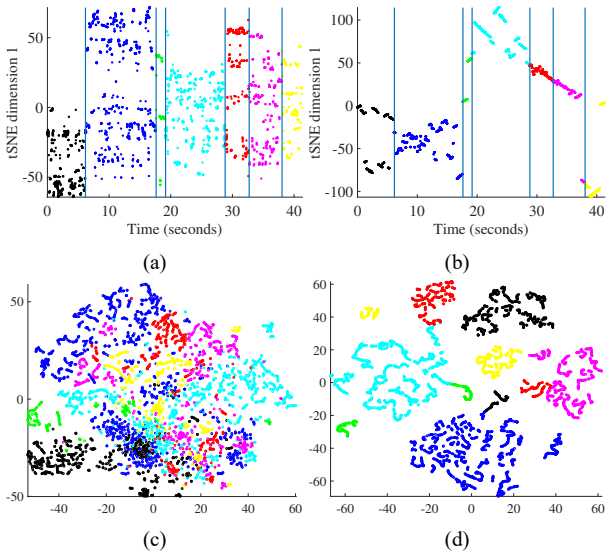

Figure 3: *tSNE visualizations of raw MFCC features and NPC-RNN-YoUSCTube embeddings for an artificial dialog containing 7 speakers (7 colors) and 6 change points (blue vertical lines).* ***[(a), (b)]*** *1D tSNE with time axis;* ***[(c), (d)]*** *2D tSNE without time axis.* ***[(a), (c)]*** *Raw MFCC features;* ***[(b), (d)]*** *NPC embeddings.*

and purity values. We can see from Table 1 that the NPC-RNN model based segmentation outperforms the baseline by a large margin (11% absolute improvement in F1 score). Moreover, it achieves better segment purity for all conditions, and better coverage than BIC with $w = 1s$ condition.

4.3. Frame-level: Embedding visualization

Fig. 3 shows the tSNE [36] visualizations of the *frame-level* MFCC features and the NPC embeddings for an artificial dialog generated by picking random segments from 7 speakers in TED-LIUM test dataset (1 segment per speaker). Fig. 3a and 3b plot the 1D tSNE versus time. Different colors represent different speakers. We can see that as time proceeds the active speaker changes. The MFCC features are scattered, while the NPC embeddings are much more compact in representing the speakers. Fig. 3c and 3d show the 2D tSNE plots without the time axis for MFCC and NPC respectively. The NPC embeddings form clear clusters within the same speaker, while the MFCC features are not able to form well separated clusters (for example, see the blue points). This shows the efficacy of the NPC models to generate embeddings with reduced within-speaker and increased across-speaker variability.[1]

4.4. Utterance-level: Speaker classification

We perform utterance-level speaker classification on the utterances (extracted using the provided manual timestamps) of TED-LIUM test set containing 11 unique speakers. We use kNN classifier with $k = 1$ for this purpose so that the simplicity of the classifier lets us properly asses the strength of different features. We compare the performance of raw MFCC features, i-vectors and the NPC-RNN-YoUSCTube embeddings. The i-vector system is trained OOD (note that the NPC model is also trained OOD) on Fisher English dataset [37]. Table 2 compares the performances of different features with different number of enrollment [2] utterances. For a given number of enrollment utterances, n, we randomly hold out 5 utterances per speaker for testing, and n other utterances are randomly selected (as enrollment utterances) to train the kNN classifier. In each case the same enrollment and test utterances are picked for all different features. The whole process is repeated 5 times (therefore, $11 \times 5 \times 5 = 275$ random test utterances for each of the 6 different enrollment scenarios) and the average accuracies are reported. We can see that NPC-RNN consistently outperforms others. Moreover, when concatenated with i-vectors, sometimes it gives complimentary information to i-vectors.

Table 2: *Utterance-level speaker classification accuracies (%) on TED-LIUM test set using 1-NN classifier*

Number of enrollment utterances, n	MFCC	i-vector	NPC-RNN-YoUSCTube	i-vector + NPC-RNN-YoUSCTube
1	73.82	78.54	83.64	**86.18**
2	89.46	88.73	94.91	**96.73**
3	90.91	89.45	**97.82**	96.73
5	94.91	92.73	98.54	**98.55**
8	94.55	94.55	**99.64**	98.91
10	97.45	95.64	**100.00**	99.27

5. Conclusions and Future Directions

We introduced an unsupervised training scheme for learning speaker embeddings from unlabeled data that might contain multi-speaker audio streams with unknown speaker change points. The framework is based on the proposed short-term speaker stationarity hypothesis which lets us build a contrastive loss based binary classification scenario even with an unlabeled dataset. The experimental results on speaker change points detection, frame-level visualization of the embeddings and the performance on utterance-level speaker classification task show the validity and potency of the method.

In the future we will build an end-to-end RNN-based unsupervised framework to produce fixed dimensional embeddings for variable length utterances, so the need of statistics over the utterance will be eliminated. We believe this will both simplify the framework and improve performance.

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.

[1] Note that the $1s$ moving window creates some overlap around the speaker change points. It can be seen in Fig. 3d that small part of a cluster is corrupting the cluster of the next speaker. But, this neither affects speaker segmentation since maximum discrimination will be at the actual change point, nor deteriorates utterance-level classification since there every utterance comes from a single speaker. One challenging case, for all algorithms, is when the whole segment is shorter than the window length, as is the case of the green cluster.

7. References

[1] S. Narayanan and P. G. Georgiou, "Behavioral signal processing: Deriving human behavioral informatics from speech and language," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1203–1233, 2013.

[2] J. H. Hansen and T. Hasan, "Speaker recognition by machines and humans: A tutorial review," *IEEE Signal processing magazine*, vol. 32, no. 6, pp. 74–99, 2015.

[3] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors." in *ASRU*, 2013, pp. 55–59.

[4] 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.

[5] 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.

[6] 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.

[7] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[8] P. Kenny, M. Mihoubi, and P. Dumouchel, "New map estimators for speaker recognition." in *INTERSPEECH*, 2003.

[9] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1435–1447, 2007.

[10] N. Dehak, P. J. 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.

[11] A. Kanagasundaram, R. Vogt, D. B. Dean, S. Sridharan, and M. W. Mason, "I-vector based speaker recognition on short utterances," in *Proceedings of the 12th Annual Conference of the International Speech Communication Association*. International Speech Communication Association (ISCA), 2011, pp. 2341–2344.

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

[13] 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.

[14] 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.

[15] D. Garcia-Romero, D. Snyder, G. Sell, D. Povey, and A. McCree, "Speaker diarization using deep neural network embeddings," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4930–4934.

[16] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," *Proc. Interspeech 2017*, pp. 999–1003, 2017.

[17] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," *ICASSP, Calgary*, 2018.

[18] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *arXiv preprint arXiv:1705.02304*, 2017.

[19] M. M. Saleem and J. H. Hansen, "A discriminative unsupervised method for speaker recognition using deep learning," in *Machine Learning for Signal Processing (MLSP), 2016 IEEE 26th International Workshop on*. IEEE, 2016, pp. 1–5.

[20] X.-L. Zhang, "Multilayer bootstrap network for unsupervised speaker recognition," *arXiv preprint arXiv:1509.06095*, 2015.

[21] 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.

[22] 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. [Online]. Available: https://catalog.ldc.upenn.edu/LDC93S1

[23] A. Jati and P. Georgiou, "Speaker2vec: Unsupervised learning and adaptation of a speaker manifold using deep neural networks with an evaluation on speaker segmentation," in *Proceedings of Interspeech*, August 2017.

[24] ——, "Neural predictive coding using convolutional neural networks towards unsupervised learning of speaker characteristics," *IEEE Trans. Speech, Audio, and Language Processing*, 2018, under review; arXiv:1802.07860.

[25] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[26] D. O'Shaughnessy, "Linear predictive coding," *IEEE potentials*, vol. 7, no. 1, pp. 29–32, 1988.

[27] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 539–546.

[28] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, vol. 2. IEEE, 2006, pp. 1735–1742.

[29] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML Deep Learning Workshop*, vol. 2, 2015.

[30] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[31] 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.

[32] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.

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

[34] S. Chen, P. Gopalakrishnan *et al.*, "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.

[35] H. Bredin, "pyannote.metrics: a toolkit for reproducible evaluation, diagnostic, and error analysis of speaker diarization systems," in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association*, Stockholm, Sweden, August 2017. [Online]. Available: http://pyannote.github.io/pyannote-metrics

[36] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[37] "Fisher english training speech part 1 speech." [Online]. Available: https://catalog.ldc.upenn.edu/LDC2004S13