Interspeech 2018
2-6 September 2018, Hyderabad

Triplet Network with Attention for Speaker Diarization

Huan Song[1] *, Megan Willi*[2]*, Jayaraman J. Thiagarajan*[3]*, Visar Berisha*[1,2]*, Andreas Spanias*[1]

[1]SenSIP Center, School of ECEE, Arizona State University, Tempe, AZ
[2]Department of Speech and Hearing Science, Arizona State University, Tempe, AZ
[3]Lawrence Livermore National Labs, 7000 East Avenue, Livermore, CA

{huan.song, megan.willi, visar, spanias}@asu.edu, jjayaram@llnl.gov

Abstract

In automatic speech processing systems, speaker diarization is a crucial front-end component to separate segments from different speakers. Inspired by the recent success of deep neural networks (DNNs) in semantic inferencing, triplet loss-based architectures have been successfully used for this problem. However, existing work utilizes conventional i-vectors as the input representation and builds simple fully connected networks for metric learning, thus not fully leveraging the modeling power of DNN architectures. This paper investigates the importance of learning effective representations from the sequences directly in metric learning pipelines for speaker diarization. More specifically, we propose to employ attention models to learn embeddings and the metric jointly in an end-to-end fashion. Experiments are conducted on the CALLHOME conversational speech corpus. The diarization results demonstrate that, besides providing a unified model, the proposed approach achieves improved performance when compared against existing approaches.

Index Terms: speaker diarization, triplet network, metric learning, attention models

1. Introduction

With the ever-increasing volume of multimedia content on the Internet, there is a crucial need for tools that can automatically index and organize the content. In particular, speaker diarization deals with the problem of indexing speakers in a collection of recordings, without *a priori* knowledge about the speaker identities. In scenarios where the single-speaker assumption of recognition systems is violated, it is critical to first separate speech segments from different speakers prior to downstream processing. Typical challenges in speaker diarization include the need to deal with similarities between a large set of speakers, differences in acoustic conditions, and the adaptation of a trained system to new speaker sets.

An important class of diarization approaches rely on extracting i-vectors to represent speech segments, and then scoring similarities between i-vectors using pre-defined similarity metrics (e.g. cosine distance) to achieve speaker discrimination. Despite its widespread use, it is well known that the i-vector extraction process requires extensive training of a Gaussian Mixture Model based Universal Background Model (GMM-UBM) and estimation of the total variability matrix (i-vector extractor) beforehand using large corpora of speech recordings. While several choices for the similarity metric currently exist, likelihood ratios obtained through a separately trained Probabilistic Linear Discriminant Analysis (PLDA) model are commonly utilized [1].

More recently, with the advent of modern representation learning paradigms, designing effective metrics for comparing i-vectors has become an active research direction. In particular, inspired by its success in computer vision tasks [2, 3, 4], many recent efforts formulate the diarization problem as deep metric learning [5, 6, 7]. For instance, a triplet network that builds latent spaces, wherein a simple Euclidean distance metric is highly effective at separating different classes, is a widely adopted architecture. However, in contrast to its application in vision tasks, metric learning is carried out on the i-vector representations instead of the raw data [5]. Consequently, the first stage of the diarization pipeline stays intact, while the second stage is restricted to using fully connected networks. Though this modification produced state-of-the-art results in diarization and outperformed conventional scoring strategies, it does not support joint representation and task-based learning, which has become the *modus operandi* in deep learning. On the other hand, Garcia-Romero et al. [6] propose to perform joint embedding and metric learning, but use siamese networks for metric learning, which have generally shown poorer performance when compared to triplet networks [4].

In this paper, we propose to explore the use of joint representation learning and similarity metric learning with triplet loss in speaker diarization, while entirely dispensing the need for i-vector extraction. Encouraged by the recent success of *self-attention* mechanism in sequence modeling tasks [8, 9], for the first time, we leverage attention networks to model the temporal characteristics of speech segments. Experimental results on the CALLHOME corpus demonstrate that, with an appropriate embedding architecture, triplet network applied on raw audio features from a comparatively smaller dataset outperforms the same applied on i-vectors, wherein the GMM-UBM was trained using a much larger corpus.

2. Related Work

In this section, we briefly review the recent literature on techniques for speaker diarization. Over the last few years, speaker diarization approaches have quickly evolved from the traditional MFCC based GMM segmentation and BIC clustering [10, 11, 12] to systems centered around i-vector representations [13, 14]. Initially proposed for speaker verification tasks [15], i-vectors are low-dimensional features extracted over variable-length speech segments to compensate for within and between-speaker variabilities. Different speakers can then be effectively discriminated by utilizing either standard similarity metrics (e.g. cosine distance) [16] or likelihood ratios from PLDA [17]) to cluster i-vectors from the segments.

More recently, several deep learning-based solutions have

This work was supported in part by the SenSIP center at Arizona State University. This work was performed under the auspices of the U.S. Dept. of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52- 07NA27344.

10.21437/Interspeech.2018-2305

been developed to automatically infer similarity metrics to compare speech segments. More specifically, supervised metric learning architectures namely *siamese* [18, 19] and *triplet* [4, 2] networks are prevalent. Broadly speaking, these architectures infer a non-linear mapping $\mathcal{A}(\cdot)$, such that, in the resulting latent space the within-class sample distances are minimized while the between-class distances are maximized based on a certain margin. For instance, in [5], Lan *et al.* proposed to employ triplet networks on i-vectors to infer a similarity metric, and achieved state-of-the-art results over conventional metrics in the diarization literature. Despite its effectiveness, it is important to note that the feature extraction process is disentangled from the metric learning network and hence cannot support end-to-end inferencing. However, recent success of such end-to-end learning systems in computer vision applications [20, 21, 22] motivates the design of a deep metric learning architecture that works directly on the temporal sequences.

Long Short-Term Memory (LSTM) based recurrent networks have become the *de facto* solution to sequence modeling tasks including acoustic modeling [23], speech recognition [24] and Natural Language Processing (NLP) [25]. Recently, architectures entirely based on attention mechanism have shown promising value in sequence-to-sequence learning [8] and clinical data analysis [9]. Besides providing significantly faster training, attention networks demonstrate efficient modeling of long-term dependencies.

In this paper, we utilize attention networks with the triplet ranking loss to jointly learn embeddings and a similarity metrics for speech segments. To the best of our knowledge, the approaches in [7] and [6] are the most related to our work. While Bredin *et al.* [7] used triplet networks based on LSTMs, they applied it to a simpler binary classification task of speaker turn identification. Whereas, in [6], Romero *et al.* performed a similar joint learning for diarization, but based on a siamese network. Compared to the triplet ranking loss, which requires a margin to be satisfied for each given reference sample, the cross-entropy loss used in [6] requests correct prediction of all different-speaker or same-speaker pairs and hence exhibits much less flexibility.

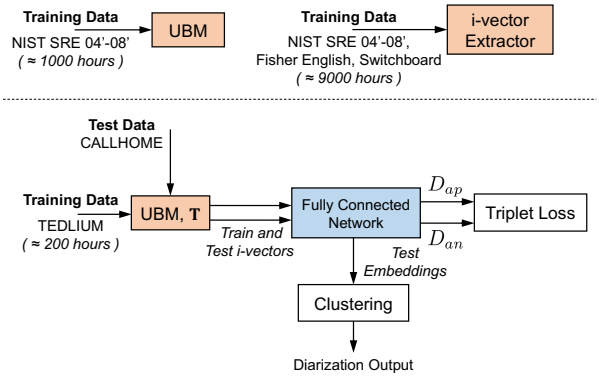
3. Proposed Approach

As shown in Figure 1(b), the proposed approach works directly with raw temporal speech features to learn a similarity metric for diarization. Compared to the baseline in Figure 1(a), the two-stage training process is simplified into a single end-to-end learning strategy, wherein deep attention models are used for embedding computation and the triplet loss is used to infer the metric. Similar to existing diarization paradigms, we first train our network using out-of-domain labeled corpus, and then perform diarization on a target dataset using unsupervised clustering. In the rest of this section, we describe the proposed approach in detail.

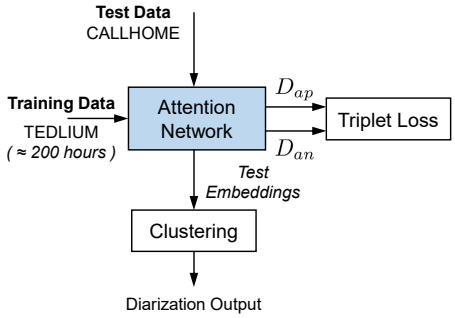
3.1. Temporal Segmentation and Feature Extraction

For the speech recordings, we first perform non-overlapping temporal segmentation into 2-second segments. Following the Voice Biometry Standardization (VBS) [1], we extract MFCC features using 25ms Hamming windows with 15ms overlap. After adding delta and double-delta coefficients, we obtain 60-dimensional feature vectors at every frame. Consequently, each data sample corresponds to a temporal sequence feature

[1] http://voicebiometry.org/



(a) *Baseline approach: (top) i-vectors are extracted using a large corpus of recordings with GMM-UBM and i-vector extractor modules; (bottom) Similarity metric is trained using a triple network trained on the i-vectors.*



(b) *Proposed approach: Joint learning of embedding and similarity metric for diarization. As observed, it completely eliminates the i-vector extraction process and enables effective training with limited data.*

Figure 1: *Comparison of diarization strategies and training data requirements for the baseline approach in [5] and the proposed approach.*

$\mathbf{x}_i \in \mathbb{R}^{T \times d}$, where T is the number of frames in each segment and $d = 60$ is the feature dimension.

3.2. Embeddings using Attention Models

As described earlier, we use attention models to learn embeddings directly from MFCC features for the subsequent metric learning task. The attention model used in our architecture is illustrated in Figure 2. The module comprised of a multi-head, self-attention mechanism is the core component of the attention model [8]. More specifically, denoting the input representation at layer ℓ as $\{\mathbf{h}_t^{\ell-1}\}_{t=1}^{T}$, we can obtain the hidden representation at time step i based on attention as follows:

$$\mathbf{h}_i^{\ell} = \sum_{t=1}^{T} w_t^{(i)} \mathbf{h}_t^{\ell-1}, \quad 1 \leq i \leq T, \tag{1}$$

$$w_t^{(i)} = \text{softmax}\left(\frac{\mathbf{h}_i^{\ell-1} \cdot \mathbf{h}_t^{\ell-1}}{\sqrt{D}}\right), \tag{2}$$

$$\mathbf{h}_i^{\ell} \leftarrow \mathcal{F}(\mathbf{h}_i^{\ell}), \tag{3}$$

Here, $D = 256$ refers to the size of the hidden layer and $\mathcal{F}$ denotes a feed-forward neural network. The attention weight in equation (2) denotes the interaction between temporal positions

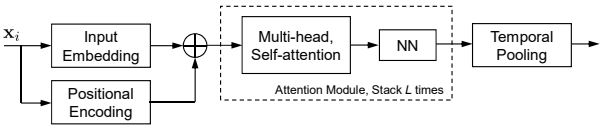


Figure 2: *Illustration of the attention model used for computing embeddings from MFCC features of speech segments.*

i and t by performing scaled inner product between the two representations. During the computation of hidden representation at time step i, $w_t^{(i)}$ weights the contribution from other temporal positions. Note that, these representations are processed by $\mathcal{F}$ before connecting to the next attention module, as shown in Figure 2. We employ a 1D convolutional layer (kernel size is 1) with ReLU activation [26] for $\mathcal{F}$. Finally, the attention module is stacked L times to learn increasingly deeper representations.

Attention-based representations in equation (1) are computed within each speech segment independently and hence this process is referred to as *self-attention*. Furthermore, the hidden representations $\mathbf{h}_t^\ell$ are computed using H different network parameterizations, denoted as heads [8], and the resulting H attention representations are concatenated together. This can be loosely interpreted as an ensemble of representations. Such a *multi-head* operation facilitates dramatically different temporal parameterizations and significantly expands the modeling power. Our current implementation sets $L = 2$ and $H = 8$.

Although attention computation explicitly models the temporal interactions, it does not encode the crucial ordering information contained in speech. The front-end positional encoding block handles this problem by mapping every relative frame position t in the segment to fixed locations in a random lookup table. As shown in Figure 2, the encoded representation is subsequently added up with the input embedding (obtained also from a 1D CNN layer). Finally, we include a temporal pooling layer to reduce the final representation $\mathbf{h}^L \in \mathbb{R}^{T\times D}$ into a D-dimensional vector by averaging along the time-axis.

3.3. Metric Learning with Triplet Loss

The representations from the deep attention model are then used to learn a similarity metric with the triplet ranking loss. Note that the attention model parameters and the metric learner are optimized jointly using back-propagation. In a triplet network, each input is constructed as a set of 3 samples $\mathbf{x} = \{\mathbf{x}_p, \mathbf{x}_r, \mathbf{x}_n\}$, where $\mathbf{x}_r$ denotes an anchor, $\mathbf{x}_p$ denotes a positive sample belonging to the same class as $\mathbf{x}_r$, and $\mathbf{x}_n$ a negative sample from a different class. Each of the samples in $\mathbf{x}$ are processed using the attention model (Section 3.2) $\mathcal{A}(\cdot) : \mathbb{R}^{T\times d} \mapsto \mathbb{R}^D$ and distances are computed in the resulting latent spaces:

$$D_{rp} = \|\mathcal{A}(\mathbf{x}_r) - \mathcal{A}(\mathbf{x}_p)\|_2$$
$$D_{rn} = \|\mathcal{A}(\mathbf{x}_r) - \mathcal{A}(\mathbf{x}_n)\|_2$$

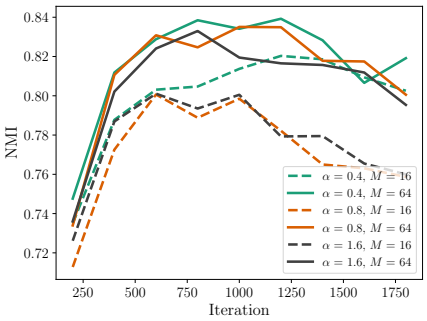
The triplet loss is defined as

$$l(\mathbf{x}_p, \mathbf{x}_r, \mathbf{x}_n) = \max(0, D_{rp}^2 - D_{rn}^2 + \alpha) \qquad (4)$$

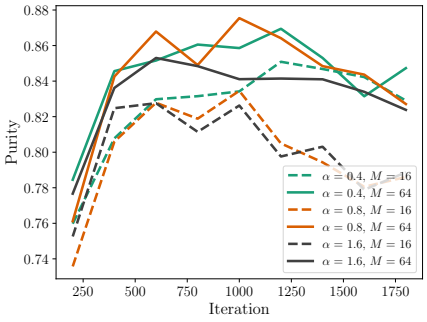
where α is the margin and the objective is to achieve $D_{rn}^2 \geq D_{rp}^2 + \alpha$. In comparison, the contrastive loss often used in siamese network includes the hinge term $\max(0, \alpha - D_{ij})$ for different-class samples $\mathbf{x}_i$ and $\mathbf{x}_j$, and hence requires α to be a global margin. Such a formulation significantly restricts the model flexibility and expressive power.

Given a large number of samples N, the computation of equation (4) is infeasible among the $\mathcal{O}(N^3)$ triplet space. It is tempting to greedily select the most effective triplets, which maximizes D_{rp} and minimizes D_{rn}. Instead of performing such hard sampling, we follow [2] to sample all possible $\mathbf{x}_p$ and only selecting *semi-hard* $\mathbf{x}_n$: the negative samples satisfying $D_{rp}^2 \leq D_{rn}^2 \leq D_{rp}^2 + \alpha$. Additionally, we adopt an online sampling strategy that restricts the sampling space to the current mini-batch during training. All sampled triplets are gathered to compute the loss in equation (4).

For the online sampling scheme, the mini-batch construction step is crucial. Ideally, each batch should cover both a large number of speakers and sufficient samples per speaker. However, we are constrained by the GPU memory (8GB) and only able to set maximum batch size $B = 256$. We preset M as the number of speakers per batch and when sampling each mini-batch, M speakers are first sampled and B/M speech segments are then sampled for every speaker. As a result, the parameter M represents the trade-off between modeling more speakers each time, and covering sufficient samples for those speakers. In our experiments, M was tuned based on the performance on the development set, as will be discussed in Section 4.1.



(a) *NMI score from the speaker clustering results.*



(b) *Purity score from the speaker clustering results.*

Figure 3: *Parameter tuning on TEDLIUM development set for triplet margin α and number of speakers per batch M. Curves for $M = 8$ and 32 are omitted for clarity.*

4. Experiments

In this section, we discuss the training process for our approach and evaluate its performance on the CALLHOME corpus.

4.1. Triplet Network Training

The proposed model was trained on the TEDLIUM corpus which consists of 1495 audio recordings. After ignoring speakers with less than 45 transcribed segments, we have a set of 1211 speakers with an average recording length of 10.2 minutes. All recordings were down-sampled to 8kHz to match the

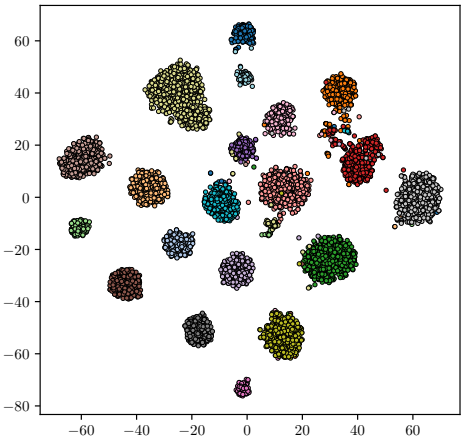


Figure 4: *2D t-SNE visualization of the first* 20 *speakers from TEDLIUM development set. Each point corresponds to one speech segment and they are color coded by the speaker.*

target CALLHOME corpus. The temporal segmentation and MFCC extraction were carried out as discussed in Section 3.1.

For the proposed approach, there are two important training parameters that need to be selected, i.e. triplet margin α and the number of speakers per mini-batch M. In order to quickly configure the parameters, we build a training subset by randomly selecting 20% of the total recordings and a development set by taking 50 recordings from the original TEDLIUM train, dev and test sets. At every 200 iterations of training on the subset, we extract the embeddings for the development set and perform speaker clustering using k-Means, with a known number of speakers. The clustering performance is evaluated by the standard Normalized Mutual Information (NMI) and Purity scores. Based on this procedure, we jointly tuned both parameters by performing a grid search on $\alpha = [0.4, 0.8, 1.6]$ and $M = [8, 16, 32, 64]$. As shown in Figure 3, having a higher M value consistently provides better clustering results and alleviates model overfitting. Additionally, a lower triplet margin generally helps the training process. Based on these observations, we configured $\alpha = 0.8, M = 64$ to train our model on the entire TEDLIUM corpus.

To study the embeddings from the attention model and the impact of triplet loss, we show the 2D t-SNE visualization [27] of samples in the development set in Figure 4. It is observed that the model is highly effective at separating unseen speakers and provides little distinction on segments from the same speakers. These embeddings achieve 0.94 score on both NMI and Purity, with k-Means clustering for the development set.

4.2. Diarization Results

The trained model is evaluated on the CALLHOME corpus [2] for diarization performance. CALLHOME consists of telephone conversations in 6 languages: Arabic, Chinese, English, German, Japanese and Spanish. In total, there are 780 transcribed conversations containing 2 to 7 speakers. After obtaining the embeddings through the proposed approach, we perform x-means [28] to estimate the number of speakers and then use k-means clustering with the estimation. We force x-means to split at least 2 clusters by initializing it with 2 centroids. Note that there are usually multiple moving parts on complete diarization systems in the literature. In particular, more sophisticated clustering algorithms [29], overlapping test segments and calibration [14] can be incorporated to improve the overall diarization performance. However, in this work we focus on investigating the efficacy of the DNN modeling and fix the other components in their basic configurations.

[2]https://ca.talkbank.org/access/CallHome/

Table 1: *Diarization Results on CALLHOME Corpus.*

System		DER (%)
i-vector	cosine	18.7
	PLDA [1]	17.6
	Triplet with FCN [5]	13.4
Proposed Approach		**12.7**

We utilize `pyannote.metric` [30] to calculate Diarization Error Rate (DER) as the evaluation metric. Although DER collectively considers false alarms, missed detections and confusion errors, most existing systems evaluated on CALLHOME [14, 6] accounts for only the confusion rate and ignores overlapping segments. Following this convention, we use the oracle speech activity regions and use only the non-overlapping sections. Additionally, there is a collar tolerance of 250ms at both beginning and end of each segment. We compare the proposed approach with the following baseline systems:

Baseline 1: i-vector + cosine/PLDA scoring. We utilize VBS pre-trained models for i-vector extraction on CALLHOME corpus. The specific GMM-UBM and i-vector extractor training data are shown in Figure 1(a). Though different from ours, the training corpus is significantly more comprehensive than the TEDLIUM set we used. The GMM-UBM consists of 2048 Gaussian components and the i-vectors are $600-$dimensional. We also used the backend LDA model contained in VBS for i-vector pre-processing. In the actual clustering, cosine or PLDA scores are used to calculate the sample-to-centroid similarities at each iteration.

Baseline 2: i-vector + triplet with FCN training. This baseline is very similar to [5] except for 2 modifications: 1) We do not consider the speaker linking procedure as there are very few repeated speakers in CALLHOME. 2) We use a larger FCN network than [5] to allow a fair comparison to the proposed approach. The hidden layers have size $512 - 1024 - 512 - 256$ and batch normalization [31] is applied at each layer after the ReLU activation. Further, i-vectors are extracted on TEDLIUM based on the transcribed speech sections with average length of 8.6 seconds. The triplet network is tuned in a similar procedure as in Section 4.1 and the best parameters were found to be $\alpha = 0.4, M = 16$.

The comparison between the proposed approach and the baselines is shown in Table 1. It is observed that baseline 2 indeed exceeds both conventional i-vector scoring methods. However, our unified learning approach trained on a much smaller TEDLIUM corpus achieves better performance, this evidencing the effectiveness of end-to-end learning.

5. Conclusions

This paper studies the role of learning embeddings under a triplet ranking loss for speaker diarization. Results on the CALLHOME corpus show that when compared to training a UBM model and then a separate triplet DNN, the two steps can be combined together to achieve improved performance with less training effort. Future work will investigate more sophisticated sampling strategies for metric learning [32] and comparative studies with existing DNN architectures [29, 6].

6. References

[1] E. Khoury, L. El Shafey, M. Ferras, and S. Marcel, "Hierarchical speaker clustering methods for the nist i-vector challenge," in *Odyssey: The Speaker and Language Recognition Workshop*, 2014.

[2] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[3] T.-Y. Lin, Y. Cui, S. Belongie, and J. Hays, "Learning deep representations for ground-to-aerial geolocalization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5007–5015.

[4] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *International Workshop on Similarity-Based Pattern Recognition*. Springer, 2015, pp. 84–92.

[5] G. Le Lan, D. Charlet, A. Larcher, and S. Meignier, "A triplet ranking-based neural network for speaker diarization and linking," *Proc. Interspeech 2017*, pp. 3572–3576, 2017.

[6] 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.

[7] H. Bredin, "Tristounet: triplet loss for speaker turn embedding," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5430–5434.

[8] 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. 6000–6010.

[9] H. Song, D. Rajan, J. J. Thiagarajan, and A. Spanias, "Attend and diagnose: Clinical time series analysis using attention models," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-2018)*, 2018.

[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] 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.

[12] 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.

[13] J. Prazak and J. Silovsky, "Speaker diarization using plda-based speaker clustering," in *Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2011 IEEE 6th International Conference on*, vol. 1. IEEE, 2011, pp. 347–350.

[14] G. Sell and D. Garcia-Romero, "Speaker diarization with plda i-vector scoring and unsupervised calibration," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, 2014, pp. 413–417.

[15] 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.

[16] I. Shapiro, N. Rabin, I. Opher, and I. Lapidot, "Clustering short push-to-talk segments," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[17] P. Kenny, "Bayesian speaker verification with heavy-tailed priors." in *Odyssey*, 2010, p. 14.

[18] 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.

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

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[21] A. Gordo, J. Almazan, J. Revaud, and D. Larlus, "End-to-end learning of deep visual representations for image retrieval," *International Journal of Computer Vision*, vol. 124, no. 2, pp. 237–254, 2017.

[22] H. Song, J. J. Thiagarajan, P. Sattigeri, and A. Spanias, "Optimizing kernel machines using deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2018.

[23] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Fifteenth annual conference of the international speech communication association*, 2014.

[24] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, 2013, pp. 6645–6649.

[25] M. Sundermeyer, R. Schlüter, and H. Ney, "Lstm neural networks for language modeling," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[26] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[27] L. Van Der Maaten, "Accelerating t-sne using tree-based algorithms." *Journal of machine learning research*, vol. 15, no. 1, pp. 3221–3245, 2014.

[28] D. Pelleg, A. W. Moore *et al.*, "X-means: Extending k-means with efficient estimation of the number of clusters." in *Icml*, vol. 1, 2000, pp. 727–734.

[29] Q. Wang, C. Downey, L. Wan, P. A. Mansfield, and I. L. Moreno, "Speaker diarization with lstm," *arXiv preprint arXiv:1710.10468*, 2017.

[30] 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*, 2017.

[31] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[32] R. Manmatha, C.-Y. Wu, A. J. Smola, and P. Krähenbühl, "Sampling matters in deep embedding learning," in *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2859–2867.