SUPERVISED ONLINE DIARIZATION WITH SAMPLE MEAN LOSS FOR MULTI-DOMAIN DATA

Enrico Fini[1]*, Alessio Brutti*[2]

[1]PerVoice Spa, Trento (Italy), [2]Fondazione Bruno Kessler, Trento (Italy)

enrico.fini@gmail.com, brutti@fbk.eu

ABSTRACT

Recently, a fully supervised speaker diarization approach was proposed (UIS-RNN) which models speakers using multiple instances of a parameter-sharing recurrent neural network. In this paper we propose qualitative modifications to the model that significantly improve the learning efficiency and the overall diarization performance. In particular, we introduce a novel loss function, we called *Sample Mean Loss* and we present a better modelling of the speaker turn behaviour, by devising an analytical expression to compute the probability of a new speaker joining the conversation. In addition, we demonstrate that our model can be trained on fixed-length speech segments, removing the need for speaker change information in inference. Using x-vectors as input features, we evaluate our proposed approach on the multi-domain dataset employed in the DIHARD II challenge: our online method improves with respect to the original UIS-RNN and achieves similar performance to an offline agglomerative clustering baseline using PLDA scoring.

Index Terms— Speaker diarization, x-vectors, clustering, supervised learning, recurrent neural networks

1. INTRODUCTION

The speaker diarization task consists in establishing "who spoke when" in a given audio recording [1, 2]. Despite having been investigated for decades, diarization is still an unsolved problem in many real scenarios, as highlighted by the recent DIHARD I and DIHARD II challenges [3].

Typically, speaker diarization is addressed integrating several different components: voice activity detection, speaker change detection, feature extraction and clustering. Most of the research works in literature focus on extracting highly discriminative feature vectors. The first example in this direction are i-vectors [4, 5], which represent a given utterance with a single fixed-dimensional feature vector. The recent rise of neural paradigms has led to the introduction of a variety of approaches to extract the so-called speaker embeddings. These are, typically, derived from the outputs of the inner layers of a neural network trained on a speaker classification task [6]. The most popular embeddings are d-vectors [7] and x-vectors [8, 9].

Conversely, not much progress has been done with regard to clustering. In most of the approaches, this stage is still based on the Agglomerative Hierachical Clustering (AHC) [10] in combination with Probabilistic Linear Discriminat Analysis (PLDA) scoring [11]. Recently, spectral clustering [12][13], and variatioanl bayesian clustering [14, 15] have been introduced, showing promising result. Also, alternatives to the PLDA scoring have been introduced using neural networks that learn how to score two speech segments [16], using siamese networks [17] or Bi-LSTMs [18]. Nevertheless, clustering remains unsupervised and heavily dependent on fine-tuned hyperparameters (e.g. thresholds to stop clustering).

Recently, efforts have been made to formulate clustering in a supervised learning framework [19, 20, 21]. Supervised clustering is attractive because it can be optimized on the diarization metrics directly, or learning context dependent parameters. Additionally, supervision allows to improve performance by learning from the increasing amount of data at our disposal. For example, [19] tackles the diarization problem as a classification task, while [20] uses a permutation invariant loss and a clustering loss to dynamically identify speakers. Both [19] and [20] assume that the number of speaker is known apriori or at least bounded. This assumption is removed in the UIS-RNN [21]: a fully supervised approach which handles an unbound number of speakers using an online generative process. Speaker distributions are modelled with multiple instances of a parameter-sharing Recurrent Neural Network (RNN). A further, strong advantage of [21] over traditional clustering algorithms is the fact that decoding is online using beam search [22]. Though online diarization had already been explored, using both unsupervised [23, 24, 25] and supervised [19] paradigms, the UIS-RNN stands out in terms of performance, outperforming the previous offline state of the art on telephone data.

Although these are very interesting results, an online system that works well across multiple domains still remains an open problem. As a matter of fact, diarization systems presented in the literature appear to work relatively well on domains with a low number of speakers and no overlapping speech, like telephone data, while performance tends to deteriorate in more challenging contexts such as meetings or dinner parties.

In this paper we present an evolution of the UIS-RNN [21], which substantially improves the performance. First of all, we introduce a new loss function for training the RNN that models speakers, which provides faster convergence, encouraging the network to find deeper minima, and generalizes better on the evaluation set. Secondly, we propose a semantically grounded formulation for the unseen speaker intervention probability that is easy to calculate and improves performance in inference. In addition we train on fixed-length speech segments, and let the neural network aggregate embeddings, removing the constraint on speaker change information in inference. Finally, we shed light on the performance of the proposed method with respect to the original UIS-RNN in a multi-domain scenario through extensive testing on the DIHARD datasets. We also make our results reproducible, since we use a publicly available embedding extractor and fully disclose our code[1].

[1]The first author performed this work as an intern at PerVoice and Fondazione Bruno Kessler. The implementation of this paper is available at: https://github.com/DonkeyShot21/uis-rnn-sml

2. PROPOSED APPROACH

Given a set of embeddings $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$ and the related speaker labels $\mathbf{Y} = (y_1, \ldots, y_T)$, where T is the total number of observations, we can cast the diarization problem in a probabilistic framework, looking for the sequence of speaker labels that maximizes the joint probability:

$$\hat{\mathbf{Y}} = \arg\max_{\mathbf{Y}} P(\mathbf{X}, \mathbf{Y}), \tag{1}$$

If we model eq. 1 as an online generative problem as in [21], we can rewrite the joint probability as:

$$\begin{aligned} P(\mathbf{X}, \mathbf{Y}) &= p(\mathbf{x}_t, y_t, z_t | \mathbf{x}_{[t-1]}, y_{[t-1]}, z_{[t-1]}) \\ &= \underbrace{p(\mathbf{x}_t | \mathbf{x}_{[t-1]}, y_t)}_{\text{sequence generation}} \cdot \underbrace{p(y_t | y_{[t-1]}, z_t)}_{\text{assignment}} \cdot \underbrace{p(z_t | z_{[t-1]})}_{\text{speaker change}}, \end{aligned} \tag{2}$$

where $z_t = \mathbb{1}(y_t \neq y_{t-1})$ is a hidden binary indicator of speaker change and $[t]$ denotes all observations up to t included. In the original definition of the UIS-RNN [21], the **speaker change** term of eq. 2 is modelled by a coin flipping process where the only parameter is p_0, the transition probability. The **speaker assignment** term is implemented as a distance dependent Chinese Restaurant Processes (ddCRP) [26], a Bayesian nonparametric process that guides how speakers interleave in the time domain. Finally, the **sequence generation** part of eq. 2 is modelled using an RNN, specifically a Gated Recurrent Unit (GRU), that parametrizes the distribution of embeddings assuming a Gaussian distribution as follows:

$$\mathbf{x}_t | \mathbf{x}_{[t-1]}, y_{[t]} \sim \mathcal{N}\left(\boldsymbol{\mu}\left(GRU_{\boldsymbol{\theta}}\left(\mathbf{x}_{t'} \in \mathbf{x}_{[t-1]} | y_{t'} = y_t\right)\right), \sigma^2 \boldsymbol{I}\right), \tag{3}$$

where $\boldsymbol{\mu}\left(GRU_{\boldsymbol{\theta}}(\cdot)\right)$ is the averaged output of the neural network with parameters $\boldsymbol{\theta}$ instantiated for speaker y_t.

2.1. Original UIS-RNN training

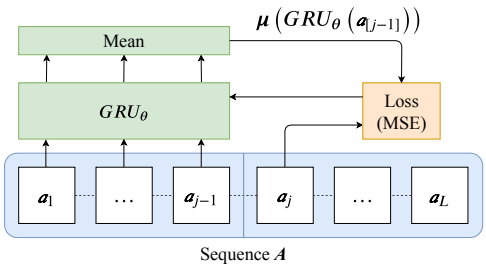

Fig. 1. Block diagram of the original UIS-RNN training strategy for a generic sequence $\boldsymbol{A}$.

Given a dataset $\boldsymbol{\mathcal{D}} = \{(\mathbf{X}_1, \ldots, \mathbf{X}_M), (\mathbf{Y}_1, \ldots, \mathbf{Y}_M)\}$, including M sequences of embeddings and related label, the optimal set of network parameters $\boldsymbol{\theta}^*$ can be obtained minimizing the following negative log likelihood [21]:

$$\mathcal{L} = \sum_{m=1}^{|\boldsymbol{\mathcal{D}}|} -\ln p(\mathbf{X}_m | \mathbf{Y}_m; \boldsymbol{\theta}). \tag{4}$$

Using the model in eq. 3, eq. 4 can be reformulated in a Mean Squared Error (MSE) fashion [27]:

$$\mathcal{L}_{\text{MSE}} = \sum_{i=1}^{|\boldsymbol{\mathcal{D}}_A|} \sum_{j=1}^{|\boldsymbol{A}_i|} \left\| \boldsymbol{a}_{i,j} - \boldsymbol{\mu}\left(GRU_{\boldsymbol{\theta}}\left(\boldsymbol{a}_{i,[j-1]}\right)\right) \right\|^2. \tag{5}$$

Given S speakers and P permutations applied to the data for augmentation purposes, $\boldsymbol{\mathcal{D}}_A = (\boldsymbol{A}_1, \ldots, \boldsymbol{A}_{S \times P})$ is a set of single speaker sequences, where each sequence $\boldsymbol{A}_i = (\boldsymbol{a}_{i,1}, \ldots, \boldsymbol{a}_{i,L_i}) \in \boldsymbol{\mathcal{D}}_A$ is obtained by concatenating a random permutation of the embeddings generated by the i-th speaker. L_i and $\boldsymbol{a}_{i,j}$ are respectively the length and the j-th embedding of sequence $\boldsymbol{A}_i$.

Note that, since the sequences are shuffled, the network can not learn any causal relationship between observations and how to predict the next embedding. Basically, the network is trained to generate samples $\varkappa_t$ from an auxiliary distribution q with expectation $\mathbb{E}[\varkappa_t]$ equal to $\mathbb{E}[\mathbf{x}_t]$. Therefore, the network will learn to predict the mean of the distribution of the embeddings. Figure 1 graphically describes the training presented in [21] and implemented in [27].

2.2. Sample Mean Loss training: UIS-RNN-SML

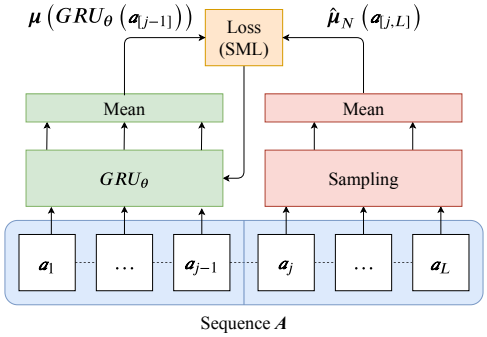

Fig. 2. Block diagram of the proposed UIS-RNN-SML training approach for a generic sequence $\boldsymbol{A}$.

In this section we propose a modified loss that relies on more accurate targets for the network output. Rather than adjusting the network by comparing the mean of its outputs with the next observed embedding, we define a MSE loss with respect to the actual mean of the speaker embeddings of a given speaker. This results in defining a predictor of the mean of the embedding distribution, having seen only a small sample of it. More formally, we replace the MSE loss in eq. 5 with:

$$\mathcal{L}' = \sum_{i=1}^{|\boldsymbol{\mathcal{D}}_A|} \sum_{j=1}^{|\boldsymbol{A}_i|} \left\| \mathbb{E}\left[s(i)\right] - \boldsymbol{\mu}\left(GRU_{\boldsymbol{\theta}}\left(\boldsymbol{a}_{i,[j-1]}\right)\right) \right\|^2, \tag{6}$$

where $s(i)$ is a function that maps each index i to the embedding distribution of the speaker who generated the observations in $\boldsymbol{A}_i$.

In practice, the actual probability distribution of the embeddings is not available. In addition, given the limited amount of labelled data, using the bare mean over the sequence would lead to overfitting. Therefore, we build the ground truth for the network by estimating the mean over a collection of unseen samples we draw randomly with replacement from the permuted sequence itself. In formulas, given a generic sequence $\boldsymbol{A}_i$ and a subset $\boldsymbol{H} = (\boldsymbol{h}_1, \ldots, \boldsymbol{h}_N) \subseteq \boldsymbol{A}_i$ of N randomly sampled embeddings, we estimate the mean of the embeddings as: $\hat{\boldsymbol{\mu}}_N(\boldsymbol{A}_i) = \left(\sum_i^N \boldsymbol{h}_i\right)/N$ Eq. 6 is then rewritten leading to our Sample Mean Loss (SML) definition:

$$\mathcal{L}_{\text{SML}} = \sum_{i=1}^{|\boldsymbol{\mathcal{D}}_A|} \sum_{j=1}^{|\boldsymbol{A}_i|} \left\| \hat{\boldsymbol{\mu}}_N\left(\boldsymbol{a}_{i,[j,L_i]}\right) - \boldsymbol{\mu}\left(GRU_{\boldsymbol{\theta}}\left(\boldsymbol{a}_{i,[j-1]}\right)\right) \right\|^2, \tag{7}$$

where we denote the ordered set $(j, \dots, L_i)$ as $[j, L_i]$. Figure 2 depicts the proposed training approach for a generic sequence $\boldsymbol{A}$.

2.3. New speaker probability

One of the most interesting advantages of the UIS-RNN [21] over other supervised methods, like [20], is its ability to model an unbounded number of speakers. This is achieved using a ddCRP model [26] that provides the probability of switching back to a previously seen speaker proportionally to the number of turns of that speaker and accounts for the probability of a new speaker joining the conversation. Assuming speakers are numerated in order of appearance starting from 1, we let:

$$p\left(y_t = k | z_t = 1, y_{[t-1]}\right) \propto N_{k,t-1} \tag{8}$$

$$p\left(y_t = \max\left(y_{[t-1]}\right) + 1 | z_t = 1, y_{[t-1]}\right) \propto \alpha, \tag{9}$$

where $N_{k,t-1}$ is the number of blocks of contiguous utterances of speaker k. The probability of switching to a new speaker is controlled by the parameter α which is critical for the correct functioning of the whole framework: large values of α force the model to over estimate the number of speakers, instantiating several networks; conversely small values result in limiting the number of speakers by merging clusters.

With respect to the estimation performed in [21], we propose the following analytical formulation for α:

$$\alpha = \frac{\sum_{m=1}^{|\boldsymbol{\mathcal{D}}|} \left(\max\left(\mathbf{Y}_m\right) - 1\right)}{\sum_{m=1}^{|\boldsymbol{\mathcal{D}}|} \sum_{t=1}^{|\mathbf{Y}_m|} \mathbb{1}\left(y_{m,t} \neq y_{m,t+1}\right)}. \tag{10}$$

This formulation has the advantage that it can be derived from eq. 9, and therefore it is semantically coherent with the role of the parameter. In addition, the value of the parameter is estimated straight from the data, independently of any the error metric or heuristic.

3. EXPERIMENTS AND RESULTS

3.1. Dataset

We train and evaluate our method on the data used in the DIHARD-II challenge [3]. The challenge features two audio input conditions: single channel and multi-channel. We focus on single channel data with reference Speech Activity Detection (SAD), as per the track 1 of the competition. The dataset is divided into two subsets, development and evaluation, each consisting of selections of 5-10 minute audio files sampled from 11 different conversational domains for a total of approximately 2 hours of audio.

Using stratified holdout, we further split the development set into training set (80%) and validation set (20%). Also, we randomize the holdout procedure, such that for every experiment we get a different data partitioning. Stratification is performed over the set of domains, according to their frequency in the whole development set.

Although the proposed approach does not handle cases where multiple speakers are active simultaneously, we do not exclude overlapping speech segment from the training material. In fact, we observed that considering multi-speaker segments as a separate speaker slightly improves performance.

3.2. Experimental setup

We use as speaker embeddings of our supervised diarization system x-vectors [9] using the pre-trained models available in the Kaldi diarization recipe [28]. X-vectors with dimension 512 are extracted from non-overlapped 1 second speech segments and are subsequently reduced to dimension 200 with Principal Component Analysis (PCA) before feeding them to the model.

For what concerns the sequence generation component, our network resembles the architecture presented in [21]. However, since we are using different features, x-vectors on fixed-length segments instead of d-vectors extracted from ground truth speaker segments, we explored several configurations varying the sizes of the layers. We found that reasonable results are obtained using one recurrent and one fully connected layer with 200 units each.

The other two parameters, p_0 and α for the speaker change and the speaker assignment components respectively, are estimated using their analytical formulations. For the transition probability p_0 we apply the same formula as in [21], while for α we use eq. 10. We also explored some search based techniques for hyperparameter optimization, like grid search and line search, but we found they do not provide noticeable improvements in performance. Furthermore, the value for the variance of the observations σ^2 is optimized during training using Adam, as in [21].

Apart from the SML loss, two more regularization losses help the model to converge [27]. The first one is a simple L2 loss on the parameters of the GRU, the second one uses an inverse gamma distribution to regularize the value of σ^2 that would otherwise diverge to very large values.

In inference we use beam search with beam size $\beta = 15$. Unlike in [21], in our dataset we can not consider the number of speakers to be bounded. This makes inference expensive.

Networks are trained several times using Adam optimizer and the best model is selection by measuring the Diarization Error Rate (DER) on the validation set, using a smaller beam width ($\beta = 2$) to reduce the computational cost.

DER is measured using *dscore* [29], the official scoring tool of DIHARD-II competition which does not account for any forgiveness collar, considering also overlapped speech segment. However, since none of the methods under evaluation handle overlapped speech we also report performance without overlap.

3.3. Results

Method	DER	DER - no overlap
Cum. mean + beam search	34.0	26.7
UIS-RNN [21][27]	30.9	23.4
UIS-RNN + eq.10	30.3	22.8
UIS-RNN-SML + eq.10	**27.3**	**19.4**
PLDA + AHC [3] (offline)	26.1	17.7

Table 1. DER on track 1 of DIHARD II test data, with and without overlapping speech. PLDA+AHC refers to the off-line baseline provided with the challenge. $N = 2$ in UIS-RNN-SML.

Table 1 reports the performance of our proposed UIS-RNN-SML, based on SML and α estimation, in comparison against two online baselines. The first one is a naïve implementation in which the GRU is replaced by a simple cumulative mean of the embeddings (Cum. mean + beam search in Table 1). This naïve baseline helps highlighting the contribution of the neural network, disentangling it from the other components of the framework. The second is the original UIS-RNN [21], using the implementation provided in [27]. To give an idea of how difficult the task is, we also report the offline baseline provided in the DIHARD-II challenge [3], which performs

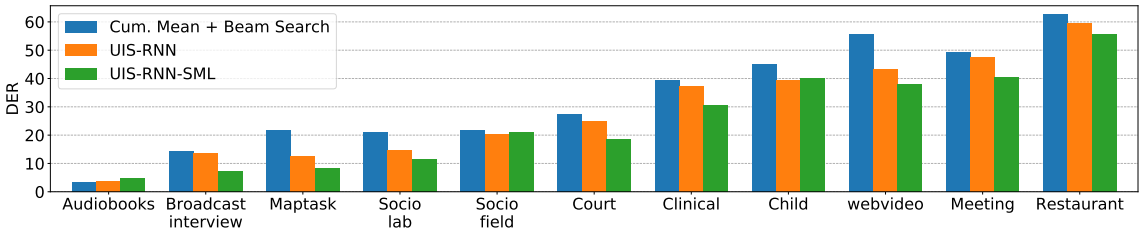

Fig. 3. DER for each domain in track 1 of the DIHARD-II test set. Domains are displayed in ascending order of difficulty.

diarization by scoring the x-vectors with PLDA [11], and clustering using AHC [10].

The naïve implemenation based on cumulative mean with beam search is outperformed by the UIS-RNN by a large margin, with and without overlapping speech segments. This confirms that the simple mean of a partial sequence of embeddings does not properly model the speaker and that the neural network makes an active contribution. A further small but significative DER reduction, both with and without overlap, with respect to the original implementation is provided by estimating α with eq. 10 (third row in Table 1).

Finally, a larger leap in performance is achieved by replacing the original loss function with the SML we proposed. Note that the UIS-RNN-SML achieves similar performance to the offline baseline used in DIHARD-II [3], although online unsupervised clustering algorithms usually perform significantly worse than offline clustering algorithms. The performance improvement is due the regularizing effect introduced by the SML in training. We observed that, keeping *learning rate* and *batch size* fixed, training with SML is much less noisy than the original one: using the more accurate supervision given by the sample mean results in better gradients, which in turn helps convergence to deeper minima. The stabilizing effect of the SML is evident in Fig. 4 where we report the variance of the means of the speaker clusters generated by the network during training. Models trained with eq. 7 exhibit less output variance compared to those trained with eq. 5. This behaviour turns out to be very beneficial in the decoding phase when the means of the clusters should not change dramatically while the sequence unfolds.

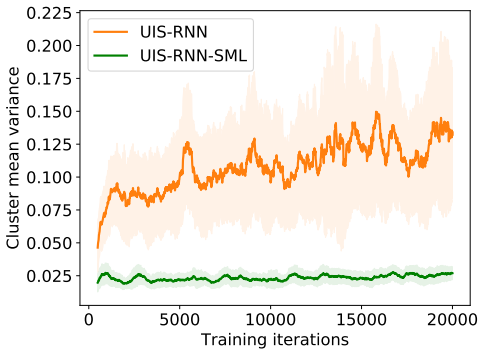

Fig. 4. Cluster mean variance over sequences during the first 20K training iterations.

For a better understanding of the behaviour of our proposed method, Fig. 3 reports the DER for each context in the dataset. Our method is better than the original UIS-RNN in all the most challenging contexts, except for "socio field" and "child", where our performance is basically aligned to the other methods. We observe a small performance deterioration in "Audiobooks". This occurs because the UIS-RNN-SML, predicting the mean more accurately, produces slightly smaller values for the cluster variance σ^2. Although this is beneficial in most cases, it can marginally reduce performance in contexts with very low number of speakers. This disadvantage can be partially alleviated by defining context dependent α and p_0.

Finally we evaluate the impact of the number of samples N used to estimate the mean of the distribution. Fig. 5 shows the DER on the whole evaluation set for different values of N. On these data, $N = 2$ provides the lower DER, but values from 2 to 4 produce very similar results. Unsurprisingly, performance degrades using larger values for N, due to overfitting, because the sample mean approximates the real mean too tightly. Note that the case $N = 1$ would be equivalent to the UIS-RNN except for the fact that observations are sampled with replacement. This gives a considerable improvement (27.83% against 30.3%) because outliers of the speaker clusters are less likely to be observed by the network as targets during training, reducing the overall variance of the output.

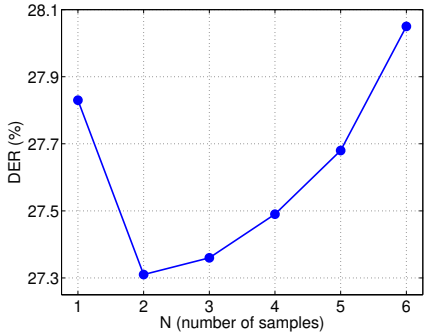

Fig. 5. DER on track 1 of DIHARD II test data, varying number of samples N in SML.

4. CONCLUSIONS

In this paper we presented an evolution of a supervised speaker diarization system where the clustering module is replaced by a trainable model called unbounded interleaved-state RNN. Specifically, we proposed a modified loss function that stimulates the neural network to model speakers more accurately. In addition, we introduced a semantically grounded formulation for the estimation of the parameter that controls the speaker assignment probability. We evaluated the proposed online diariaztion approach on the DIHARD-II multi-domain data, showing, through extensive experiments, that it outperforms the original UIS-RNN formulation. Finally, we fully disclose our code and trained models to make our results reproducible.

5. REFERENCES

[1] 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.

[2] 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, Feb 2012.

[3] N. Ryant, K. Church, C. Cieri, A. Cristia, J. Du, S. Ganapathy, and M. Liberman, "The Second DIHARD Diarization Challenge: Dataset, Task, and Baselines," in *INTERSPEECH*, 2019, pp. 978–982.

[4] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 5, pp. 980–988, 2008.

[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] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *INTERSPEECH*, 2018.

[7] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *International Conference on Acoustics, Speech and Signal Processing*, 2014, pp. 4052–4056.

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

[9] G. Sell, D. Snyder, A. McCree, D. Garcia-Romero, J. Villalba, M. Maciejewski, V. Manohar, N. Dehak, D. Povey, S. Watanabe, and S. Khudanpur, "Diarization is hard: Some experiences and lessons learned for the JHU team in the inaugural DIHARD challenge," in *INTERSPEECH*, 2018, pp. 2808–2812.

[10] K. J. Han, S. Kim, and S. S. Narayanan, "Strategies to improve the robustness of agglomerative hierarchical clustering under data source variation for speaker diarization," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 8, pp. 1590–1601, 2008.

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

[12] H. Ning, M. Liu, H. Tang, and T. S. Huang, "A spectral clustering approach to speaker diarization," in *Ninth International Conference on Spoken Language Processing*, 2006.

[13] Q. Wang, C. Downey, L. Wan, P. A. Mansfield, and I. L. Moreno, "Speaker diarization with lstm," in *International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2018, pp. 5239–5243.

[14] M. Diez, L. Burget, and P. Matejka, "Speaker diarization based on bayesian hmm with eigenvoice priors," in *Speaker Odissey*, 06 2018, pp. 147–154.

[15] M. Diez, L. Burget, S. Wang, J. Rohdin, and J. ernock, "Bayesian HMM Based x-Vector Clustering for Speaker Diarization," in *INTERSPEECH*, 2019, pp. 346–350.

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

[17] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-attentive speaker embeddings for text-independent speaker verification," in *INTERSPEECH*, 2018, pp. 3573–3577.

[18] Qingjian Lin, Ruiqing Yin, Ming Li, Herv Bredin, and Claude Barras, "LSTM Based Similarity Measurement with Spectral Clustering for Speaker Diarization," in *INTERSPEECH*, 2019, pp. 366–370.

[19] C.Chaitanya Asawa, N. Bhattasali, and A. Jiang, "Deep learning approaches for online speaker diarization," 2017.

[20] Y Fujita, N Kanda, S Horiguchi, K. Nagamatsu, and S Watanabe, "End-to-end neural speaker diarization with permutation-free objectives," *INTERSPEECH*, Sep 2019.

[21] A Zhang, Q Wang, Z Zhu, J Paisley, and C Wang, "Fully supervised speaker diarization," in *International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2019, pp. 6301–6305.

[22] M. F. Medress, F. S. Cooper, J. W. Forgie, C. C. Green, D. H. Klatt, M. H. O'Malley, E. P. Neuburg, A. Newell, D. R. Reddy, B. Ritea, J. E. Shoup-Hummel, D. E. Walker, and W. A. Woods, "Speech understanding systems: Report of a steering committee," *Artificial Intelligence*, vol. 9, no. 3, pp. 307 – 316, 1977.

[23] J. Geiger, F. Wallhoff, and G. Rigoll, "GMM-UBM based open-set online speaker diarization," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[24] P. A. Mansfield, Q. Wang, C. Downey, L. Wan, and Ignacio Lopez Moreno, "Links: A high-dimensional online clustering method," *arXiv preprint arXiv:1801.10123*, 2018.

[25] D. Dimitriadis and P. Fousek, "Developing on-line speaker diarization system," in *INTERSPEECH*, 2017.

[26] D. M. Blei and P. Frazier, "Distance dependent chinese restaurant processes," *Journal of Machine Learning Research*, vol. 12, no. Aug, pp. 2461–2488, 2011.

[27] Google, "Official library for the Unbounded Interleaved-State Recurrent Neural Network (UIS-RNN) algorithm," `https://github.com/google/uis-rnn` (Oct. 2019).

[28] 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 *Workshop on Automatic Speech Recognition and Understanding*, Dec. 2011.

[29] N. Ryant, "dscore, official scoring tool for DIHARD-II," `https://github.com/nryant/dscore` (Oct. 2019).