FULLY SUPERVISED SPEAKER DIARIZATION

Aonan Zhang[1,2]   Quan Wang[1]   Zhenyao Zhu[1]   John Paisley[2]   Chong Wang[1]

[1]Google Inc., USA   [2]Columbia University, USA

[1] { aonan, quanw, zyzhu, chongw } @google.com   [2] { az2385, jpaisley } @columbia.edu

ABSTRACT

In this paper, we propose a fully supervised speaker diarization approach, named *unbounded interleaved-state recurrent neural networks* (UIS-RNN). Given extracted speaker-discriminative embeddings (*a.k.a.* d-vectors) from input utterances, each individual speaker is modeled by a parameter-sharing RNN, while the RNN states for different speakers interleave in the time domain. This RNN is naturally integrated with a distance-dependent Chinese restaurant process (ddCRP) to accommodate an unknown number of speakers. Our system is fully supervised and is able to learn from examples where time-stamped speaker labels are annotated. We achieved a 7.6% diarization error rate on NIST SRE 2000 CALLHOME, which is better than the state-of-the-art method using spectral clustering. Moreover, our method decodes in an online fashion while most state-of-the-art systems rely on offline clustering.

Index Terms— Speaker diarization, d-vector, clustering, recurrent neural networks, Chinese restaurant process

1. INTRODUCTION

Aiming to solve the problem of "who spoke when", most existing speaker diarization systems consist of multiple relatively independent components [1, 2, 3], including but not limited to: (1) A speech segmentation module, which removes the non-speech parts, and divides the input utterance into small segments; (2) An embedding extraction module, where speaker-discriminative embeddings such as speaker factors [4], i-vectors [5], or d-vectors [6] are extracted from the small segments; (3) A clustering module, which determines the number of speakers, and assigns speaker identities to each segment; (4) A resegmentation module, which further refines the diarization results by enforcing additional constraints [1].

For the embedding extraction module, recent work [2, 3, 7] has shown that the diarization performance can be significantly improved by replacing i-vectors [5] with neural network embeddings, *a.k.a.* d-vectors [6, 8]. This is largely due to the fact that neural networks can be trained with big datasets, such that the model is sufficiently robust against varying speaker accents and acoustic conditions in different use scenarios.

However, there is still one component that is unsupervised in most modern speaker diarization systems — the clustering module. Examples of clustering algorithms that have been used in diarization systems include Gaussian mixture models [7, 9], mean shift [10], agglomerative hierarchical clustering [2, 11], k-means [3, 12], Links [3, 13], and spectral clustering [3, 14].

The first author performed this work as an intern at Google. The implementation of the algorithms in this paper is available at: https://github.com/google/uis-rnn

Since both the number of speakers and the segment-wise speaker labels are determined by the clustering module, the quality of the clustering algorithm is critically important to the final diarization performance. However, the fact that most clustering algorithms are unsupervised means that, we will not able to improve this module by learning from examples when the time-stamped speaker labels ground truth are available. In fact, in many domain-specific applications, it is relatively easy to obtain such high quality annotated data.

In this paper, we replace the unsupervised clustering module by an online generative process that naturally incorporates labelled data for training. We call this method *unbounded interleaved-state recurrent neural network* (UIS-RNN), based on these facts: (1) Each speaker is modeled by an instance of RNN, and these instances share the same parameters; (2) An unbounded number of RNN instances can be generated; (3) The states of different RNN instances, corresponding to different speakers, are interleaved in the time domain. Within a fully supervised framework, our method in addition handles complexities in speaker diarization: it automatically learns the number of speakers within each utterance via a Bayesian non-parametric process, and it carries information through time via the RNN.

The contributions of our work are summarized as follows:

1. Unbounded interleaved-state RNN, a trainable model for the general problem of segmenting and clustering temporal data by learning from examples.
2. Framework for a fully supervised speaker diarization system.
3. New state-of-the-art performance on NIST SRE 2000 CALL-HOME benchmark.
4. Online diarization solution with offline quality.

2. BASELINE SYSTEM USING CLUSTERING

Our diarization system is built on top of the recent work by Wang *et al.* [3]. Specifically, we use exactly the same segmentation module and embedding extraction module as their system, while replacing their clustering module by an unbounded interleaved-state RNN.

As a brief review, in the baseline system [3], a text-independent speaker recognition network is used to extract embeddings from sliding windows of size 240ms and 50% overlap. A simple voice activity detector (VAD) with only two full-covariance Gaussians is used to remove non-speech parts, and partition the utterance into non-overlapping segments with max length of 400ms. Then we average window-level embeddings to segment-level d-vectors, and feed them into the clustering algorithm to produce final diarization results. The workflow of this baseline system is shown in Fig. 1.

The text-independent speaker recognition network for computing embeddings has three LSTM layers and one linear layer. The network is trained with the state-of-the-art generalized end-to-end loss [6]. We have been retraining this model for better performance, which will be later discussed in Section 4.1.
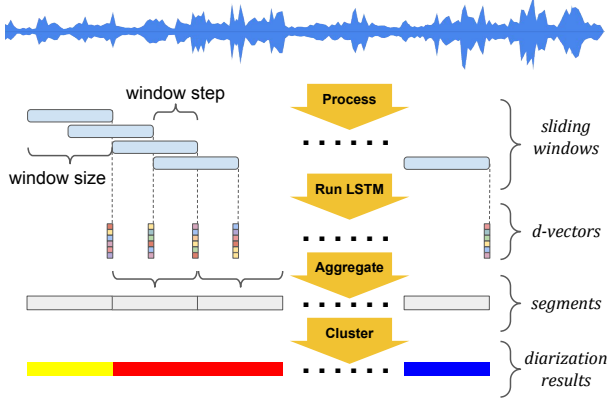
Fig. 1. The baseline system architecture [3].

3. UNBOUNDED INTERLEAVED-STATE RNN

3.1. Overview of approach

Given an utterance, from the embedding extraction module, we get an *observation sequence* of embeddings $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$, where each $\mathbf{x}_t \in \mathbb{R}^d$. Each entry in this sequence is a real-valued d-vector corresponding to a segment in the original utterance. In the supervised speaker diarization scenario, we also have the ground truth speaker labels for each segment $\mathbf{Y} = (y_1, y_2, \ldots, y_T)$. Without loss of generality, let $\mathbf{Y}$ be a sequence of positive integers by the order of appearance.

For example, $\mathbf{Y} = (1, 1, 2, 3, 2, 2)$ means this utterance has six segments, from three different speakers, where $y_t = k$ means segment t belongs to speaker k.

UIS-RNN is an online generative process of an entire utterance $(\mathbf{X}, \mathbf{Y})$, where[1]

$$p(\mathbf{X}, \mathbf{Y}) = p(\mathbf{x}_1, y_1) \cdot \prod_{t=2}^{T} p(\mathbf{x}_t, y_t | \mathbf{x}_{[t-1]}, y_{[t-1]}). \quad (1)$$

To model speaker changes, we use an augmented representation

$$p(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = p(\mathbf{x}_1, y_1) \cdot \prod_{t=2}^{T} p(\mathbf{x}_t, y_t, z_t | \mathbf{x}_{[t-1]}, y_{[t-1]}, z_{[t-1]}), \quad (2)$$

where $\mathbf{Z} = (z_2, \ldots, z_T)$, and $z_t = \mathbb{1}(y_t \neq y_{t-1}) \in \{0, 1\}$ is a binary indicator for speaker changes. For example, if $\mathbf{Y} = (1, 1, 2, 3, 2, 2)$, then $\mathbf{Z} = (0, 1, 1, 1, 0)$. Note that $\mathbf{Z}$ is uniquely determined by $\mathbf{Y}$, but $\mathbf{Y}$ cannot be uniquely determined by a given $\mathbf{Z}$, since we don't know which speaker we are changing to. Here we leave z_1 undefined, and factorize each product term in Eq. (2) as three parts that separately model sequence generation, speaker assignment, and speaker change:

$$\begin{aligned} &p(\mathbf{x}_t, y_t, z_t | \mathbf{x}_{[t-1]}, y_{[t-1]}) \\ &= \underbrace{p(\mathbf{x}_t | \mathbf{x}_{[t-1]}, y_{[t]})}_{\text{sequence generation}} \cdot \underbrace{p(y_t | z_t, y_{[t-1]})}_{\text{speaker assignment}} \cdot \underbrace{p(z_t | z_{[t-1]})}_{\text{speaker change}}. \end{aligned} \quad (3)$$

For the first entry of the sequence, we let $y_1 = 1$ and there is no need to model speaker assignment and speaker change. In Section 3.2, we introduce these components separately.

------
[1]We denote an ordered set $(1, 2, \ldots, t)$ as $[t]$.

3.2. Details on model components

3.2.1. Speaker change

We assume the probability of $z_t \in \{0, 1\}$ follows:

$$p(z_t = 1 | z_{[t-1]}, \boldsymbol{\lambda}) = g_{\boldsymbol{\lambda}}(z_{[t-1]}), \quad (4)$$

where $g_{\boldsymbol{\lambda}}(\cdot)$ is a function paramaterized by $\boldsymbol{\lambda}$. Since z_t indicates speaker change at time t, we have

$$p(y_t = y_{t-1} | z_t, y_{[t-1]}) = 1 - z_t. \quad (5)$$

In general, $g_{\boldsymbol{\lambda}}(\cdot)$ could be any function, such as an RNN. But for simplicy, in this work, we make it a constant value $g_{\boldsymbol{\lambda}}(z_{[t-1]}) = p_0 \in [0, 1]$. This means $\{z_t\}_{t \in [2,T]}$ are independent binary variables parameterized by $\boldsymbol{\lambda} = \{p_0\}$:

$$z_t \sim_{iid.} \text{Binary}(p_0). \quad (6)$$

3.2.2. Speaker assignment process

One of the biggest challenges in speaker diarization is to determine the total number of speakers for each utterance. To model the speaker turn behavior in an utterance, we use a distance dependent Chinese restaurant process (ddCRP) [15], a Bayesian nonparametric model that can potentially model an unbounded number of speakers. Specifically, when $z_t = 0$, the speaker remains unchanged. When $z_t = 1$, we let

$$\begin{aligned} p(y_t = k | z_t = 1, y_{[t-1]}) &\propto N_{k,t-1}, \\ p(y_t = K_{t-1} + 1 | z_t = 1, y_{[t-1]}) &\propto \alpha. \end{aligned} \quad (7)$$

Here $K_{t-1} := \max y_{[t-1]}$ is the total number of unique speakers up to the $(t-1)$-th entry. Since $z_t = 1$ indicates a speaker change, we have $k \in [K_{t-1}] \setminus \{y_{t-1}\}$. In addition, we let $N_{k,t-1}$ be the number of *blocks* for speaker k in $y_{[t-1]}$. A *block* is defined as a maximum-length subsequence of *continuous* segments that belongs to a single speaker. For example, if $y_{[6]} = (1, 1, 2, 3, 2, 2)$, then there are four blocks $(1, 1)|(2)|(3)|(2, 2)$ separated by the vertical bar, with $N_{1,5} = 1$, $N_{2,5} = 2$, $N_{3,5} = 1$. The probability of switching back to a previously appeared speaker is proportional to the number of continuous speeches she/he has spoken. There is also a chance to switch to a new speaker, with a probability proportional to a constant α. The joint distribution of $\mathbf{Y}$ given $\mathbf{Z}$ is

$$p(\mathbf{Y} | \mathbf{Z}, \alpha) = \frac{\alpha^{K_T - 1} \prod_{k=1}^{K_T} \Gamma(N_{k,T})}{\prod_{t=2}^{T} (\sum_{k \in [K_{t-1}] \setminus \{y_{t-1}\}} N_{k,t-1} + \alpha)^{\mathbb{1}(z_t = 1)}}. \quad (8)$$

3.2.3. Sequence generation

Our basic assumption is that, the observation sequence of speaker embeddings $\mathbf{X}$ is generated by distributions that are parameterized by the output of an RNN. This RNN has multiple instantiations, corresponding to different speakers, and they share the same set of RNN parameters $\boldsymbol{\theta}$. In our work, we use gated recurrent unit (GRU) [16] as our RNN model, to memorize long-term dependencies.

At time t, we define $\mathbf{h}_t$ as the state of the GRU corresponding to speaker y_t, and

$$\mathbf{m}_t = f(\mathbf{h}_t | \boldsymbol{\theta}) \quad (9)$$

as the output of the entire network, which may contain other layers. Let $t' := \max\{0, s < t : y_s = y_t\}$ be the last time we saw speaker y_t before t, then:

$$\mathbf{h}_t = \text{GRU}(\mathbf{x}_{t'}, \mathbf{h}_{t'} | \boldsymbol{\theta}), \quad (10)$$
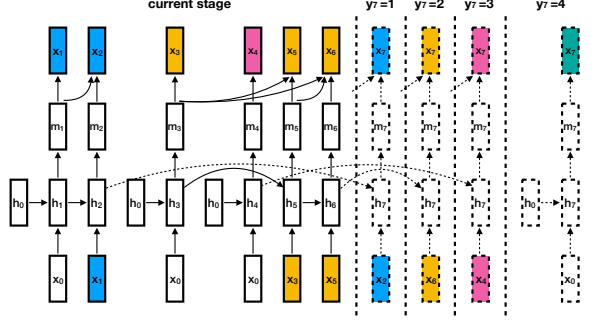
Fig. 2. Generative process of UIS-RNN. Colors indicate labels for speaker segments. There are four options for y_7 given $\mathbf{x}_{[6]}, y_{[6]}$.

where we can assume $\mathbf{x}_0 = \mathbf{0}$ and $\mathbf{h}_0 = \mathbf{0}$, meaning all GRU instances are initialized with the same zero state.

Based on the GRU outputs, we assume the speaker embeddings are modeled by:

$$\mathbf{x}_t | \mathbf{x}_{[t-1]}, y_{[t]} \sim \mathcal{N}(\boldsymbol{\mu}_t, \sigma^2 \mathbf{I}), \tag{11}$$

where $\boldsymbol{\mu}_t = (\sum_{s=1}^{t} \mathbb{1}(y_s = y_t))^{-1} \cdot (\sum_{s=1}^{t} \mathbb{1}(y_s = y_t)\mathbf{m}_s)$ is the averaged GRU output for speaker y_t.

3.2.4. Summary of the model

We briefly summarize UIS-RNN in Fig. 2, where $\mathbf{Z}$ and $\boldsymbol{\lambda}$ are omitted for a simple demonstration. At the current stage (shown in solid lines) $y_{[6]} = (1, 1, 2, 3, 2, 2)$. There are four options for y_7: $1, 2, 3$ (existing speakers), and 4 (a new speaker). The probability for generating a new observation $\mathbf{x}_7$ (shown in dashed lines) depends both on previous label assignment sequence $y_{[6]}$, and previous observation sequence $\mathbf{x}_{[6]}$.

3.3. MLE Estimation

Given a training set $(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N)$ containing N utterances together with their labels $(\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_N)$, we maximize the following log joint likelihood:

$$\max_{\boldsymbol{\theta}, \alpha, \sigma^2, \boldsymbol{\lambda}} \quad \sum_{n=1}^{N} \ln p(\mathbf{X}_n, \mathbf{Y}_n, \mathbf{Z}_n | \boldsymbol{\theta}, \alpha, \sigma^2, \boldsymbol{\lambda}). \tag{12}$$

Here we include all hyper-parameters, and each term in Eq. (12) can be factorized exactly as Eq. (2).

The estimation of $\boldsymbol{\lambda}$ depends on how $g_{\boldsymbol{\lambda}}(\cdot)$ is defined. When we simply have $g_{\boldsymbol{\lambda}}(z_{[t-1]}) = p_0$, we have a closed-form solution:

$$p_0^* = \frac{\sum_{n=1}^{N} \sum_{t=2}^{T_n} \mathbb{1}(\mathbf{x}_{n,t} = \mathbf{x}_{n,t-1})}{\sum_{n=1}^{N} T_n - N}, \tag{13}$$

where T_n denotes the sequence length of the nth utterance.

For $\boldsymbol{\theta}$ and σ^2, there is no closed-form update. We use stochastic gradient ascent by randomly selecting a subset $B^{(\tau)} \subset [N]$ of $|B^{(\tau)}| = b$ utterances. For $\boldsymbol{\theta}$, we update:

$$\boldsymbol{\theta}^{(\tau)} = \boldsymbol{\theta}^{(\tau-1)} + \frac{N\rho^{(\tau)}}{b} \sum_{n \in B^{(\tau)}} \nabla_{\boldsymbol{\theta}} \ln p(\mathbf{X}_n | \mathbf{Y}_n, \mathbf{Z}_n, \boldsymbol{\theta}, -), \tag{14}$$

since $\boldsymbol{\theta}$ is independent of $(\mathbf{Y}_n, \mathbf{Z}_n)$. Eq. (15) also applies to σ^2 by replacing $\boldsymbol{\theta}$ with σ^2. For α, we update

$$\alpha^{(\tau)} = \alpha^{(\tau-1)} + \frac{N\rho^{(\tau)}}{b} \sum_{n \in B^{(\tau)}} \nabla_{\alpha} \ln p(\mathbf{Y}_n | \mathbf{Z}_n, \alpha, -), \tag{15}$$

Data: $\mathbf{X}^{test} = (\mathbf{x}_1^{test}, \mathbf{x}_2^{test}, \ldots, \mathbf{x}_T^{test})$
Result: $\mathbf{Y}^* = (y_1^*, y_2^*, \ldots, y_T^*)$
initialize $\mathbf{x}_0 = \mathbf{0}, \mathbf{h_0} = \mathbf{0}$;
for $t = 1, 2, \ldots, T$ **do**
    $(y_t^*, z_t^*) = \arg\max_{(y_t, z_t)} \big(\ln p(z_t)$ Eq. (6)
        $+ \ln p(y_t | z_t, y_{[t-1]}^*)$ Eq. (5, 7)
        $+ \ln p(\mathbf{x}_t | \mathbf{x}_{[t-1]}, y_{[t-1]}^*, y_t))$ Eq. (11)
    update $N_{k,t-1}$ and GRU hidden states;
end
Algorithm 1: Online greedy MAP decoding for UIS-RNN.

where $p(\mathbf{Y}_n | \mathbf{Z}_n, \alpha, -)$ is given in Eq. (8). In our experiments, we run multiple iterations with a constant step size $\rho^{(\tau)} = \rho$ until convergence.

3.4. MAP Decoding

Since we can decode each testing utterance in parallel, here we assume we are given a testing utterance $\mathbf{X}^{test} = (\mathbf{x}_1, \mathbf{x}_2 \ldots, \mathbf{x}_T)$ without labels. The ideal goal is to find

$$\mathbf{Y}^* = \arg\max_{\mathbf{Y}} \ \ln p(\mathbf{X}^{test}, \mathbf{Y}). \tag{16}$$

However, this requires an exhaustive search over the entire combinatorial label space with complexity $\mathcal{O}(T!)$, which is impractical. Instead, we use an online decoding approach which sequentially performs a greedy search, as shown in Alg. 1. This will significantly reduce computational complexity to $\mathcal{O}(T^2)$. We observe that in most cases the maximum number of speakers per-utterance is bounded by a constant C. In that case, the complexity will further reduce to $\mathcal{O}(T)$. In practice, we apply a beam search [17] on the decoding algorithm, and adjust the number of look-ahead entries to achieve better decoding results.

4. EXPERIMENTS

4.1. Speaker recognition model

We have been retraining the speaker recognition network with more data and minor tricks to improve its performance. Let's call the text-independent speaker recognition model in [3, 6, 18] as "d-vector V1". This model is trained with 36M utterances from 18K US English speakers, which are all mobile phone data based on anonymized voice query logs.

To train a new version of the model, which we call "d-vector V2" [19], we added: (1) non-US English speakers; (2) data from far-field devices; (3) public datasets including LibriSpeech [20], VoxCeleb [21], and VoxCeleb2 [22]. The non-public part contains 34M utterances from 138K speakers, while the public part is added to the training process using the MultiReader approach [6].

Another minor but important trick is that, the speaker recognizer model used in [3] and [6] are trained on windows of size 1600ms, which causes performance degradation when we run inference on smaller windows. For example, in the diarization system, the window size is only 240ms. Thus we have retrained a new model "d-vector V3" by using variable-length windows, where the window size is drawn from a uniform distribution within $[240\text{ms}, 1600\text{ms}]$ during training.

The speaker verification Equal Error Rate (EER) of the three models on two testing sets are shown in Table 1. On speaker verification tasks, adding more training data has significantly improved

Table 1. Speaker verification EER of the three speaker recognition models. en-ALL represents all English locales. The EER=3.55% for d-vector V1 on en-US phone data is the same as the number reported in Table 3 of [6].

Model	EER (%) on en-US phone data	EER (%) on en-ALL phone + farfield data
d-vector V1	3.55	6.14
d-vector V2	3.06	2.03
d-vector V3	3.03	1.91

the performance, while using variable-length windows for training also slightly further improved EER.

4.2. UIS-RNN setup

For the speaker change, as we have stated in Section 3.2.1, we assume $\{z_t\}_{t \in [2,T]}$ follow independent identical binary distributions for simplicity.

Our sequence generation model is composed of one layer of 512 GRU cells with a tanh activation, followed by two fully-connected layers each with 512 nodes and a ReLU [23] activation. The two fully-connected layers corresponds to Eq. (9).

For decoding, we use beam search of width 10.

4.3. Evaluation protocols

Our evaluation setup is exactly the same as [3], which is based on the pyannote.metrics library [24]. We follow these common conventions of other works:

- We evaluate on single channel audio.
- We exclude overlapped speech from evaluation.
- We tolerate errors less than 250ms in segment boundaries.
- We report the confusion error, which is usually directly referred to as Diarization Error Rate (DER) in the literature.

4.4. Datasets

For the evaluation, we use 2000 NIST Speaker Recognition Evaluation (LDC2001S97), Disk-8, which is usually directly referred to as "CALLHOME" in literature. It contains 500 utterances distributed across six languages: Arabic, English, German, Japanese, Mandarin, and Spanish. Each utterance contains 2 to 7 speakers.

Since our approach is supervised, we perform a 5-fold cross validation on this dataset. We randomly partition the dataset into five subsets, and each time leave one subset for evaluation, and train UIS-RNN on the other four subsets. Then we combine the evaluation on five subsets and report the averaged DER.

Besides, we also tried to use two off-domain datasets for training UIS-RNN: (1) 2000 NIST Speaker Recognition Evaluation, Disk-6, which is often referred to as "Switchboard"; (2) ICSI Meeting Corpus [25]. We first tried to train UIS-RNN purely on off-domain datasets, and evaluate on CALLHOME; we then tried to add the off-domain datasets to the training partition of each of the 5-fold.

4.5. Results

We report the diarization performance results on 2000 NIST SRE Disk-8 in Table 2. For each version of the speaker recognition model, we compare UIS-RNN with two baseline approaches: k-means and spectral offline clustering. For k-means and spectral clustering, the number of speakers is adaptively determined as in [3]. For UIS-RNN, we show results for three types of evaluation settings: (1) in-domain training (5-fold); (2) off-domain training (Disk-6 + ICSI); and (3) in-domain plus off-domain training.

Table 2. DER on NIST SRE 2000 CALLHOME, with comparison to other systems in literature. VB is short for Variational Bayesian resegmentation [1]. The DER=12.0% for d-vector V1 and spectral clustering is the same as the number reported in Table 2 of [3].

d-vector	Method	Training data	DER (%)
V1	k-means	—	17.4
	spectral	—	12.0
	UIS-RNN	5-fold	**11.7**
	UIS-RNN	Disk-6 + ICSI	**11.7**
	UIS-RNN	5-fold + Disk-6 + ICSI	**10.6**
V2	k-means	—	19.1
	spectral	—	11.6
	UIS-RNN	5-fold	**10.9**
	UIS-RNN	Disk-6 + ICSI	**10.8**
	UIS-RNN	5-fold + Disk-6 + ICSI	**9.6**
V3	k-means	—	12.3
	spectral	—	8.8
	UIS-RNN	5-fold	**8.5**
	UIS-RNN	Disk-6 + ICSI	**8.2**
	UIS-RNN	5-fold + Disk-6 + ICSI	**7.6**
Castaldo *et al.* [4]			13.7
Shum *et al.* [9]			14.5
Senoussaoui *et al.* [10]			12.1
Sell *et al.* [1] (+VB)			13.7 (11.5)
Garcia-Romero *et al.* [2] (+VB)			12.8 (9.9)

From the table, we see that the biggest improvement in DER actually comes from upgrading the speaker recognition model from V2 to V3. This is because in V3, we have the window size consistent between training time and diarization inference time, which was a big issue in V1 and V2.

UIS-RNN performs noticeably better than spectral offline clustering, when using the same speaker recognition model. It is also important to note that UIS-RNN inference produces speaker labels in an **online** fashion. As discussed in [3], online unsupervised clustering algorithms usually perform significantly worse than offline clustering algorithms such as spectral clustering.

Also, adding more data to train UIS-RNN also improved DER, which is consistent with our expectation – UIS-RNN benefits from learning from more examples. Specifically, while large scale off-domain training already produces great results in practice (Disk-6 + ICSI), the availability of in-domain data can further improve the performance (5-fold + Disk-6 + ICSI).

5. CONCLUSIONS

In this paper, we presented a speaker diarization system where the commonly used clustering module is replaced by a trainable unbounded interleaved-state RNN. Since all components of this system can be learned in a supervised manner, it is preferred over unsupervised systems in scenarios where training data with high quality time-stamped speaker labels are available. On the NIST SRE 2000 CALLHOME benchmark, using exactly the same speaker embeddings, this new approach, which is an online algorithm, out-performs the state-of-the-art spectral offline clustering algorithm.

Besides, the proposed UIS-RNN is a generic solution to the sequential clustering problem, with other potential applications such as face clustering in videos. One interesting future work direction is to directly use accoustic features instead of pre-trained embeddings as the observation sequence for UIS-RNN, such that the entire speaker diarization system becomes an end-to-end model.

6. REFERENCES

[1] Gregory Sell and Daniel Garcia-Romero, "Diarization re-segmentation in the factor analysis subspace," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4794–4798.

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

[3] Quan Wang, Carlton Downey, Li Wan, Philip Andrew Mansfield, and Ignacio Lopz Moreno, "Speaker diarization with lstm," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5239–5243.

[4] Fabio Castaldo, Daniele Colibro, Emanuele Dalmasso, Pietro Laface, and Claudio Vair, "Stream-based speaker segmentation using speaker factors and eigenvoices," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2008, pp. 4133–4136.

[5] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre 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] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno, "Generalized end-to-end loss for speaker verification," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4879–4883.

[7] Zbyněk Zajíc, Marek Hrúz, and Luděk Müller, "Speaker diarization using convolutional neural network for statistics accumulation refinement," in *INTERSPEECH*, 2017.

[8] Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer, "End-to-end text-dependent speaker verification," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5115–5119.

[9] Stephen H Shum, Najim Dehak, Réda Dehak, and James R Glass, "Unsupervised methods for speaker diarization: An integrated and iterative approach," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2015–2028, 2013.

[10] Mohammed Senoussaoui, Patrick Kenny, Themos Stafylakis, and Pierre Dumouchel, "A study of the cosine distance-based mean shift for telephone speech diarization," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, no. 1, pp. 217–227, 2014.

[11] Gregory Sell and Daniel 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.

[12] Dimitrios Dimitriadis and Petr Fousek, "Developing on-line speaker diarization system," pp. 2739–2743, 2017.

[13] Philip Andrew Mansfield, Quan Wang, Carlton Downey, Li Wan, and Ignacio Lopez Moreno, "Links: A high-dimensional online clustering method," *arXiv preprint arXiv:1801.10123*, 2018.

[14] Huazhong Ning, Ming Liu, Hao Tang, and Thomas S Huang, "A spectral clustering approach to speaker diarization.," in *INTERSPEECH*, 2006.

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

[16] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[17] Mark F. Medress, Franklin S Cooper, Jim W. Forgie, CC Green, Dennis H. Klatt, Michael H. O'Malley, Edward P Neuburg, Allen Newell, DR Reddy, B Ritea, et al., "Speech understanding systems: Report of a steering committee," *Artificial Intelligence*, vol. 9, no. 3, pp. 307–316, 1977.

[18] Ye Jia, Yu Zhang, Ron J Weiss, Quan Wang, Jonathan Shen, Fei Ren, Zhifeng Chen, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, et al., "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *Conference on Neural Information Processing Systems (NIPS)*, 2018.

[19] Quan Wang, Hannah Muckenhirn, Kevin Wilson, Prashant Sridhar, Zelin Wu, John Hershey, Rif A. Saurous, Ron J. Weiss, Ye Jia, and Ignacio Lopez Moreno, "Voicefilter: Targeted voice separation by speaker-conditioned spectrogram masking," *arXiv preprint arXiv:1810.04826*, 2018.

[20] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[21] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[22] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman, "Voxceleb2: Deep speaker recognition," *arXiv preprint arXiv:1806.05622*, 2018.

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

[24] Hervé Bredin, "`pyannote.metrics`: a toolkit for reproducible evaluation, diagnostic, and error analysis of speaker diarization systems," *hypothesis*, vol. 100, no. 60, pp. 90, 2017.

[25] Adam Janin, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, et al., "The icsi meeting corpus," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2003, vol. 1, pp. I–I.