Fully Supervised Speaker Diarization

Say Goodbye to clustering

Aonan Zhang, Quan Wang, Zhenyao Zhu, John Paisley, Chong Wang
Google Inc., Columbia University
az2385@columbia.edu · quanw@google.com

google/uis-rnn

Overview

- Most existing speaker diarization systems are based on unsupervised clustering approaches, such as k-means or hierarchical clustering.
- We propose UIS-RNN, a trainable model for segmenting and clustering temporal data by learning from examples.
- New state-of-the-art on CALLHOME, while decoding is online.

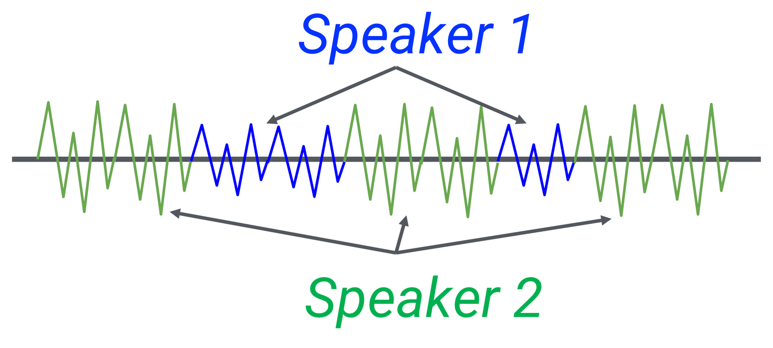


Fig. Speaker diarization solves the problem of "who spoke when".

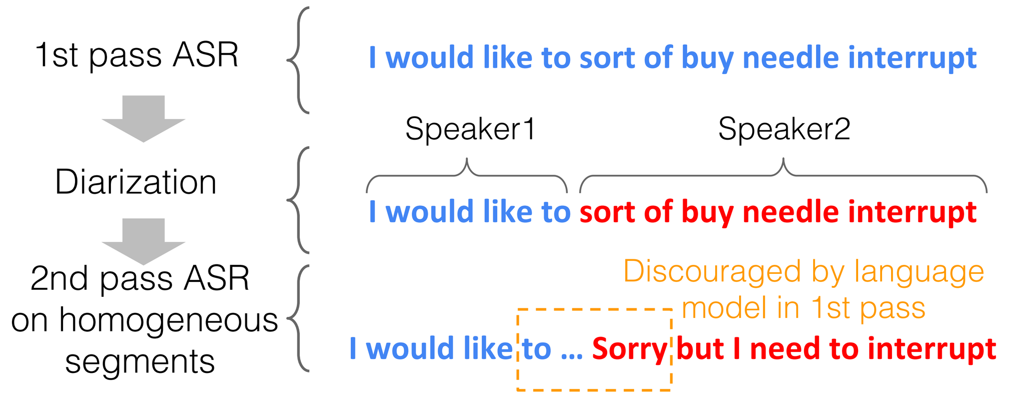


Fig. Example application: Improve ASR with diarization results.

Baseline Diarization System

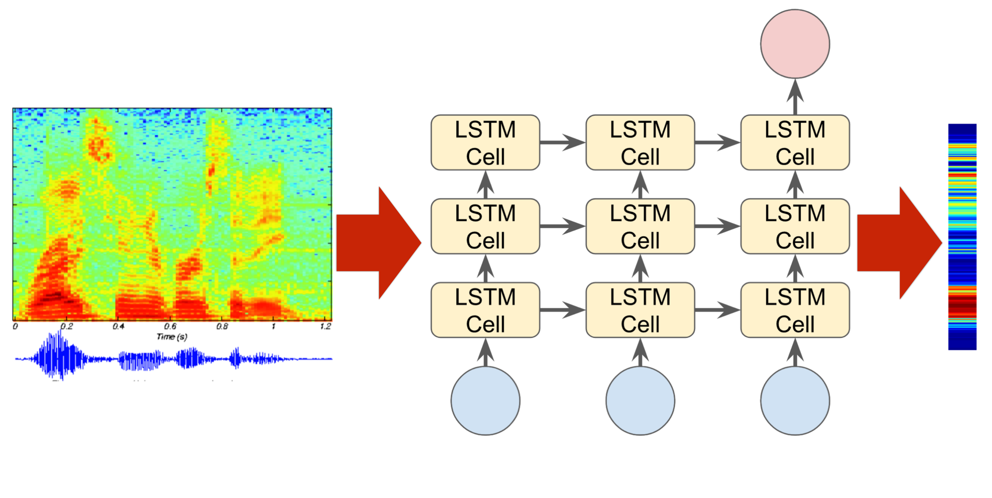


Fig. Multi-layer LSTM network as speaker encoder.

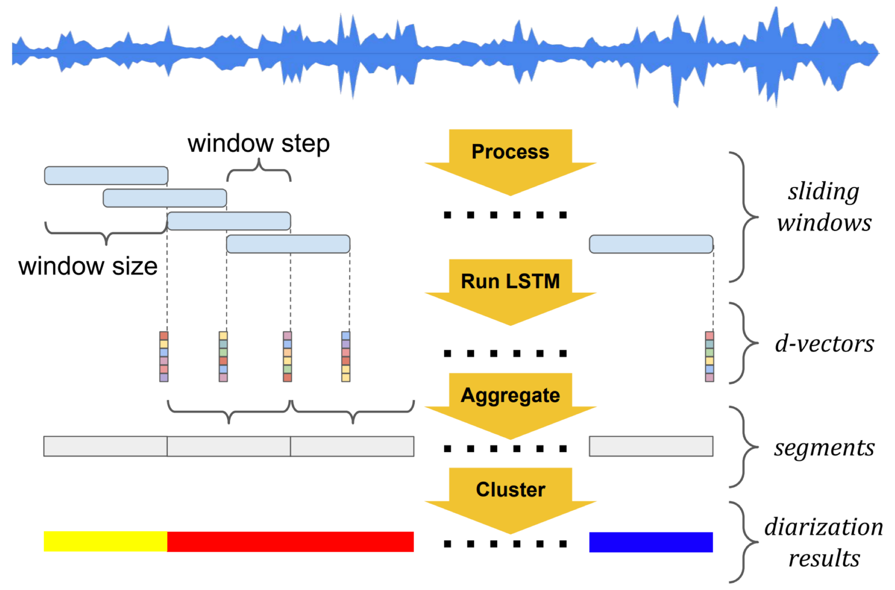


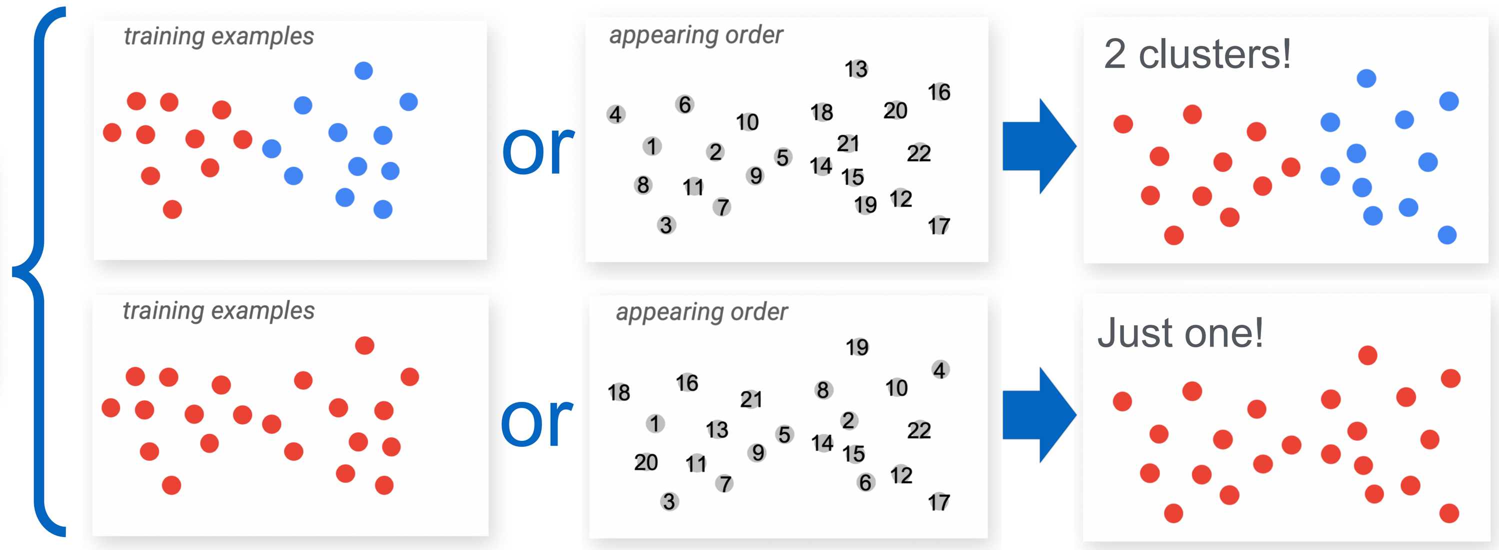
Fig. Baseline diarization system using d-vectors and unsupervised clustering.

- Speaker encoder is trained with "Generalized End-to-End Loss for Speaker Verification", ICASSP 2018. Proven to be better than softmax or triplet loss.
- Speaker embeddings (d-vectors) are extracted on sliding windows of length 240ms with 50% overlap, using log-mel-filterbank energies as features.
- Window-wise d-vectors are aggregated on non-overlapping segments. Segments are determined by VAD and a maximal length limit of 400ms.
- A modified version of spectral clustering on segment-wise embeddings, using eigen-gap for number of speakers, produces state-of-the-art performance.
- This baseline system is described in "Speaker Diarization with LSTM", ICASSP 2018. A lecture is available on YouTube

Clustering is Not Good Enough

How many clusters are there?

Difficult to answer without supervised or temporal information.



UIS-RNN

- We model the **generative process** of the speaker embedding sequence:

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

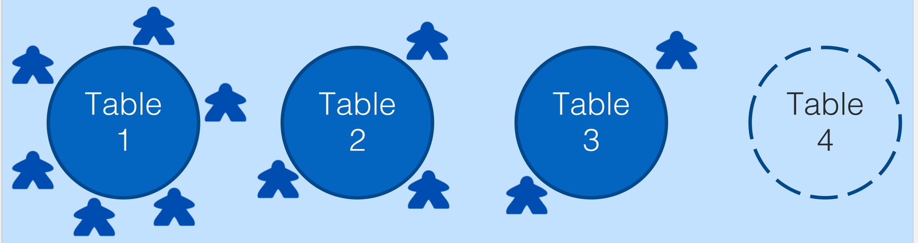
$\{\mathbf{x}_t\}$: embedding sequence
$\{y_t\}$: label sequence
$\{z_t\}$: binary speaker changes

Speaker change

- I.I.D. coin flipping distribution: $p(z_t = 0 | z_{[t-1]}, \boldsymbol{\lambda}) = p_0$

Speaker assignment

- Speaker labels are assigned using Chinese restaurant process (CRP)



Sequence generation

- Each speaker is modeled by an RNN instance, all sharing same parameters.
- Each instance has its own states. States of different speakers interleave in the time domain.
- Each embedding follows a Gaussian, where the mean is this speaker's average RNN output so far.

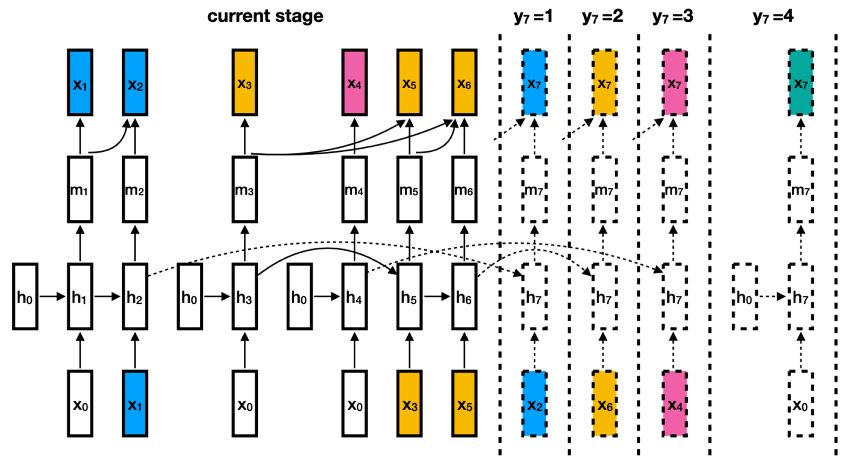


Fig (Left). Four possible generative paths in an example sequence.

Fig (Right). Online greedy MAP decoding algorithm.

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**
$\quad (y_t^*, z_t^*) = \arg\max_{(y_t, z_t)} \big(\ln p(z_t) + \ln p(y_t | z_t, y_{[t-1]}^*) + \ln p(\mathbf{x}_t | \mathbf{x}_{[t-1]}, y_{[t-1]}^*, y_t) \big)$
$\quad$ update $N_{k,t-1}$ and GRU hidden states;
end

Experiment Results

- Datasets: NIST SRE 2000 Disk-8 (CALLHOME), Disk-6, and ICSI.
- UIS-RNN requires training, so we evaluated with three different setups: in-domain training, off-domain training, and in-domain plus off-domain training.

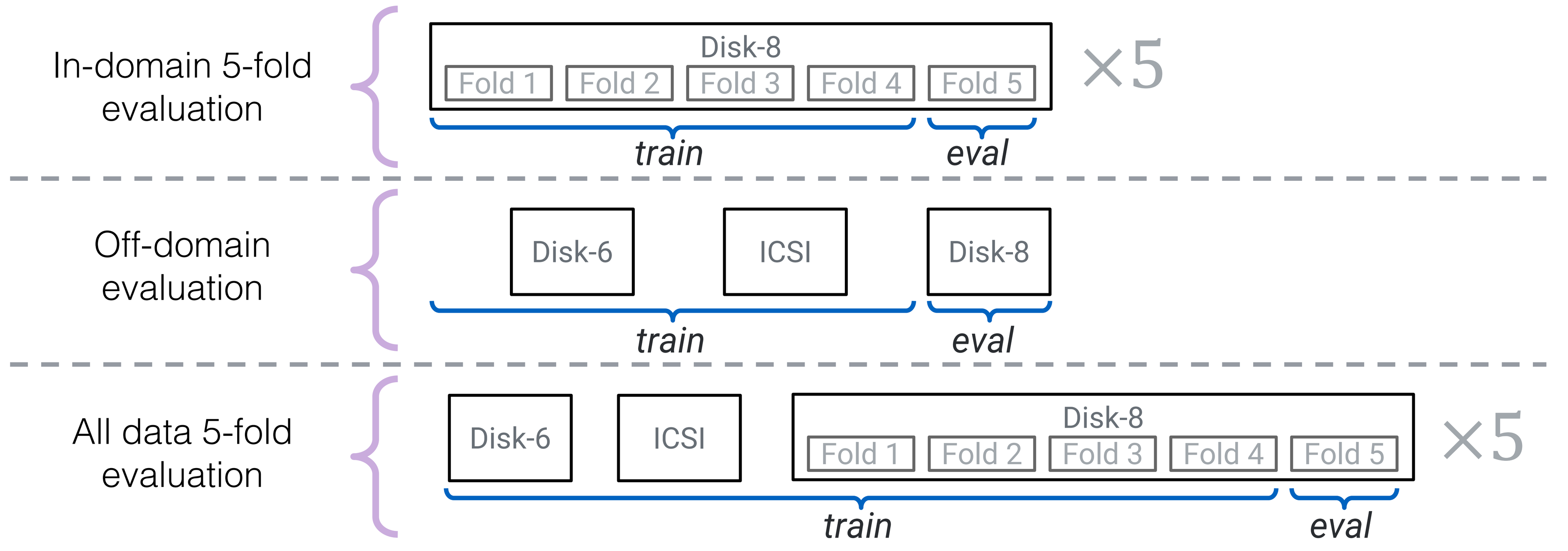


Table. DER (%) on NIST SRE 2000 Disk-8 (CALLHOME), compared with other teams' work. VB for Variational Bayesian resegmentation.

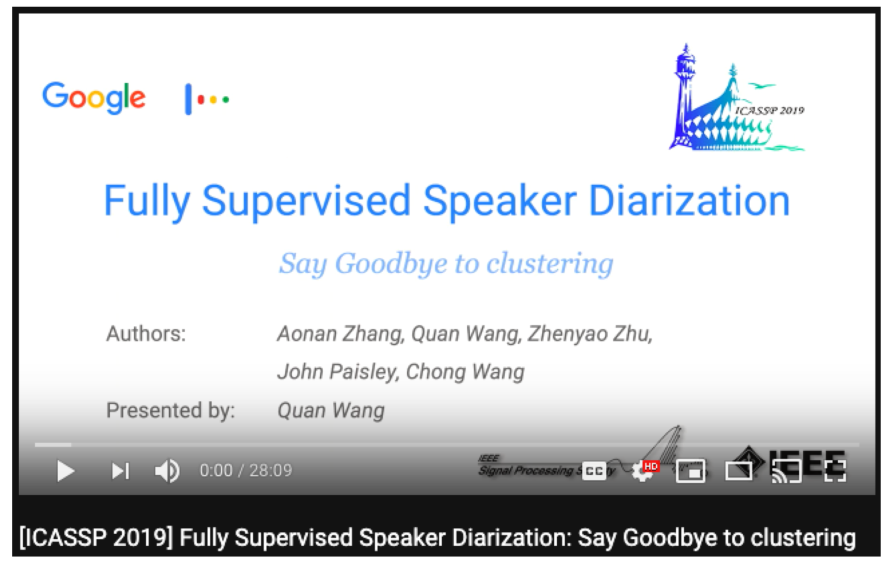
Method	Training data	DER (%)
k-means	–	12.3
spectral	–	8.8
UIS-RNN	5-fold	**8.5**
UIS-RNN	Disk-6 + ICSI	**8.2**
UIS-RNN	5-fold + Disk-6 + ICSI	**7.6**
Castaldo *et al.*		13.7
Shum *et al.*		14.5
Senoussaoui *et al.*		12.1
Sell *et al.* (+VB)		13.7 (11.5)
Garcia-Romero *et al.* (+VB)		12.8 (9.9)

Conclusions:

- Supervised diarization is helpful, when we have in-domain training data with timestamped speaker labels.
- What is learned by UIS-RNN: (1) Dialogue styles; (2) Domain-specific hints for speaker turns.
- Future work: (1) Speaker change: coin flipping → RNN; (2) Unlabeled data as part of training set; (3) Offline decoding to further improve quality.

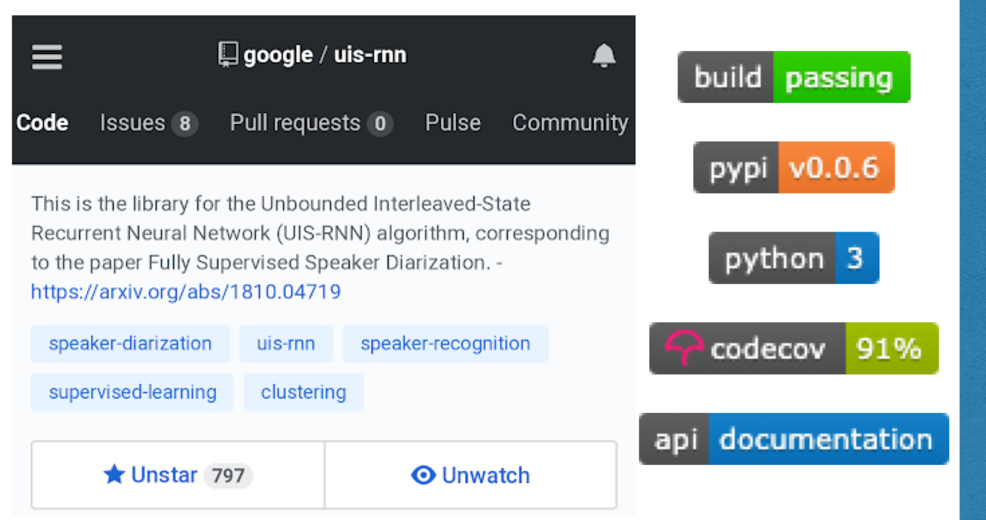
More Information

Full lecture is available on YouTube



Core algorithm on GitHub

https://github.com/google/uis-rnn



ICASSP 2019

The 44th International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 12 - 17 May, 2019 · Brighton, UK

IEEE Signal Processing Society

IEEE