DIHARD 3
Diarization Challenge

USC SAIL Team

Taejin Park (Presenter)
Raghuveer Peri, Arindam Jati, Shrikanth Narayanan


DIHARD 3
USC_SAIL

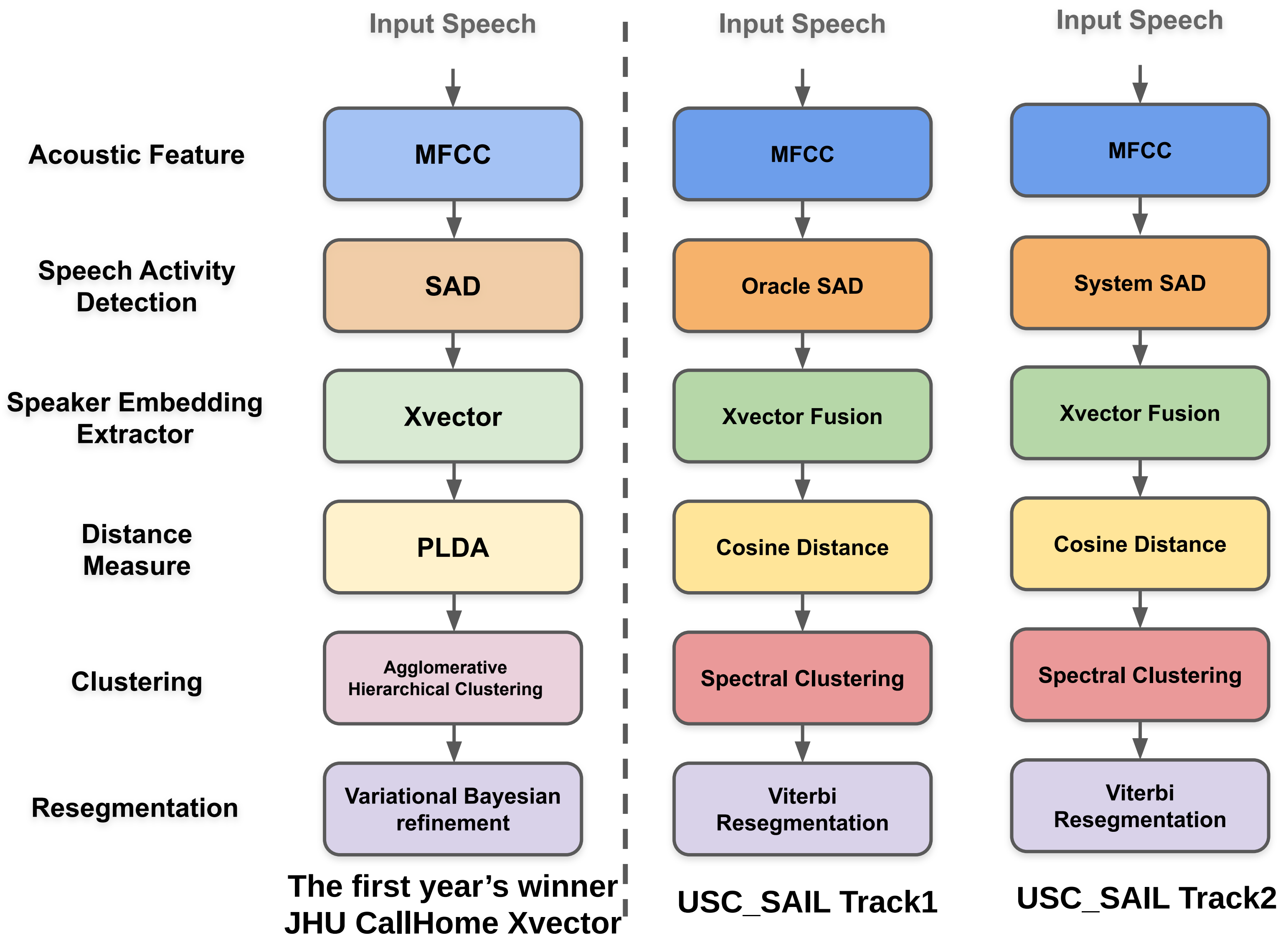
Who is USC_SAIL?

- **University of Southern California (USC)**, Los Angeles
- Meing Hsieh Electrical Engineering Dept., Signal and Image Processing Institute (SIPI)
- Research Group Name: Signal Analysis and Interpretation Laboratory (SAIL)
- Supervision under Professor **Shrikanth Narayanan**
- SAIL focuses on human-centered signal & information processing
- Homepage: https://sail.usc.edu/
- **Taejin Park (Presenter)**, Raghuveer Peri, Arindam Jati, Shrikanth Narayanan

DIHARD 2 SAIL USC

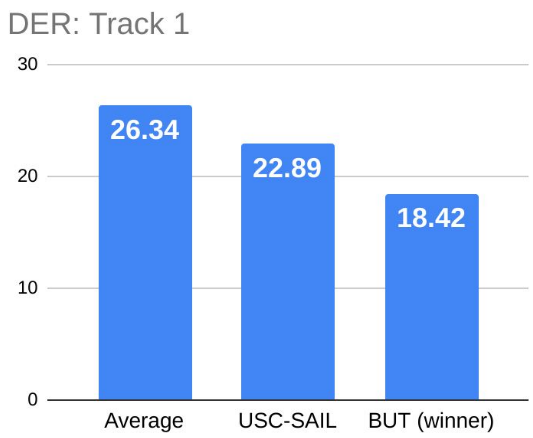
Recap

The Third DIHARD Speech Diarization Challenge

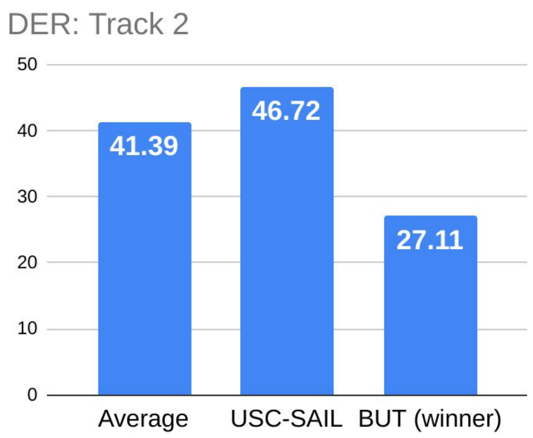


DIHARD2 Challenge 2019 Final results

9th place out of 23 teams for Track 1



12-th place for Track 2



Final results:

9-th place as a team for Track 1

12-th place as a team for Track 2

Discussion: Experimental Approaches

1. **PLDA adaptation**
   a. The performance of PLDA on dev-set data is not consistent with eval-set.
   b. Adapting PLDA to huge acoustic variability of DIHARD dataset is very challenging.

2. **Embedding Denoising did not improve the performance**
   a. Directly applied to the embedding level rather than acoustic signal.
   b. Low SNR embedding can hardly be denoised in embedding level.

3. **Overlap Detection did not work for low SNR samples**
   a. Competitive performance on high SNR utterances.
   b. Low SNR utterances heavily degrades the overlap detection performance.

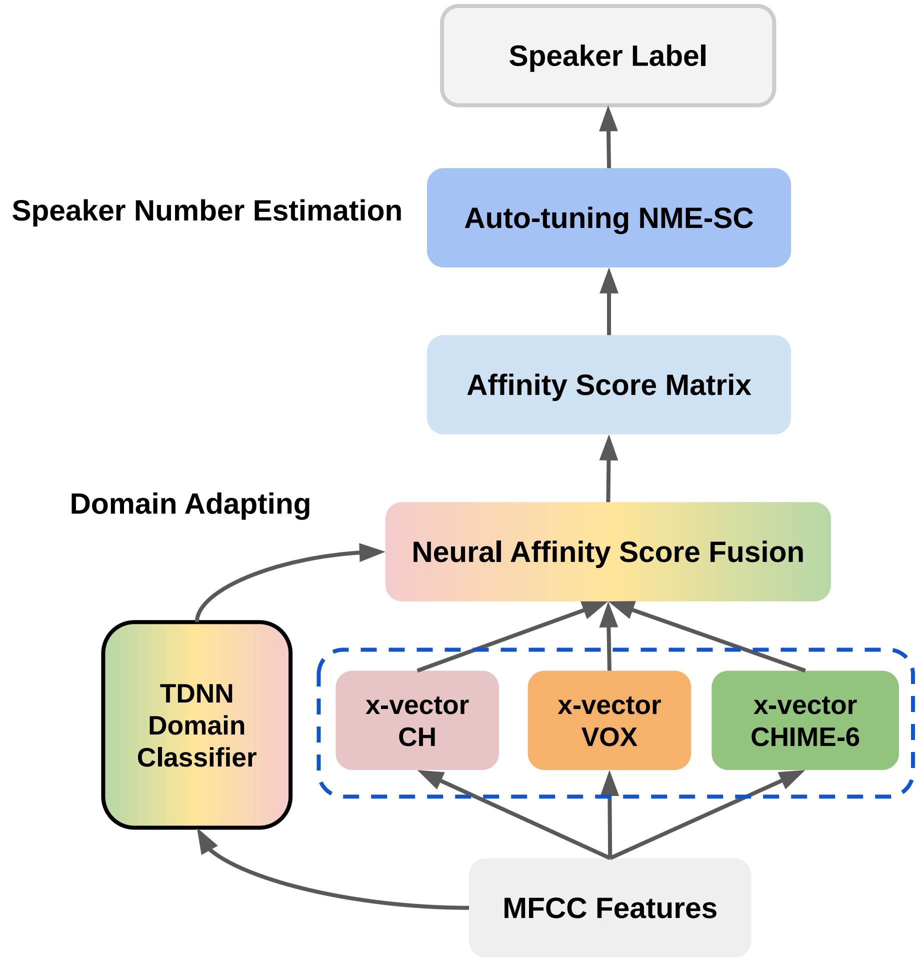
Discussion: Techniques that improved the performance

1. **Cosine Similarity + Spectral Clustering**
   a. Cosine similarity is free from adaptation issues
   b. Spectral clustering shows better performance when coupled with cosine similarity.

2. **The Fusion of two embedding extractors**
   a. Xvector trained on 8K CallHome + Xvector trained on 16K VoxCeleb
   b. The sum of cosine similarity compensates the poor performance obtained by single model.

3. **Viterbi resegmentation**
   a. Resegmentation can mitigate the effect of uniform length segmentation.
   b. Shows consistent improvement especially on system SAD (Track 2) setup

USC_SAIL 2020 DIHARD 3 System



Auto-tuning Spectral Clustering

- No manual parameter tuning on dev-set
- Clustering parameter varies over session

Domain Adaptive Affinity Fusion

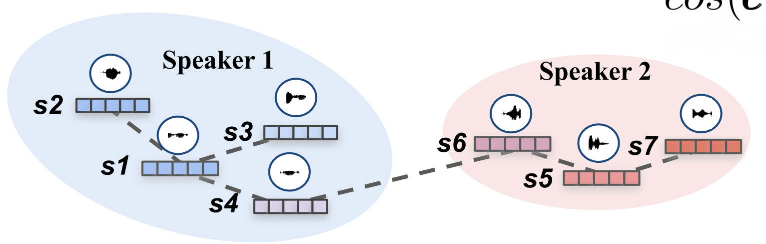
- Soft decision on affinity weight selection
- Works better than hard decision method

Auto-tuning Spectral Clustering for Speaker Diarization (Taejin Part et al.)

Does not require (1) PLDA (training) (2) Development Set

$$cos(\boldsymbol{e}_1, \boldsymbol{e}_2) = \frac{\boldsymbol{e}_1 \cdot \boldsymbol{e}_2}{||\boldsymbol{e}_1|| \cdot ||\boldsymbol{e}_2||}$$

p-Binarization

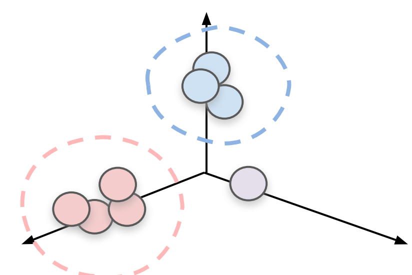


Affinity calculation

	s_1	s_2	s_3	s_4	s_5	s_6	s_7
s_1	1	1	1	1	0	0	0
s_2	1	1	0	0	0	0	0
s_3	1	0	1	0	0	0	0
s_4	1	0	0	1	0	1	0
s_5	0	0	0	0	1	1	1
s_6	0	0	0	1	1	1	0
s_7	0	0	0	0	1	0	1

$$\begin{aligned} d_i &= \textstyle\sum_{k=1}^{N} a_{ik} \\ \mathbf{D}_p &= \mathrm{diag}\{d_1, d_2, ..., d_N\} \\ \mathbf{L}_p &= \mathbf{D}_p - \bar{\mathbf{A}}_p, \end{aligned}$$

Unnormalized Laplacian matrix



K-means of eigen vectors

$$\mathbf{L}_p = \mathbf{U}_p \mathbf{\Sigma}_p \mathbf{V}_p^T$$

[1] Taejin Park et. al. “Auto-Tuning Spectral Clustering for Speaker Diarization Using Normalized Maximum Eigengap” IEEE SPL. 2019, p.381-385.

Auto-tuning Spectral Clustering for Speaker Diarization (Taejin Part et al.)

What is p-binarization ?

[Example] p=4

	S1	S2	S3	S4	S5	S6	S7
S1	1	0.7	0.5	0.8	0.1	0.2	0.1
S2							
S3							

Cosine distances of embeddings between the segment pairs

	S1	S2	S3	S4	S5	S6	S7
S1	1	0.7	0.5	0.8	0.1	0.2	0.1
S2							
S3							

Find the p-biggest values

	S1	S2	S3	S4	S5	S6	S7
S1	1	1	1	1	0	0	0
S2							
S3							

Convert the selected values to 1
Convert the other values to 0

- p-binarization makes the affinity matrix **focus only on the extremely prominent values**
- BUT, without any proper strategy, p-value should be **optimized on a dev-set**

(a p-value that gives the lowest DER is selected)

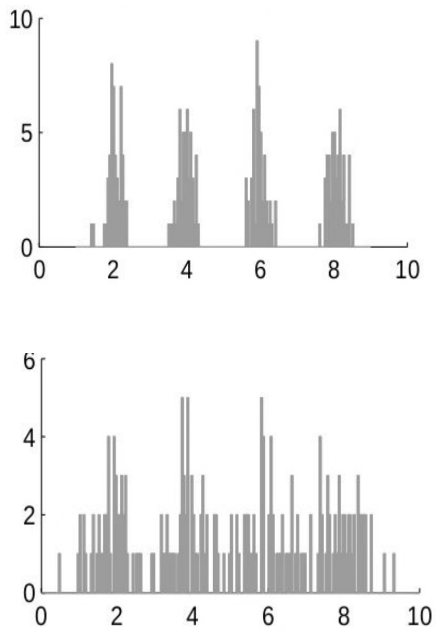
Is there any way we can determine p-value without dev-set?

[2] Taejin Park et. al. “Auto-Tuning Spectral Clustering for Speaker Diarization Using Normalized Maximum Eigengap” IEEE SPL. 2019, p.381-385.

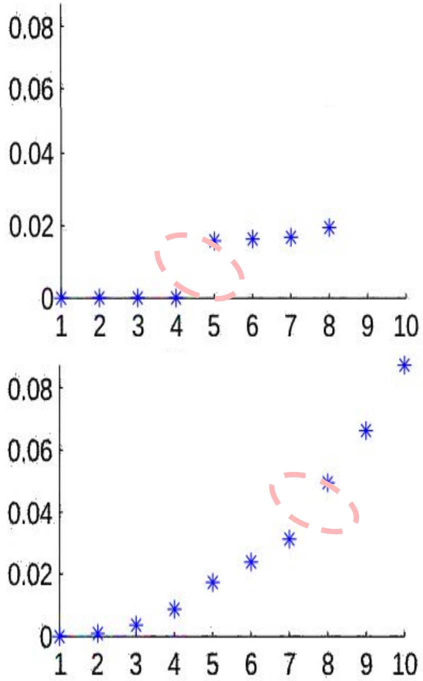
Auto-tuning Spectral Clustering for Speaker Diarization (Taejin Part et al.)

Eigengaps and Clusters: Number of speakers can be estimated by the **maximum eigengap**.

- Eigenvalues



- Eigenvalues and eigengaps



The Maximum Eigengap ∝ Cluster Clarity

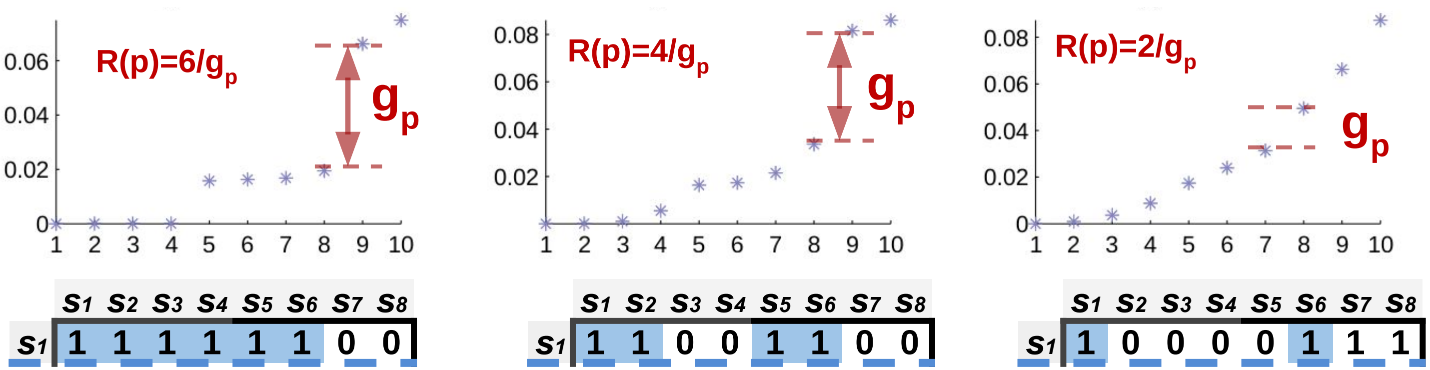
Auto-tuning Spectral Clustering for Speaker Diarization (Taejin Part et al.)

Relationship between cluster clarity and eigen gap size

- Eigenvalues

$$\begin{aligned} d_i &= \textstyle\sum_{k=1}^{N} a_{ik} \\ \mathbf{D}_p &= \mathrm{diag}\{d_1, d_2, \ldots, d_N\} \\ \mathbf{L}_p &= \mathbf{D}_p - \bar{\mathbf{A}}_p, \end{aligned} \rightarrow \mathbf{L}_p = \mathbf{U}_p \mathbf{\Sigma}_p \mathbf{V}_p^T \rightarrow \mathbf{e}_p = [\lambda_{p,2} - \lambda_{p,1}, \cdots, \lambda_{p,N} - \lambda_{p,N-1}]$$

- Eigengaps of the same data but different p values

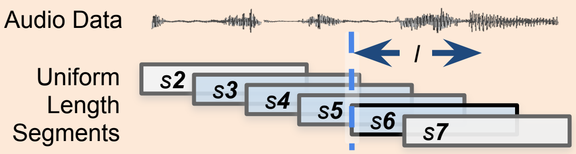


p=6 p=4 p=2

- As p gets bigger the maximum eigengap also increases.
- We focus on the ratio between p and the maximum eigengap size, r(p).
- But the relationship is not linear! → r(p) is not constant

Auto-tuning Spectral Clustering for Speaker Diarization (Taejin Part et al.)

Audio Segmentation



Binarization

$$\mathbf{A}_p = binarize(\mathbf{A}, p)$$

Symmetrization

$$\bar{\mathbf{A}}_p = \frac{1}{2}(\mathbf{A}_p + \mathbf{A}_p^T)$$

	s1	s2	s3	s4	s5	s6	s7	s8
s1	1	1	0	0	0	0	1	1
s2	1	1	0	0.5	0	0	1	0.5
s3	0	0	1	0	1	0.5	0.5	0
s4	0	0.5	0	1	0.5	1	0	0
s5	0	0	1	0.5	1	1	0.5	0
s6	0	0	0.5	1	1	1	0.5	0.5
s7	1	1	0.5	0	0.5	0.5	1	1
s8	1	0.5	0	0	0	0.5	1	1

Unnormalized Laplacian matrix

$$d_i = \sum_{k=1}^{N} a_{ik}$$

$$\mathbf{D}_p = \mathrm{diag}\{d_1, d_2, ..., d_N\}$$

$$\mathbf{L}_p = \mathbf{D}_p - \bar{\mathbf{A}}_p,$$

Eigenvalue Decomposition (EVD)

$$\mathbf{L}_p = \mathbf{U}_p \mathbf{\Sigma}_p \mathbf{V}_p^T$$

Eigengap Vector

$$\mathbf{e}_p = [\lambda_{p,2} - \lambda_{p,1}, \cdots, \lambda_{p,N} - \lambda_{p,N-1}]$$

Normalized maximum eigengap (NME)

$$g_p = \frac{\max(\mathbf{e}_p)}{\lambda_{p,N} + \epsilon}$$

Ratio between binarization parameter p and NME value g_p

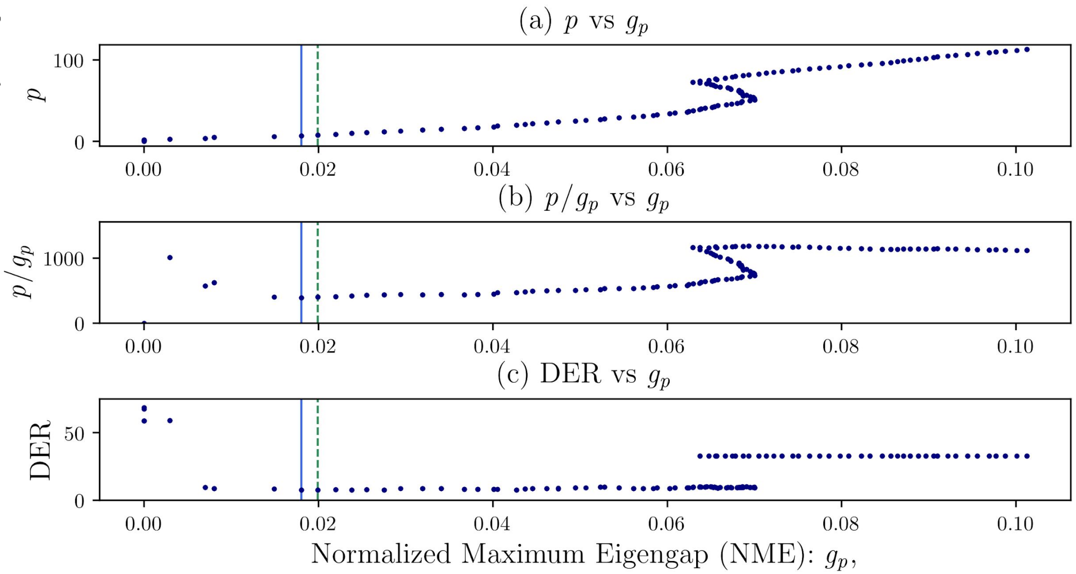
$$r(p) = \frac{p}{g_p}$$

Auto-tuning Spectral Clustering for Speaker Diarization (Taejin Part et al.)

Algorithm 1 NME-SC algorithm

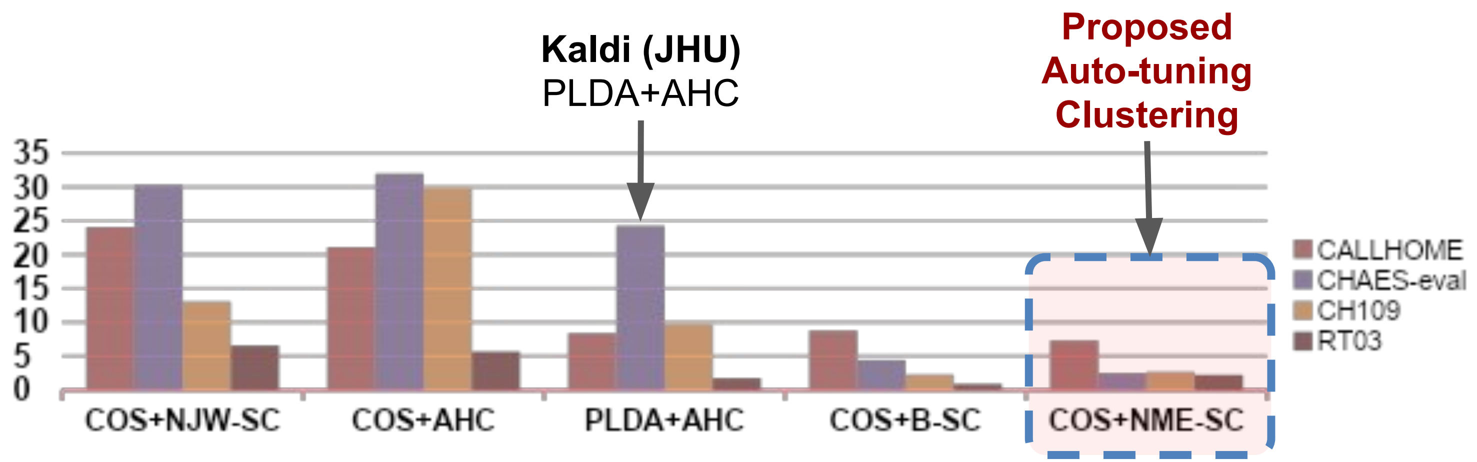
Input: Affinity Matrix $\mathbf{A}$
Output: Cluster vector $\mathbf{C}$

procedure *NME-SC*($\mathbf{A}$)
  for $p \leftarrow 1$ to P **do**
    $\mathbf{A}_p \leftarrow \mathit{binarize}(\mathbf{A}, p)$
    $\bar{\mathbf{A}}_p \leftarrow (\mathbf{A}_p + \mathbf{A}_p^T)/2$
    $\mathbf{L}_p \leftarrow \mathit{Laplacian}(\bar{\mathbf{A}}_p)$
    $\mathbf{U}_p, \mathbf{\Sigma}_p, \mathbf{V}_p^T \leftarrow \mathit{SVD}(\mathbf{L}_p)$
    $\mathbf{e}_p \leftarrow \mathit{eigengap}(\mathbf{\Sigma}_p)$
    $g_p \leftarrow \mathit{max}(\mathbf{e}_p)/\mathit{max}(\mathbf{\Sigma}_p)$
    $\mathbf{r}[p] \leftarrow p/g_p$
  end for
  $\hat{p} \leftarrow \mathit{argmin}(\mathbf{r})$
  $k \leftarrow \mathit{argmax}(\mathbf{e}_{\hat{p}})$
  $\mathbf{S} \leftarrow \mathbf{U}_{\hat{p}}[1, N; 1, k]^T$
  $\mathbf{C} \leftarrow \textit{k-means}(\mathbf{S}, k)$
  return $\mathbf{C}$
end procedure



- The ratio between p and g_p show the tendency of DER curve with respect to p
- We find the lowest p/g_p value to find a p-value that leads to presumably the lowest DER.

Auto-tuning Spectral Clustering for Speaker Diarization (Taejin Part et al.)



Downside

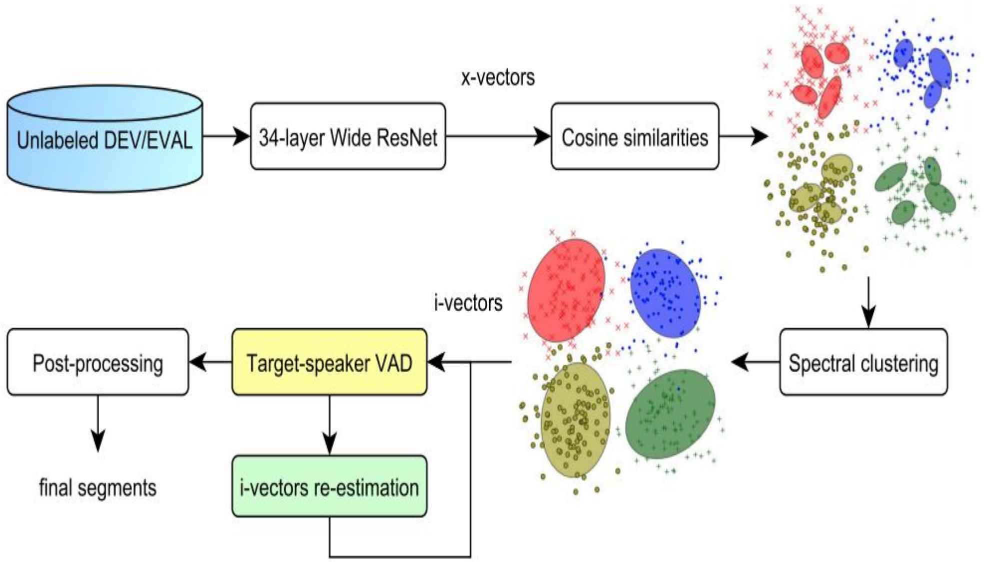
- Calculational complexity
- Hard to be performed in online fashion

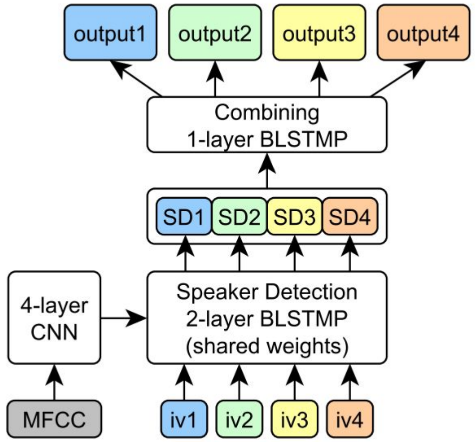
Benefits

- No training is needed for distance measure (No PLDA)
- No parameter tuning is needed.
- **Speaker embedding from NN models can be directly used**
- Superior performance by automatically find the parameter p for each independent session.
- **Appeared in CHIME-6 track 2 Challenge winner's system**

- **Speech Activity Detector (SAD) or Voice Activity Detector (VAD) for track2**

CHIME-6 Track 2 (Diar+ASR) Winner: STC system [1]





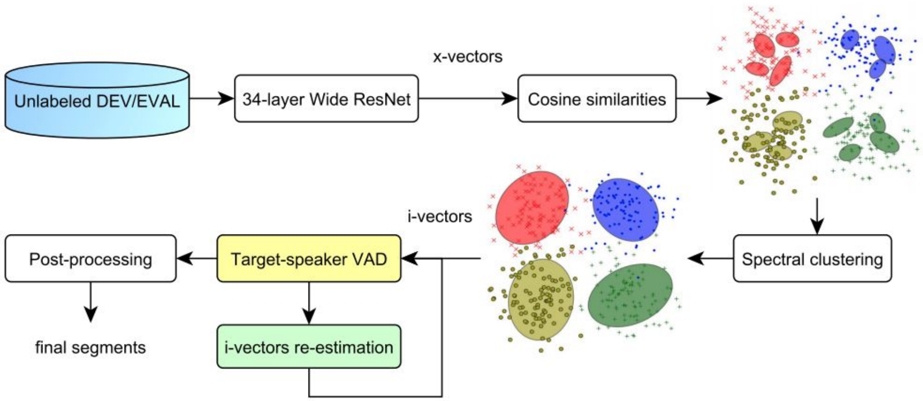
Target Speaker Voice Activity Detector (TS-VAD)

- ResNet inspired x-vectors
- Cosine Similarities with Auto-tuning Spectral Clustering method (NME-SC[2])
- Target-speaker VAD (TS-VAD) greatly improved the overall performance
  - Uses i-vector input from parallel streams of speaker detection (SD) blocks
  - STC's TS-VAD shows that target-speaker VAD can be a solution for overlapping speech

[1] https://chimechallenge.github.io/chime2020-workshop/papers/CHiME_2020_paper_medennikov.pdf
[2] Taejin Park et. al. "Auto-Tuning Spectral Clustering for Speaker Diarization Using Normalized Maximum Eigengap" IEEE SPL. 2019, p.381-385.

Auto-tuning Spectral Clustering for Speaker Diarization (Taejin Part et al.)

NME-SC method in CHIME-6 Challenge Winner’s system



	DEV DER	DEV JER	EVAL DER	EVAL JER
x-vectors + AHC	63.42	70.83	68.20	72.54
EEND + WRN x-vectors	52.20	57.42	56.01	61.49
WRN x-vectors + AHC	53.45	56.76	63.79	62.02
WRN x-vectors + SC	47.29	49.03	60.10	57.99
+ TS-VAD-1C (it1)	39.19	40.87	45.01	47.03
+ TS-VAD-1C (it2)	35.80	37.38	39.80	41.79
+ TS-VAD-MC	34.59	36.73	37.57	40.51
Fusion	**32.84**	**36.31**	**36.02**	**40.10**
Fusion*	41.76	44.04	40.71	45.32

Table 2: *Diarization results (* stands for DIHARD II reference)*

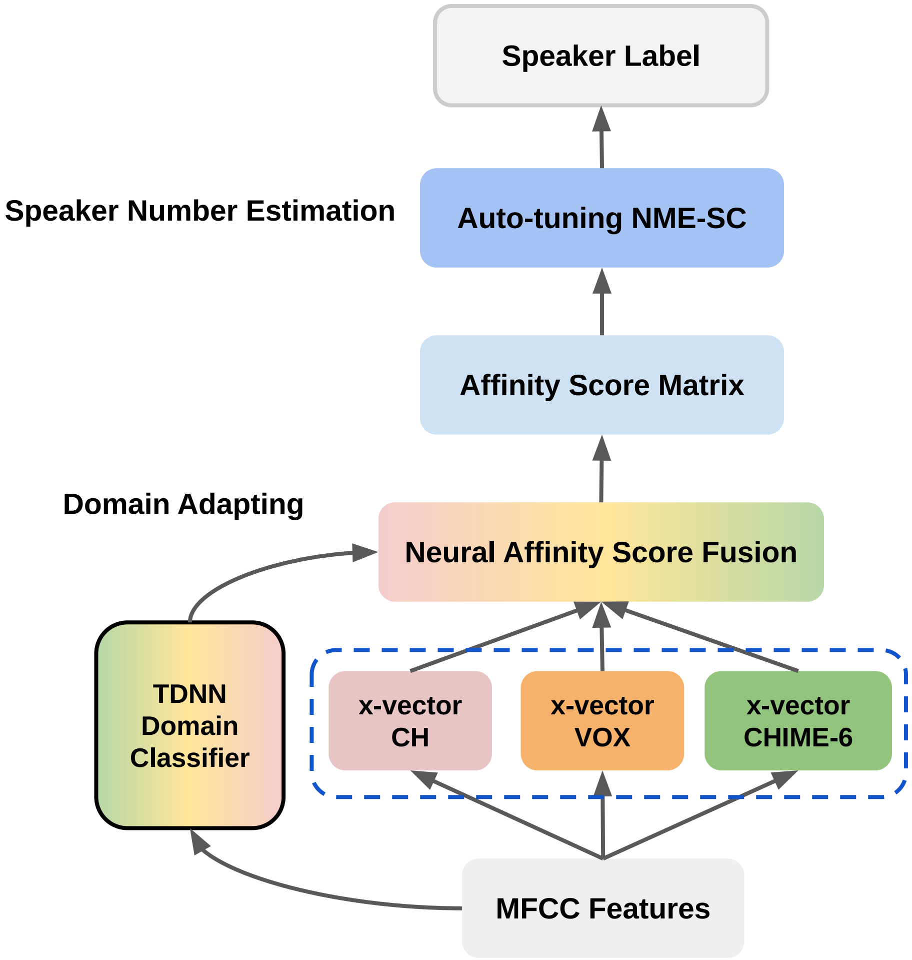
Medennikov, Ivan, et al. (Interspeech 2020)

NME-SC Challenge Winning Clustering Algorithm for Speaker Diarization

- Robust speaker clustering result provides a performance boost on Target-speaker VAD.
- In 2020 paper and CHIME-6 challenge, NME-SC showed constant improvement over AHC.

[1] https://chimechallenge.github.io/chime2020-workshop/papers/CHiME_2020_paper_medennikov.pdf
[2] Taejin Park et. al. “Auto-Tuning Spectral Clustering for Speaker Diarization Using Normalized Maximum Eigengap” IEEE SPL. 2019, p.381-385.

USC_SAIL 2020 DIHARD 3 System



Auto-tuning Spectral Clustering

- No manual parameter tuning on dev-set
- Clustering parameter varies over session

Domain Adaptive Affinity Fusion

- Soft decision on affinity weight selection
- Works better than hard decision method

Domain Adaptive Affinity Score Weighting

- **USC-SAIL Diarization System Performance**

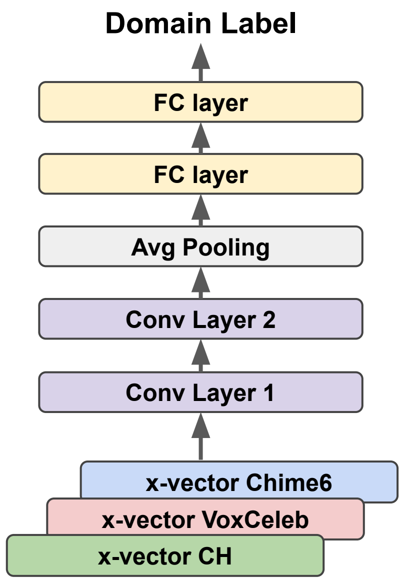
DER for each domain

	xvecCH	xvecVOX	xvecCHIME6
audiobooks	0.36	0.44	**0**
bc_interview	4.19	6.89	**3.09**
clinical	23.66	22.05	**16.63**
court	**4.63**	9.29	5.09
cts	**15.05**	19.57	19.13
maptask	6.6	**5.94**	6.64
meeting	31.44	31.79	**28.3**
restaurant	57.71	56.04	**52.59**
socio_field	16.06	15.33	**13.78**
socio_lab	**8.45**	10.36	9.61
webvideo	41.32	**39.24**	40.66

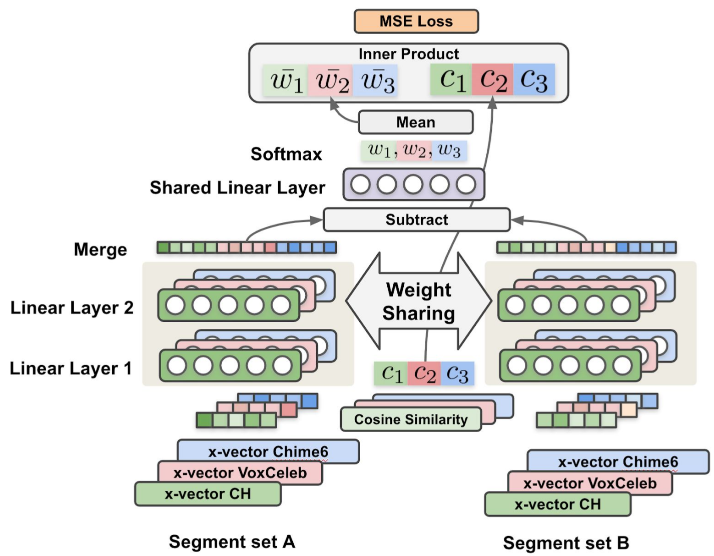
- **x-vector CallHome:** good on low-quality audio
  - Trained on SRE, SWBD (telephonic data)
  - DIHARD 1 Winner system
- **x-vector Voxceleb:** good on webvideo
  - Trained on interview videos on YouTube
  - VoxCeleb 1 and VoxCeleb 2
- **x-vector CHIME-6:** good on noisy environment
  - Trained on reverberated VoxCeleb data and CHIME-6 training data

Domain Adaptive Affinity Score Weighting

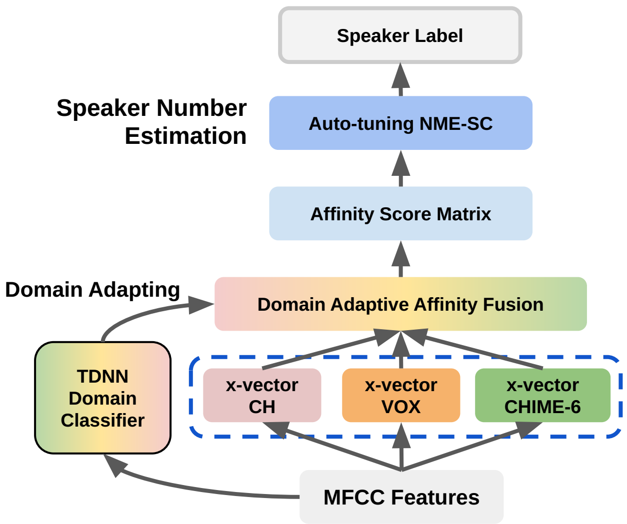
Neural Affinity Score Fusion: Domain Adaptive Speaker Diarization



Hard Decision
(Domain Estimation)



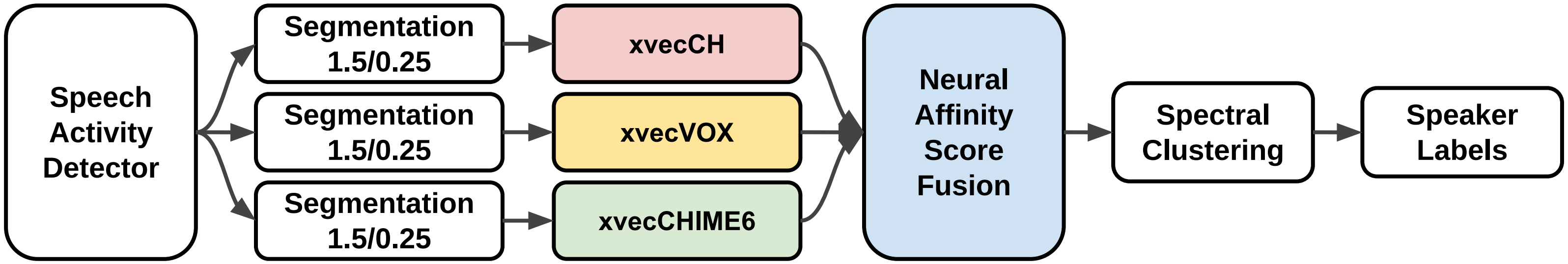
Soft Decision
(Domain Estimation)



USC SAIL
Domain Adaptive Speaker Diarization System

$$\mathbf{w} = \left(\frac{1}{N}\sum_{n=1}^{N} w_{1,n}, \frac{1}{N}\sum_{n=1}^{N} w_{2,n}, \frac{1}{N}\sum_{n=1}^{N} w_{3,n} \right)$$

Domain Adaptive Affinity Score Weighting



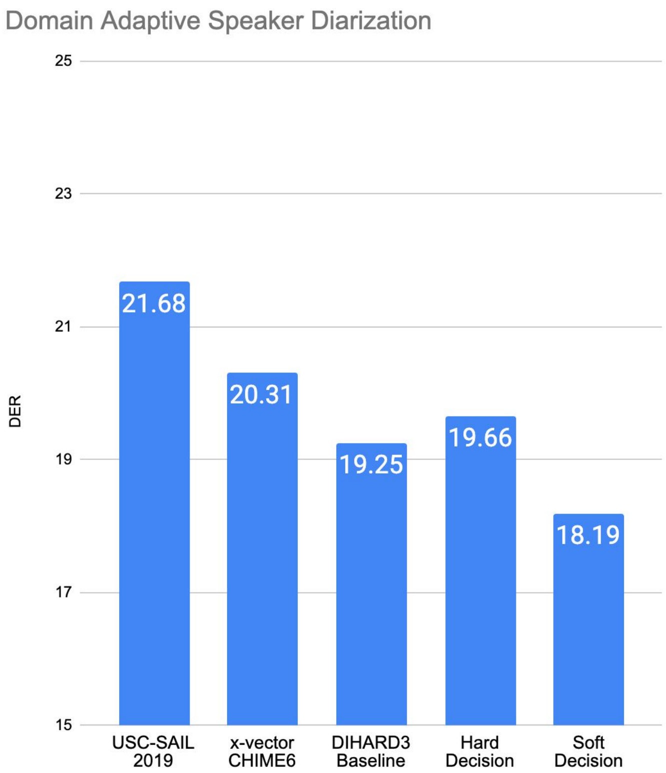
Proposed System

$$w_{CH} * a_{CH,1,1} + w_{VOX} * a_{VOX,1,1} + w_{CHIME6} * a_{CHIME6,1,1} = a_{fused}$$

Session level weighted sum of affinity matrix values

Domain Adaptive Affinity Score Weighting

Evaluation Results for DIHARD III Challenge: Track 1 Full DER (13th / 23 teams)



- **USC-SAIL 2019**: DIHARD 2 system by USC-SAIL
- **X-vector CHIME6**: The best performing embedding extractor
- **DIHARD3 Baseline**
- **Hard Decision**: The domain of each session is estimated by the domain estimator
- **Soft Decision**: The weights among embedding extractors are determined by neural affinity score fusion network.
- Soft Decision CORE set DER: **19.76%**


Discussions

Conclusion

- Auto-tuning clustering method showed improved performance over dev-set optimized binarized spectral clustering.
- Soft-decision method based on neural affinity fusion worked better than hard decision approach.
- The lack of overlap detection or source separation made the performance gap between the state-of-the-art system and our system.

Thank you!