Real-time Speaker Recognizer

YU Chuan

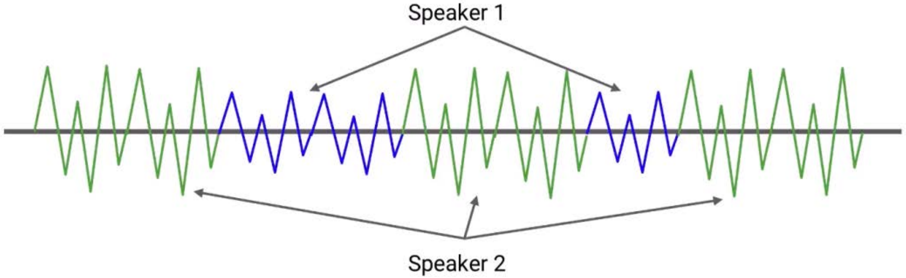
16. April. 2020

Outline

- Introduction to the speaker recognizer
- Overall workflow of our speaker recognizer
- How to evaluate our system
- Result

What is Speaker Recognizer?

- A system that recognizes/labels the speakers in a recorded audio file or live speech.
- Speaker diarization
- Who spoke when?

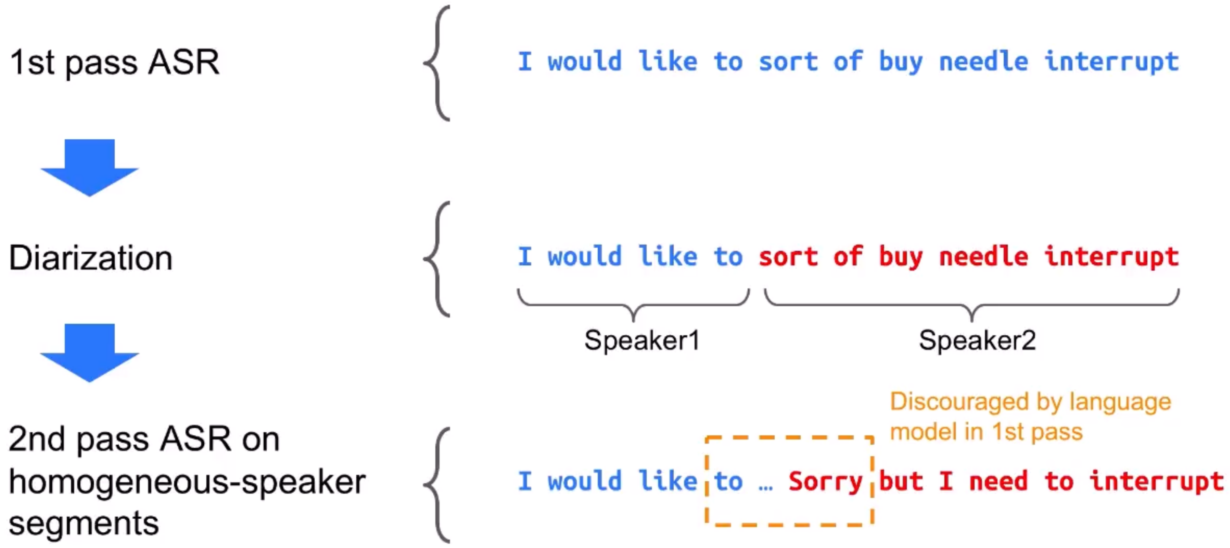


Why diarization?

- Lots of applications:
  - Medical records: doctor vs patient separation
  - Automatic notes-generation for meetings
  - Call center data analysis

Key application: improve ASR

- Speaker boundaries could help improve the accuracy of acoustic speech recognition (ASR)



1st pass ASR

I would like to sort of buy needle interrupt

Diarization

I would like to sort of buy needle interrupt

Speaker1

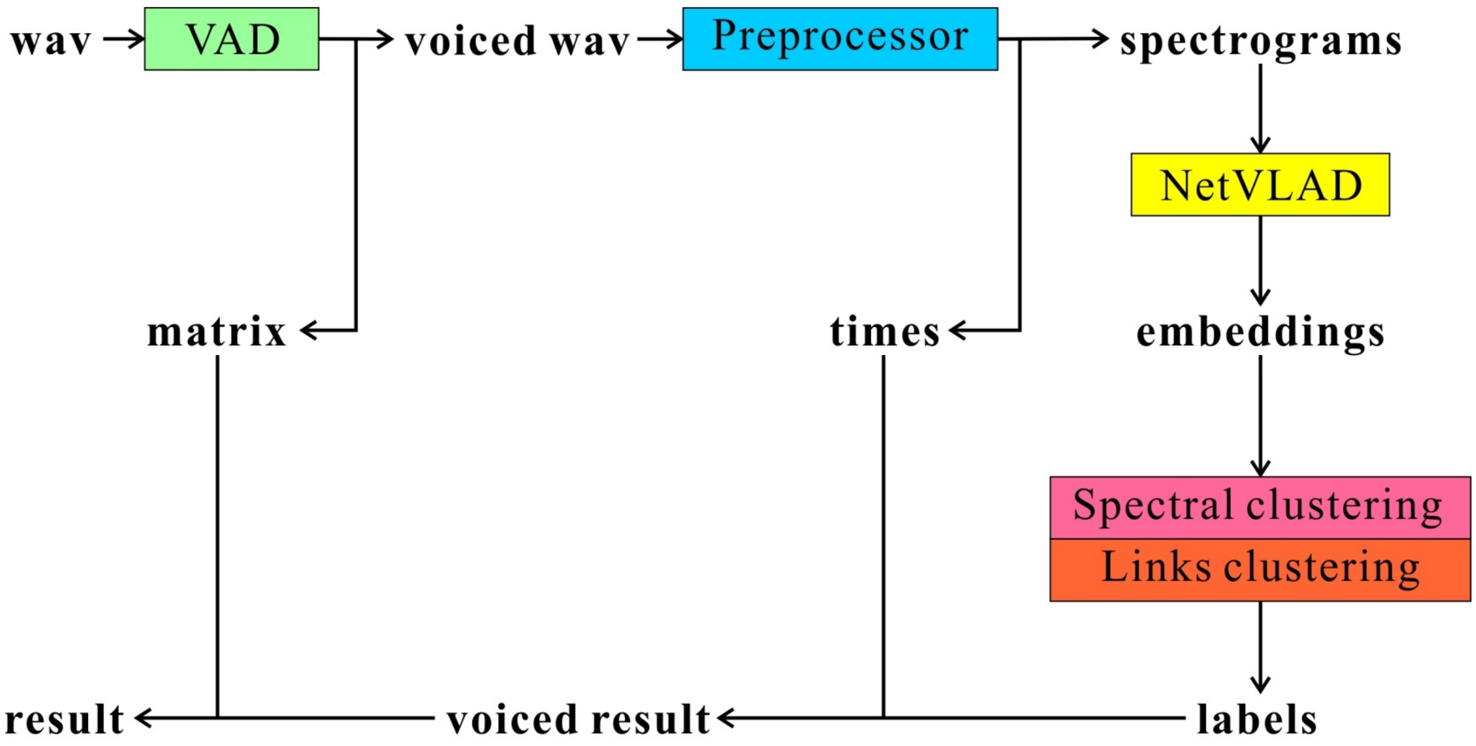
Speaker2

2nd pass ASR on homogeneous-speaker segments

Discouraged by language model in 1st pass

I would like to … Sorry but I need to interrupt

Overall workflow



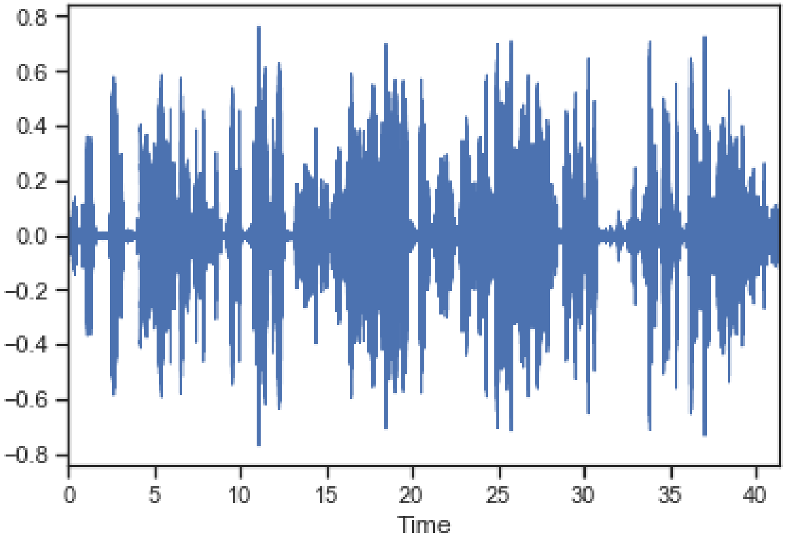
WebrtcVAD

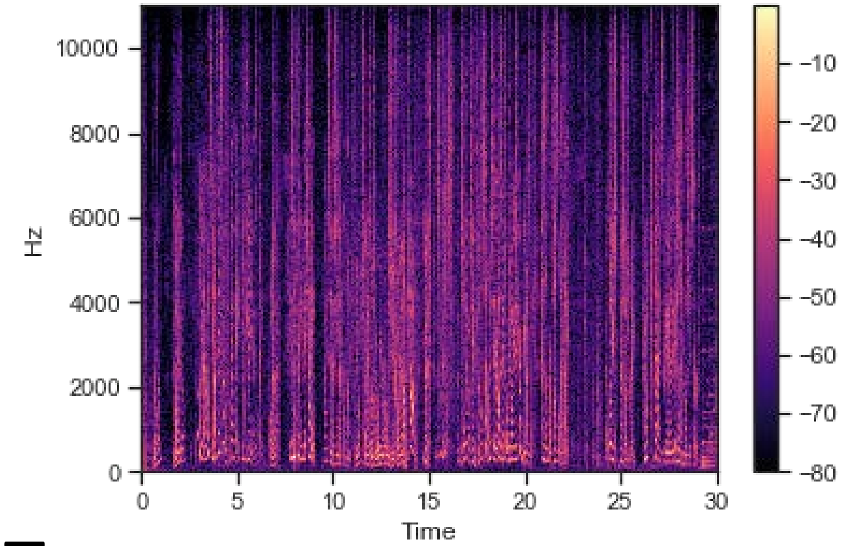
- Voice activity detector (VAD)
- A module used in audio signal processing in which absence or presence of human speech is detected.
- It is reported that the VAD developed by Google for the WebRTC project is one of the best VADs which is available, fast, and free.
- It will produce a **matrix** to record all the non-speech frames timestamp.

https://github.com/wiseman/py-webrtcvad

Preprocessor

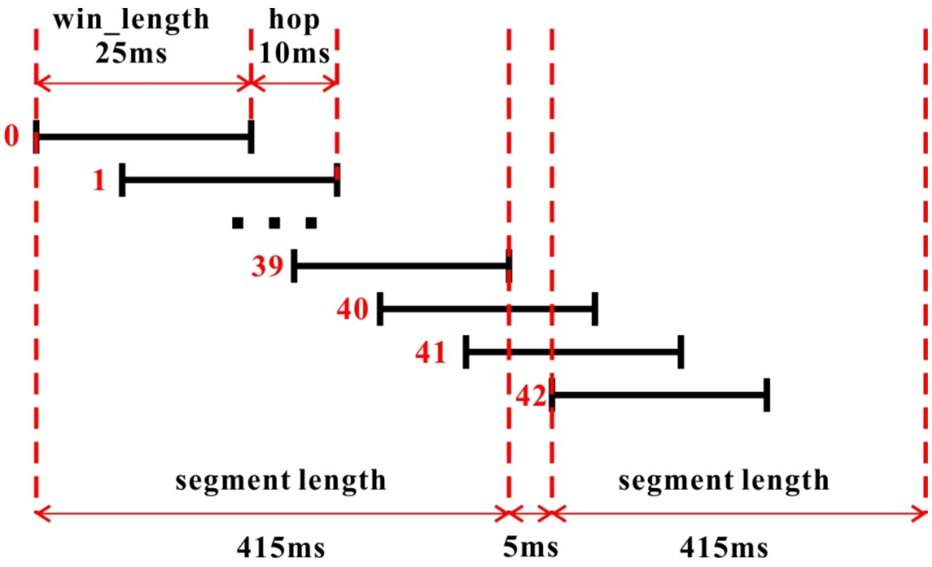
- Waveform→spectrogram





$$\text{waveform} \xrightarrow{\text{STFT}} \text{spectrogram}$$

Preprocessor



- Sampling rate: 16k Hz
- Frame length: 25ms
  $16000 \times 0.025 = 400$ points
- N of FFT is 512
- We obtain $\frac{N}{2} + 1 = 257$ values in each frame
- non-overlapping segment: Frame 0~39, 42~81 …
- Length of each segment: 415ms
- Spectrogram of each segment: 257×40

Audio Embedding Extraction

- Compact representation for each segment
  - Mel-frequency cepstral coefficients (MFCCs)
  - Speaker factors
  - d-vectors

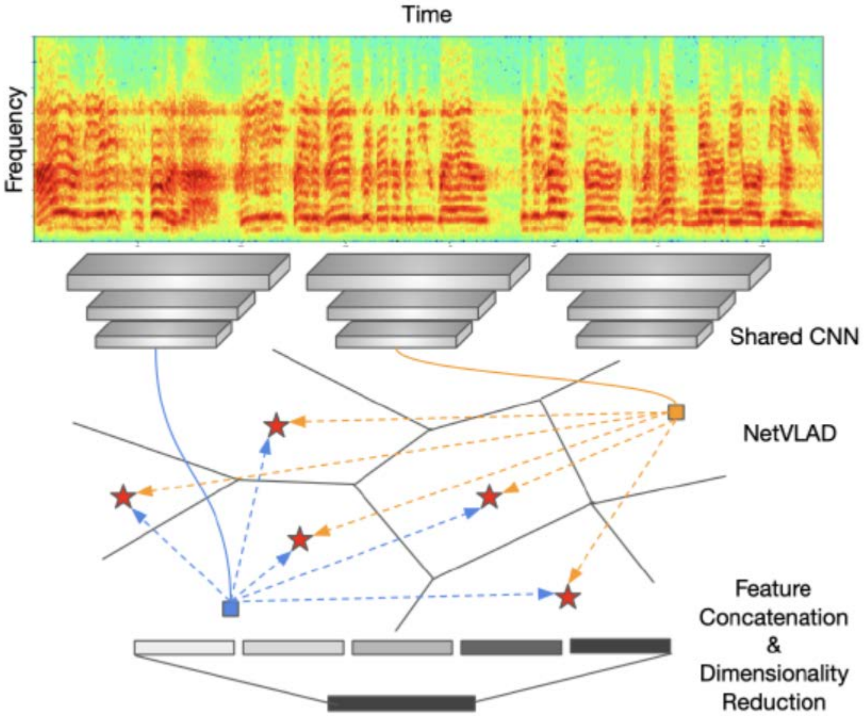
NetVLAD Embedding Extraction

- Net “Vector of Locally Aggregated Descriptors” embedding extraction
- State of the art performance by a significant margin on the VoxCeleb1 test set
- Fewer parameters than previous methods

https://github.com/WeidiXie/VGG-Speaker-Recognition

Weidi Xie et al., 2019

NetVLAD Embedding Extraction

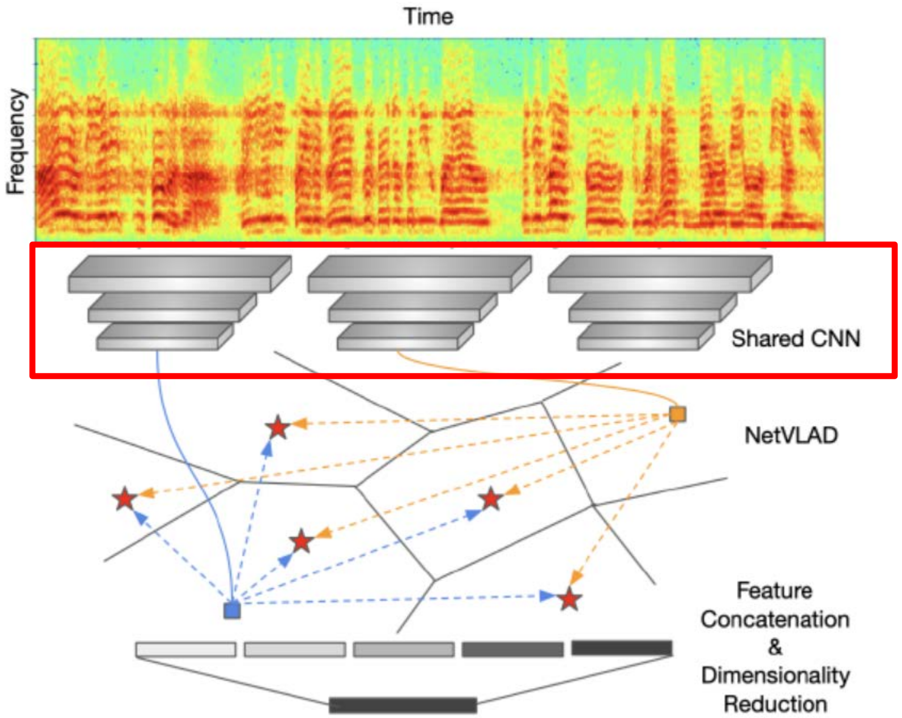


Feature extraction

NetVLAD

Weidi Xie et al., 2019

NetVLAD Embedding Extraction

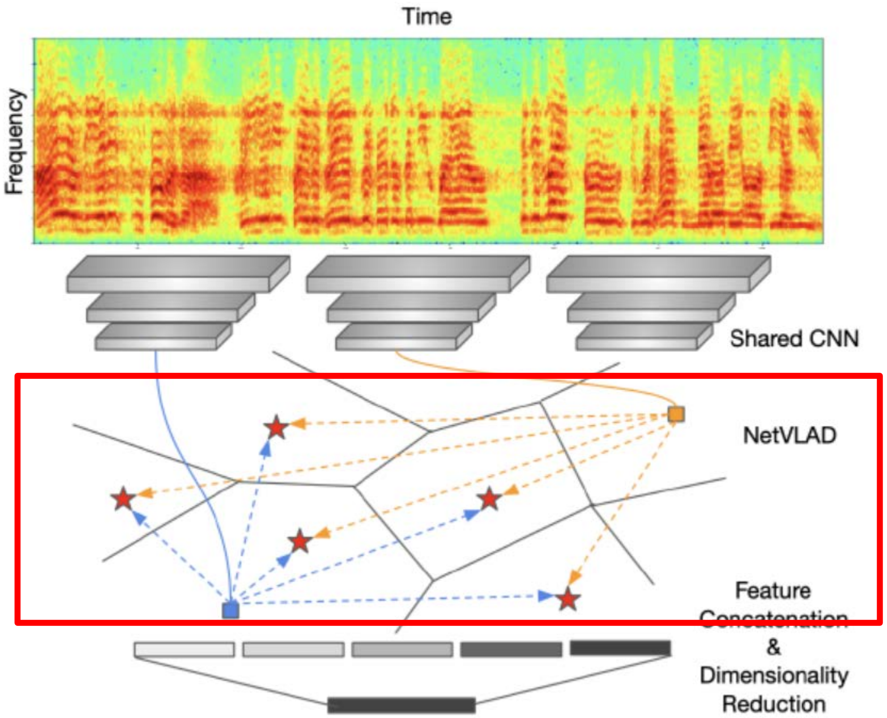


Feature extraction

NetVLAD

Weidi Xie et al., 2019

NetVLAD Embedding Extraction



Feature extraction

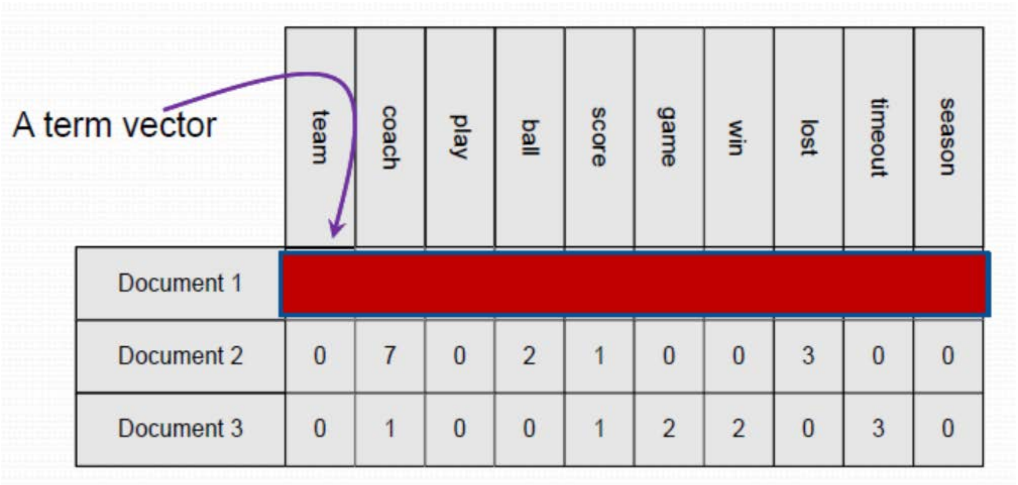
NetVLAD

Weidi Xie et al., 2019

Bag of Words

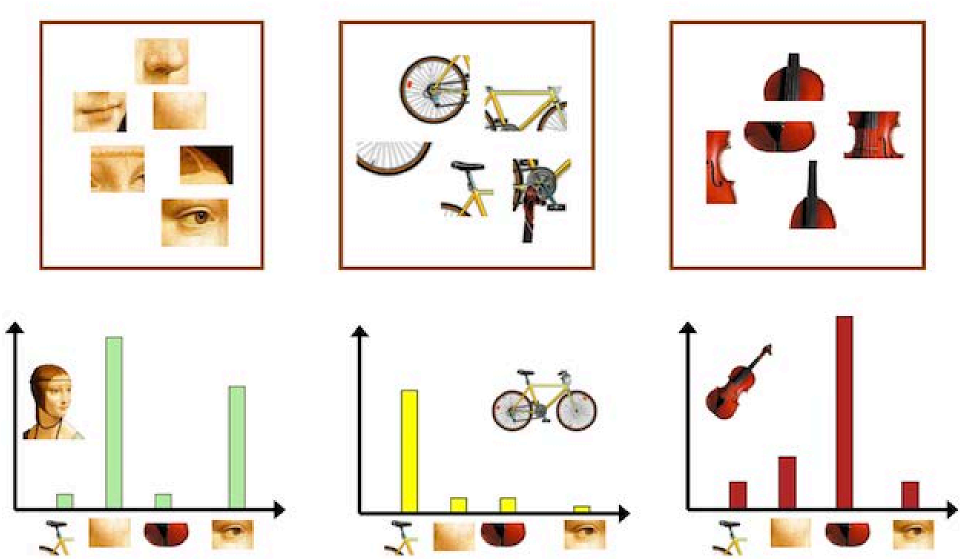
2007-01-23: State of the Union Address
George W. Bush (2001-)
baghdad challenges commitment economy elections freedom funding iraq qaeda terrorists war weapons
1962-10-22: Soviet Missiles in Cuba
John F. Kennedy (1961-63)
arms cuba economic freedom hemisphere missiles nuclear offensive soviet strength weapons
1941-12-08: Request for a Declaration of War
Franklin D. Roosevelt (1933-45)
facts freedom germany japanese navy pacific strength war

Visual words → CodeBook



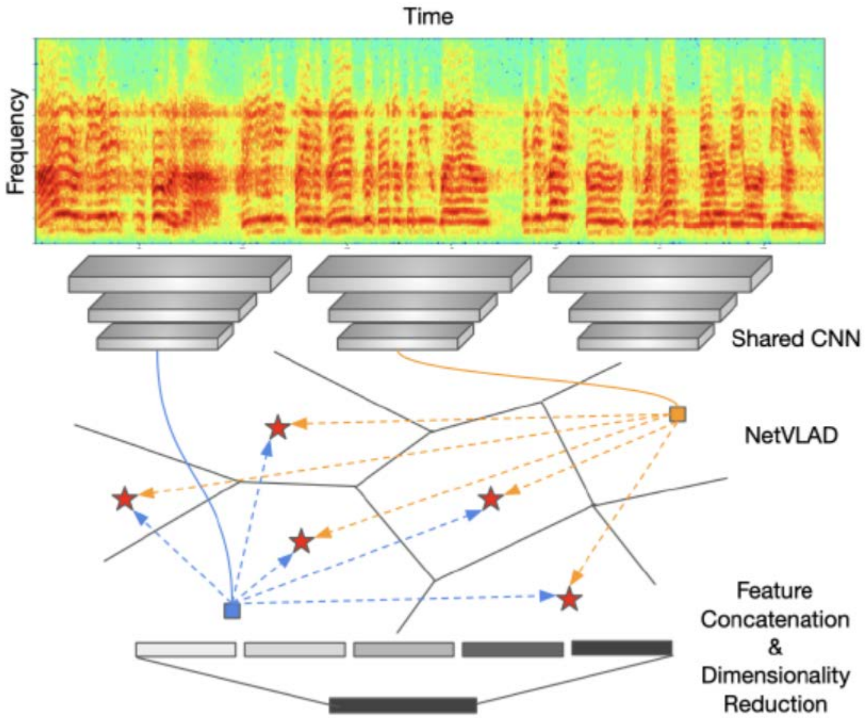
	team	coach	play	ball	score	game	win	lost	timeout	season
Document 1	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

NetVLAD: Bag of Feature



- SIFT feature
  (scale-invariant feature transform)
- K-means
- Codebook

NetVLAD Embedding Extraction



Feature extraction

NetVLAD:

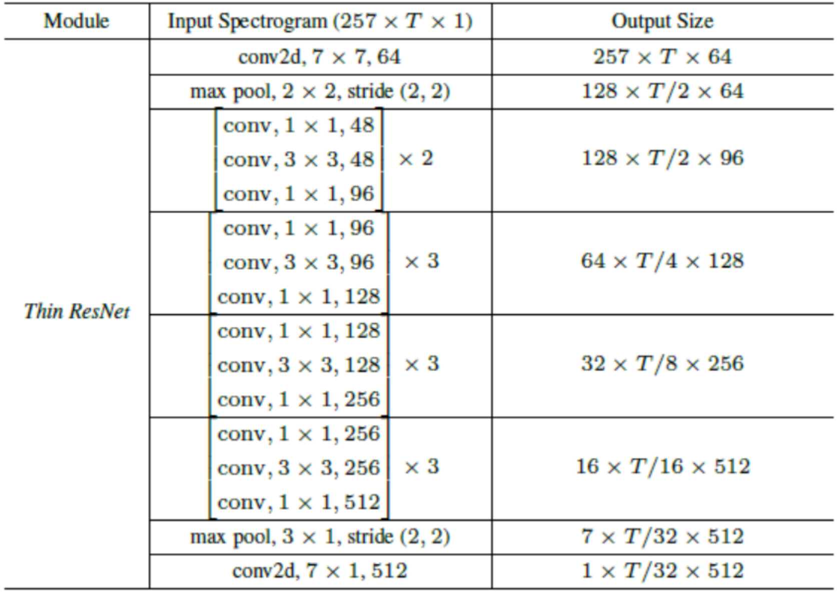
aggregate frame-level descriptors into a single utterance-level vector.

1×512

Weidi Xie et al., 2019

NetVLAD Embedding Extraction

Feature extraction



Module	Input Spectrogram ($257 \times T \times 1$)	Output Size
Thin ResNet	conv2d, 7×7, 64	$257 \times T \times 64$
	max pool, 2×2, stride (2, 2)	$128 \times T/2 \times 64$
	$\left[\begin{array}{l} \text{conv}, 1 \times 1, 48 \\ \text{conv}, 3 \times 3, 48 \\ \text{conv}, 1 \times 1, 96 \end{array}\right] \times 2$	$128 \times T/2 \times 96$
	$\left[\begin{array}{l} \text{conv}, 1 \times 1, 96 \\ \text{conv}, 3 \times 3, 96 \\ \text{conv}, 1 \times 1, 128 \end{array}\right] \times 3$	$64 \times T/4 \times 128$
	$\left[\begin{array}{l} \text{conv}, 1 \times 1, 128 \\ \text{conv}, 3 \times 3, 128 \\ \text{conv}, 1 \times 1, 256 \end{array}\right] \times 3$	$32 \times T/8 \times 256$
	$\left[\begin{array}{l} \text{conv}, 1 \times 1, 256 \\ \text{conv}, 3 \times 3, 256 \\ \text{conv}, 1 \times 1, 512 \end{array}\right] \times 3$	$16 \times T/16 \times 512$
	max pool, 3×1, stride (2, 2)	$7 \times T/32 \times 512$
	conv2d, 7×1, 512	$1 \times T/32 \times 512$

NetVLAD

$$R^{1 \times T/32 \times 512} \rightarrow K \times D \text{ matrix } V$$

K refers to the number of chosen cluster
D refers to the dimensionality of each cluster

$$V(k, j) = \sum_{t=1}^{T/32} \frac{e^{w_k x_t + b_k}}{\sum_{k'=1}^{K} e^{w'_k x_t + b_{k'}}} (x_t(j) - c_k(j))$$

wk and bk are trainable parameters

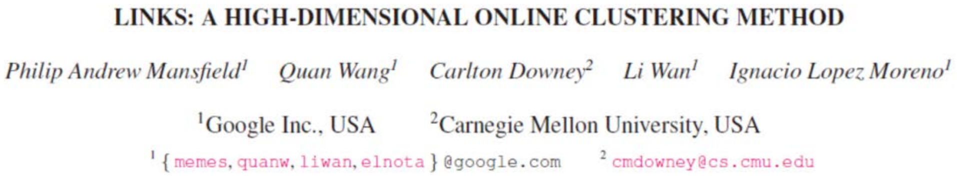
$$k \in [1, 2, ..., K]$$

Weidi Xie et al., 2019

Clustering

- Online clustering
  - Naïve online
  - Links online
- Offline clustering
  - K-means
  - Spectral clustering

Links Online clustering

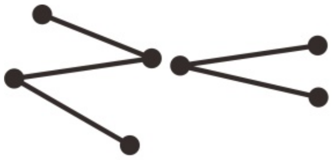


LINKS: A HIGH-DIMENSIONAL ONLINE CLUSTERING METHOD

Philip Andrew Mansfield[1] *Quan Wang*[1] *Carlton Downey*[2] *Li Wan*[1] *Ignacio Lopez Moreno*[1]

[1]Google Inc., USA [2]Carnegie Mellon University, USA

[1] {memes, quanw, liwan, elnota}@google.com [2] cmdowney@cs.cmu.edu



Node:subcluster (containing vectors)
Connected nodes: cluster

- N-dimensional vectors ($N \geq 128$)
- Two-level hierarchy

- Add new vector
- Merge subcluster
- Check edges

Philip Andrew Mansfield et al., 2018

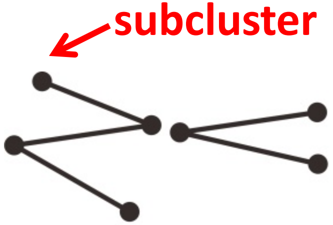
Links Online clustering

LINKS: A HIGH-DIMENSIONAL ONLINE CLUSTERING METHOD

Philip Andrew Mansfield[1] *Quan Wang*[1] *Carlton Downey*[2] *Li Wan*[1] *Ignacio Lopez Moreno*[1]

[1]Google Inc., USA [2]Carnegie Mellon University, USA

[1] {memes, quanw, liwan, elnota}@google.com [2] cmdowney@cs.cmu.edu



subcluster

Node:subcluster (containing vectors)
Connected nodes: cluster

- N-dimensional vectors ($N \geq 128$)
- Two-level hierarchy

- Add new vector
- Merge subcluster
- Check edges

Philip Andrew Mansfield et al., 2018

Links Online clustering

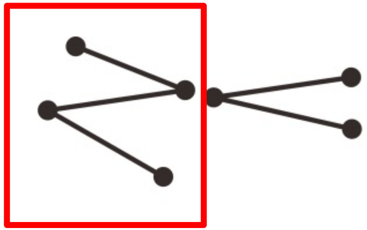
LINKS: A HIGH-DIMENSIONAL ONLINE CLUSTERING METHOD

Philip Andrew Mansfield[1] *Quan Wang*[1] *Carlton Downey*[2] *Li Wan*[1] *Ignacio Lopez Moreno*[1]

[1]Google Inc., USA [2]Carnegie Mellon University, USA

[1]{memes, quanw, liwan, elnota}@google.com [2]cmdowney@cs.cmu.edu

cluster



Node:subcluster (containing vectors)
Connected nodes: cluster

- N-dimensional vectors ($N \geq 128$)
- Two-level hierarchy

- Add new vector
- Merge subcluster
- Check edges

Philip Andrew Mansfield et al., 2018

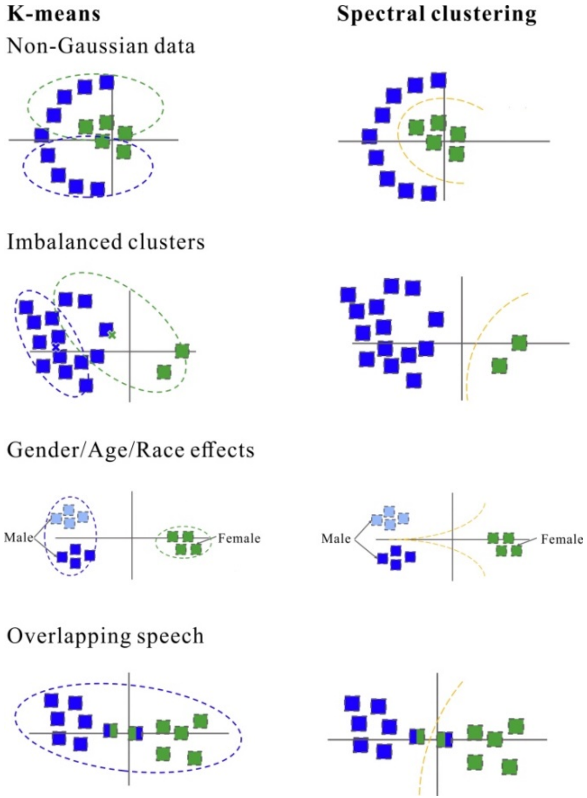
Links Online clustering

$\mathrm{T_s}$: *the subcluster similarity threshold*
$\mathrm{T_p}$: *the pair similarity maximum*
$\mathrm{T_c}$: *the cluster similarity threshold*

- Manually **label a dataset** with cluster IDs
- **Run Links clustering algorithm** on the data
- **Adjust hyperparameters** to improve the accuracy of the output cluster IDs

Philip Andrew Mansfield et al., 2018

Offline clustering



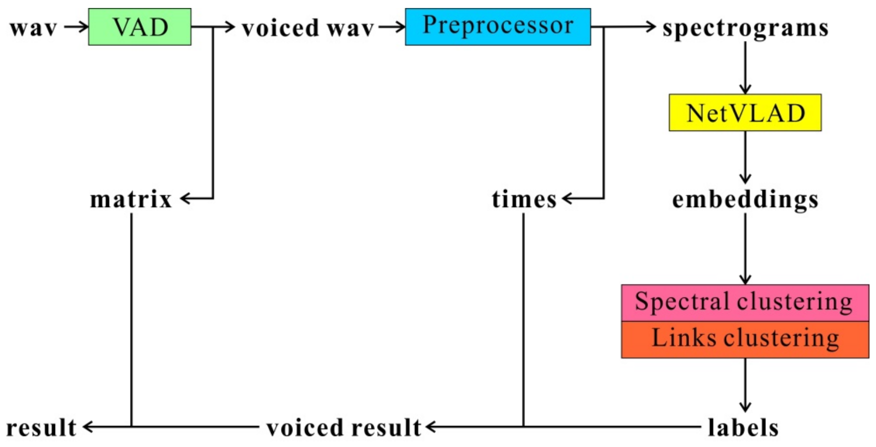
- Speech data are often non-Gaussian.
- One person may speak more often.
- Inter-gender differences large, intra-gender differences small.
- Overlapping speech creates connects between clusters.

K-means may not be good at clustering the speech data.
These problems can be mitigated by **spectral clustering**.

https://github.com/wq2012/SpectralCluster

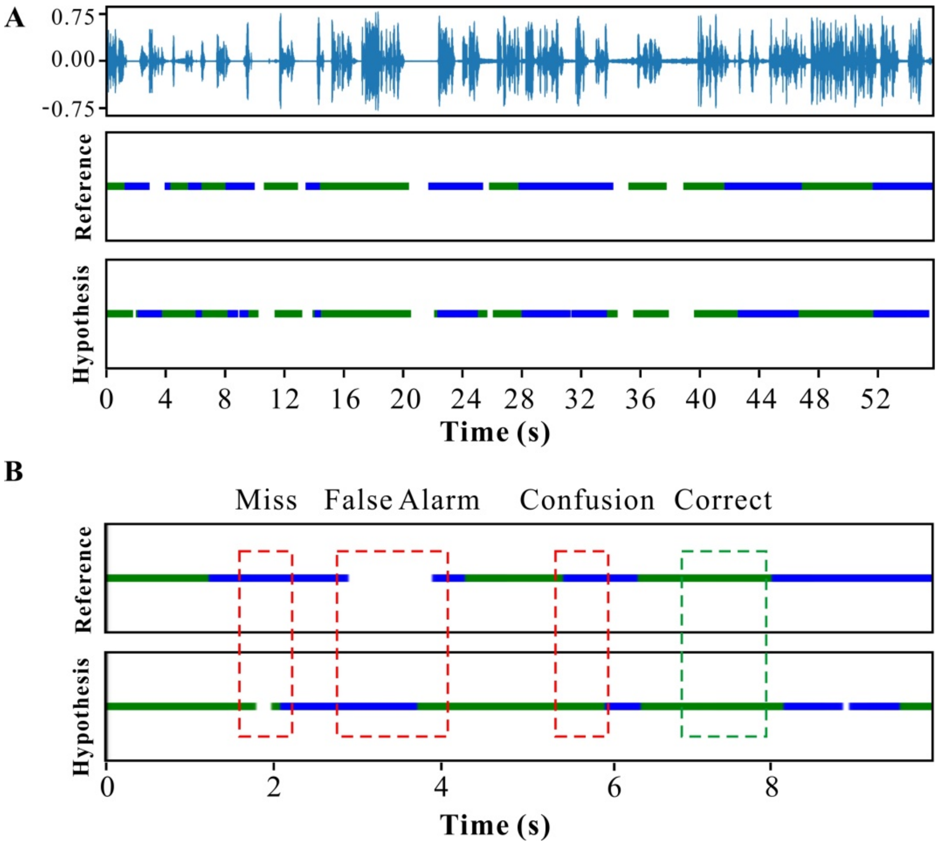
From Google

Overall workflow summary



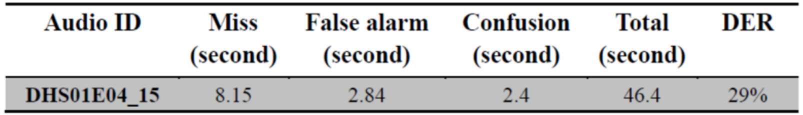
- **VAD**: WebrtcVAD
- **Preprocessor**: Waveform→spectrogram Speech segmentation: 415ms
- **Audio embedding extraction**: NetVLAD
- **Clustering**: Spectral clustering and Links clustering
- **Integration**

Evaluation



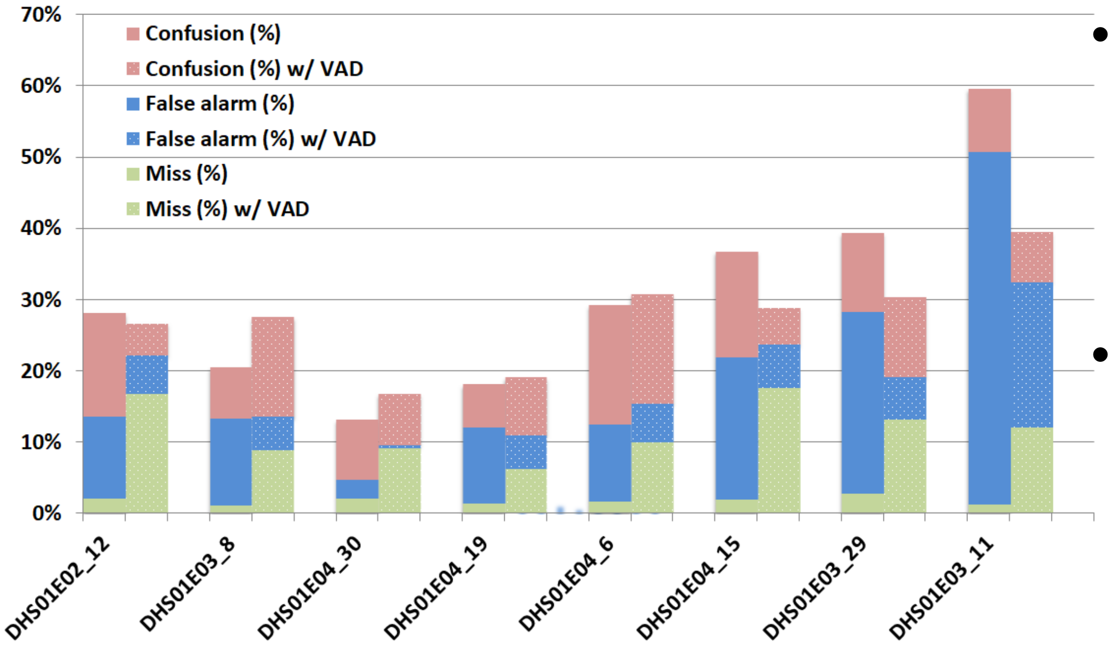
- **Diarization Error rate (DER)**
- **pyannote.metrics** python module
- **Missed detection, False Alarm:** VAD, segmentation
- **Confusion:** Some literatures only report this

$$\mathbf{DER} = \frac{\text{missed detection+false alarm+confusion}}{\text{total}}$$



Audio ID	Miss (second)	False alarm (second)	Confusion (second)	Total (second)	DER
DHS01E04_15	8.15	2.84	2.4	46.4	29%

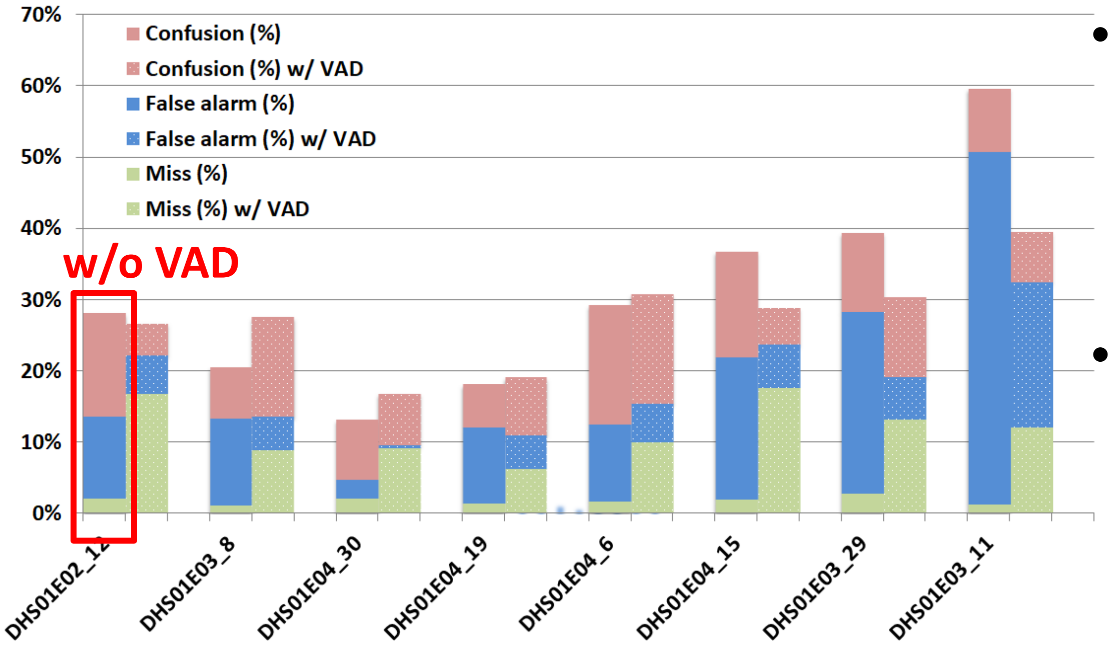
VAD impact on DER



- If DER without VAD is **less than 30%** (first five clips), VAD has limited impact on DER

- If DER without VAD is **more than 30%** (last three clips), the VAD can significantly decrease the DER.

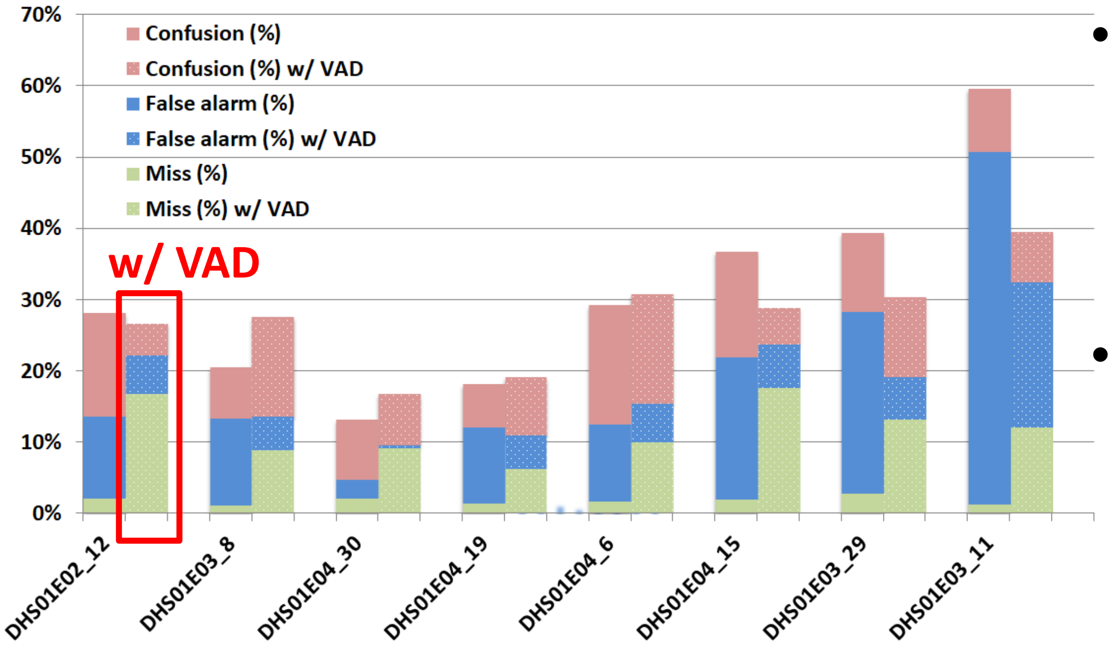
VAD impact on DER



- If DER without VAD is **less than 30%** (first five clips), VAD has limited impact on DER

- If DER without VAD is **more than 30%** (last three clips), the VAD can significantly decrease the DER.

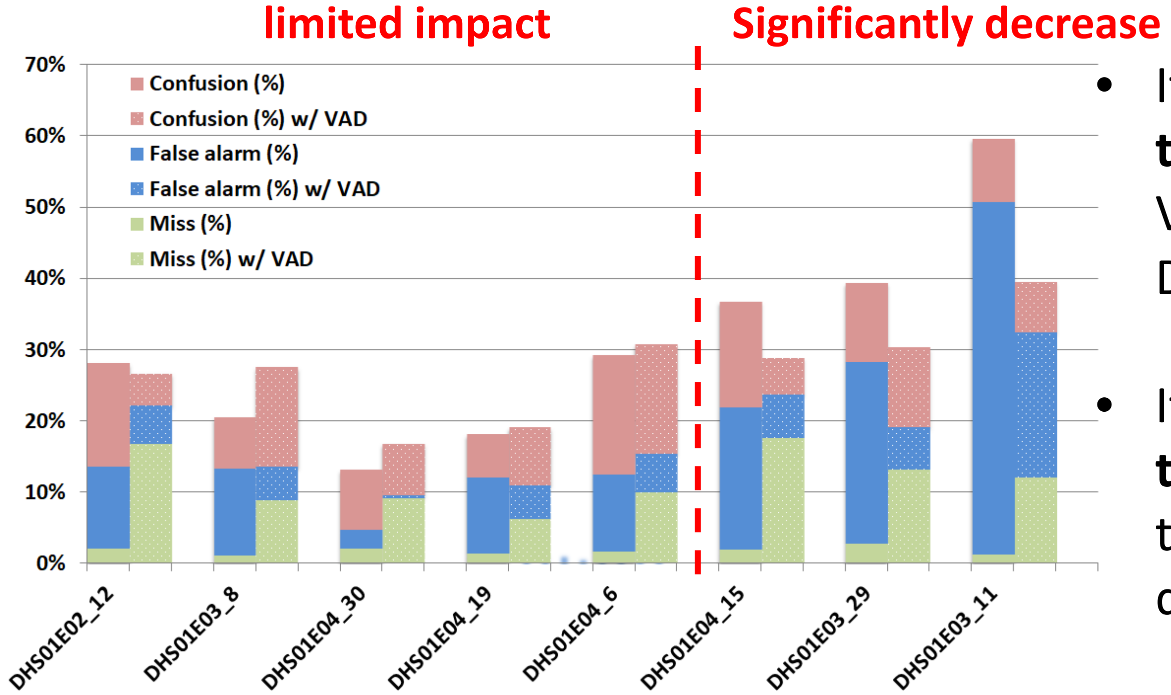
VAD impact on DER



- If DER without VAD is **less than 30%** (first five clips), VAD has limited impact on DER

- If DER without VAD is **more than 30%** (last three clips), the VAD can significantly decrease the DER.

VAD impact on DER

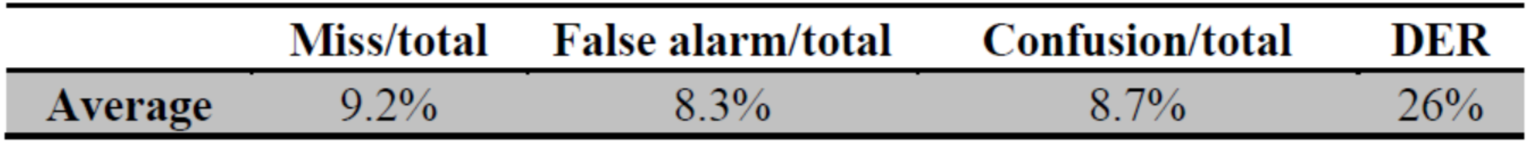


- If DER without VAD is **less than 30%** (first five clips), VAD has limited impact on DER

- If DER without VAD is **more than 30%** (last three clips), the VAD can significantly decrease the DER.

Result: Offline Clustering

- We manually labeled 16 video clips (40 sec – 1 min) as reference

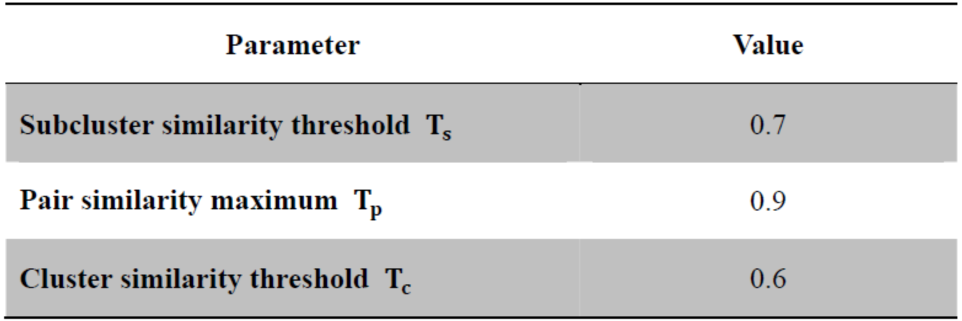


	Miss/total	False alarm/total	Confusion/total	DER
Average	9.2%	8.3%	8.7%	26%

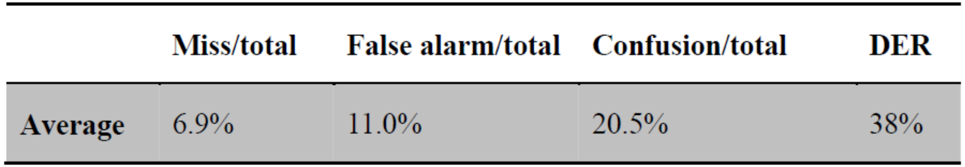
- Missed detection and False alarm parts contribute to 67% of the final DER.
- Our offline speaker recognizer is satisfactory.
- We will improve the Missed detection and False Alarm parts.

Result: Online Clustering

- Training data: 70% of labeled video clips (11 clips) to find out the optimal thresholds T_s, T_p, T_c
- Test data: the rest 30% (5 clips)



Parameter	Value
Subcluster similarity threshold T_s	0.7
Pair similarity maximum T_p	0.9
Cluster similarity threshold T_c	0.6



	Miss/total	False alarm/total	Confusion/total	DER
Average	6.9%	11.0%	20.5%	38%

- Minimal DER of Training set is 41.4%
- The DER of Test set is 38%

Conclusions

- Combine **WebrtcVAD**, **NetVLAD** Audio embedding extraction technique, **Spectral** clustering and **Links** clustering algorithm to build our offline and online speaker recognizer
- Offline speaker recognizer: DER is 26%; Percentage of Confusion is only 8.7%.
- Online speaker recognizer: DER is 38%; Percentage of Confusion is 20.5%. (only derived from 5 samples)
- The offline speaker recognizer **outperforms** the online speaker recognizer.
- Online speaker recognizer can be **real-time** which is a big advantage.

Future Works

- Label more video clips (30-50) as reference
- Find optimal thresholds $\mathrm{T_s}, \mathrm{T_p}, \mathrm{T_c}$ of Links algorithm
- Training NetVLAD embedding algorithm by ourselves

Acknowledgement

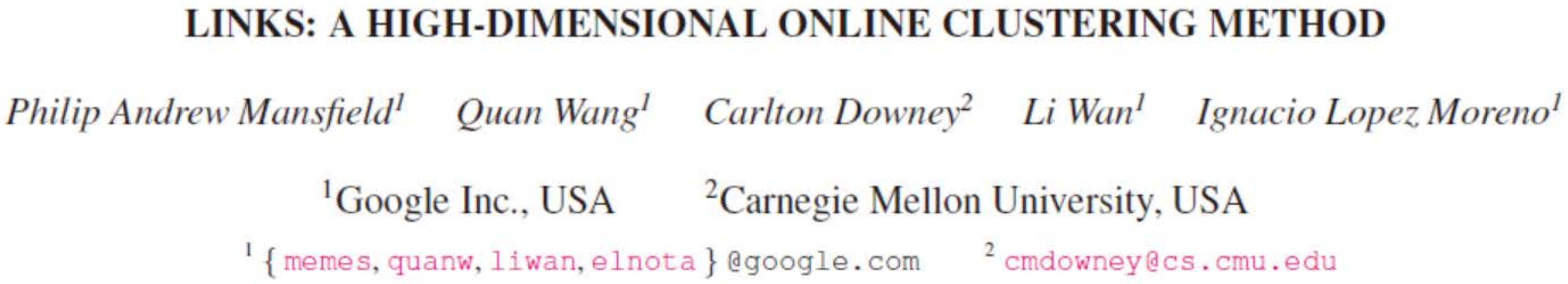
Supervisor: Dr. Beta C.L. Yip

Second Examiner: Dr. H.F. Ting

All my friends in HKU

Thanks

Online clustering



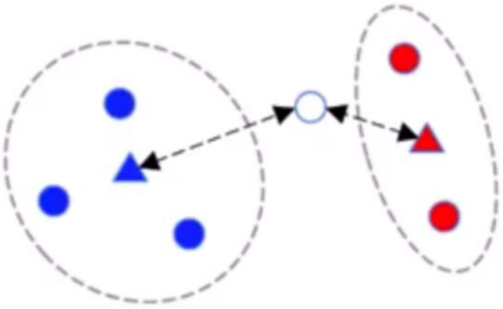
LINKS: A HIGH-DIMENSIONAL ONLINE CLUSTERING METHOD

Philip Andrew Mansfield[1] *Quan Wang*[1] *Carlton Downey*[2] *Li Wan*[1] *Ignacio Lopez Moreno*[1]

[1]Google Inc., USA [2]Carnegie Mellon University, USA

[1] {memes, quanw, liwan, elnota} @google.com [2] cmdowney@cs.cmu.edu

$\mathrm{T_s}$: *the subcluster similarity threshold*
$\mathrm{T_p}$: *the pair similarity maximum*
T_c: *the cluster similarity threshold*

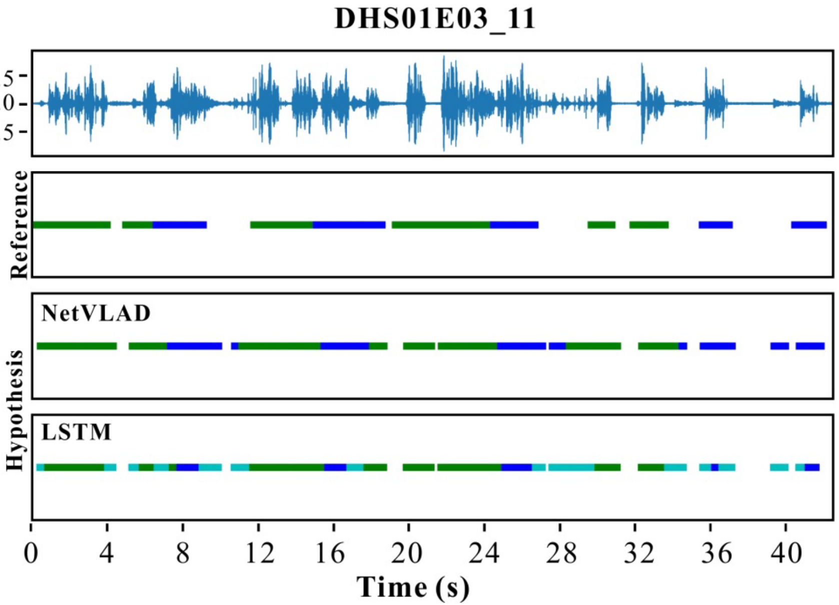


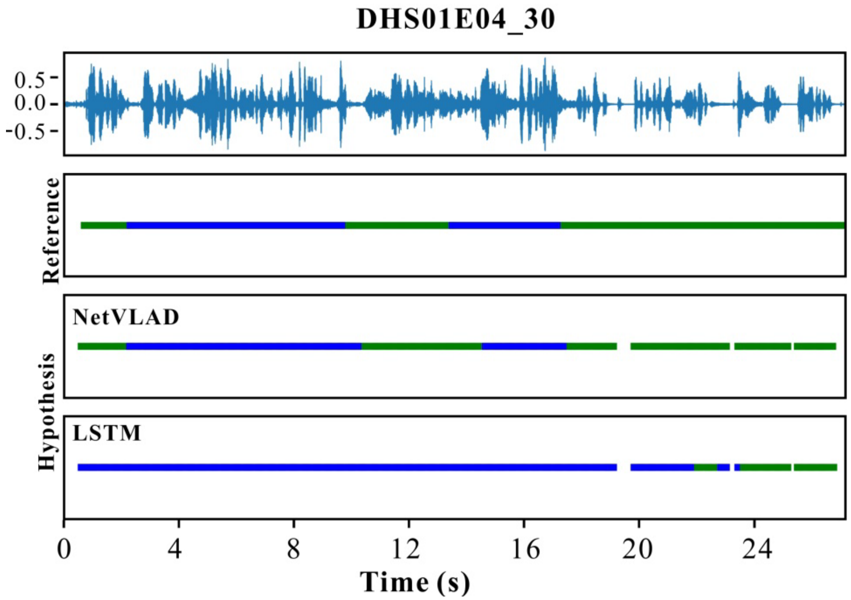
Philip Andrew Mansfield et al., 2018

Embedding extraction algorithm selection

- i-vectors
- Long Short Term Memory (LSTM)
- NetVLAD

Embedding extraction algorithm selection

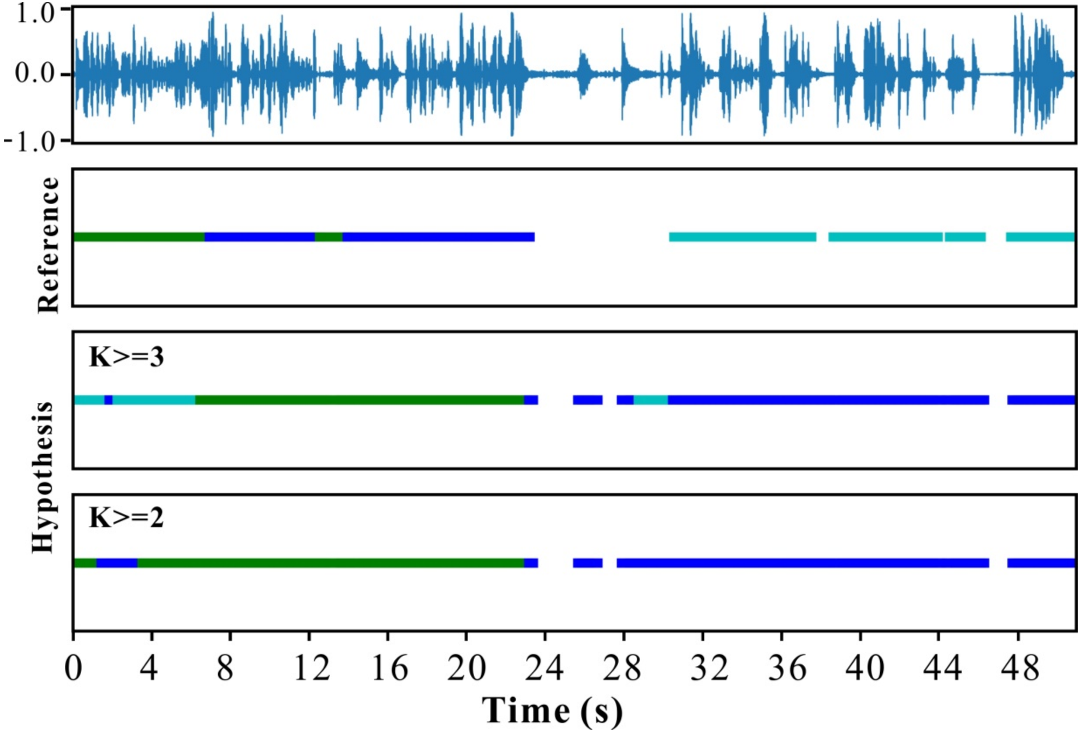




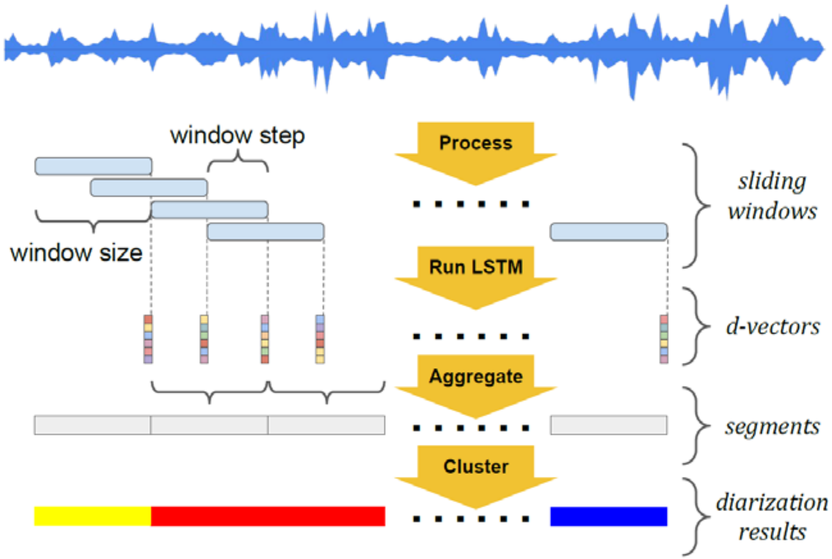
Embedding extraction algorithm selection

- The audio file length in the training set is 1.6s.
- The segment length is 415ms.
- LSTM based one may be sensitive to the audio file length which means the audio file length in the training set must be comparable with the one in the test set.
- NetVLAD may tolerate this difference which can perform better for speaker recognizer.

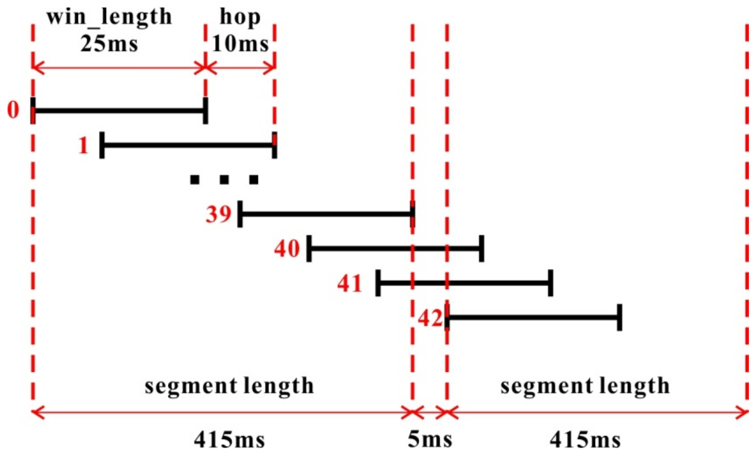
Multi-persons speech



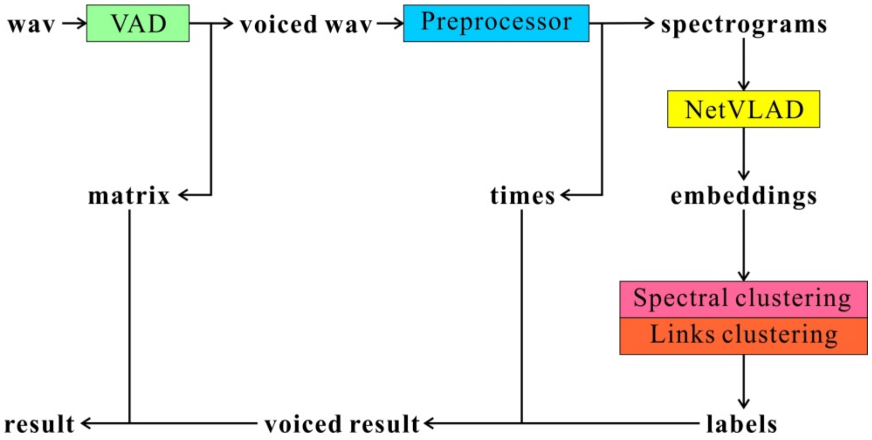
Decrease Missed Detection and False Alarm



“SPEAKER DIARIZATION WITH LSTM”



Overall workflow summary



- **VAD**: WebrtcVAD
- **Preprocessor**: Waveform→spectrogram Speech segmentation: 415ms Spectrogram of each segment: 257×40
- **Audio embedding extraction**: NetVLAD, Embedding of each segment: 1×512
- **Clustering**: Spectral clustering and Links clustering
- **Integration**