Současné trendy v ověřování a extrakci reprezentací mluvčího z řeči

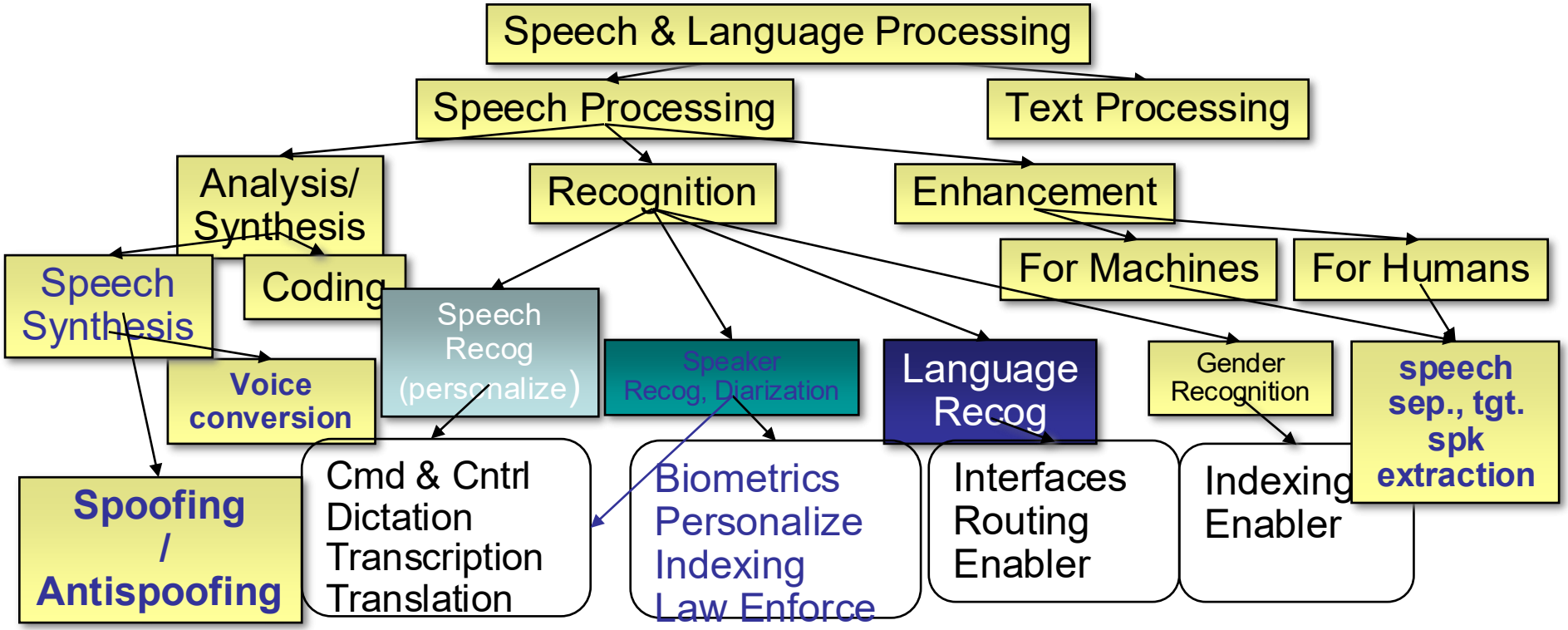
Oldřich Plchot
Speech@FIT, Brno University of Technology, Czech Republic
iplchot@fit.vut.cz

CZAI Summer school
Ostrava
9th September, 2026

Agenda

- Speaker recognition applications
  - Why we need speaker recognition
  - Different application domains
- Background and theory
  - Feature extraction
  - Modelling of embeddings, PLDA
  - Current trends in speaker verification
- Evaluation metrics and Wespeaker toolkit
- Conclusions

Speech processing, use of speaker representations



Speaker Recognition Areas of Use

Security and defense
Forensic
Looking for suspect in quantity of audio
Waiting online for suspect

Access Control
Physical facilities
Computer networks & websites

Transaction Authentication
Telephone banking
Remote purchases

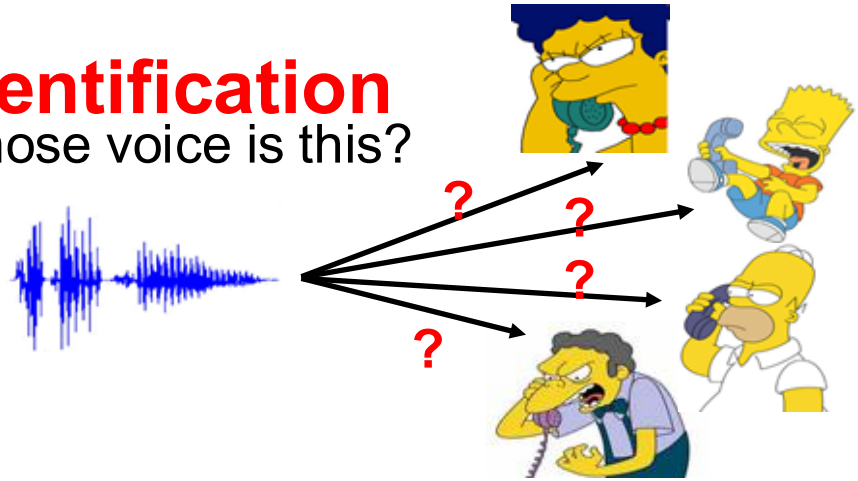
Speech Data Management
Voice mail browsing
Search in audio archives

Personalization
Voice-web/device customization
Intelligent answering machine

Speaker Recognition Tasks

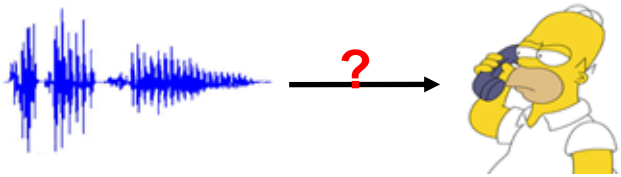
Identification

Whose voice is this?



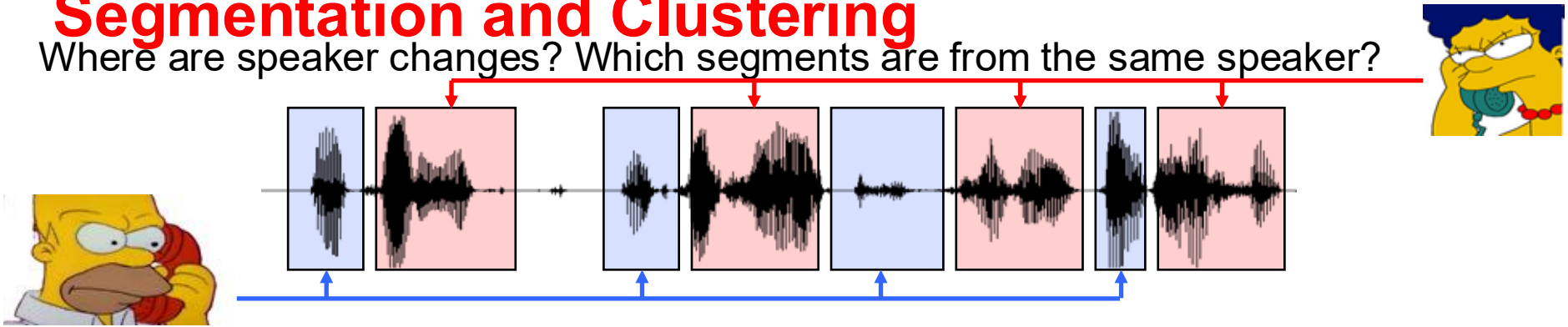
Verification

Is this Homer's voice?



Segmentation and Clustering

Where are speaker changes? Which segments are from the same speaker?



Biometrics

- **Biometric**: a human generated signal or attribute for authenticating a person's identity
- Voice biometric can be combined with other forms to reach highest security
  - Voice +
  - Finger print
  - Face, Iris
- **Voice is popular biometric**
  - Natural signal to produce
  - Does not require a specialized input device
  - Can be obtained remotely, without speaker assistance

Features for Speaker Recognition

- Desirable attributes of features for an automatic system (Wolf 72)

Practical
- **Occur naturally and frequently in speech**
- **Easily measurable**

Robust
- **Not change over time or be affected by speaker health**
- **Not be affected by reasonable background noise nor depend on specific transmission characteristics**

Secure
- **Not be subject to mimicry**

- No features have all these attributes
- Features derived from spectrum of speech have proven to be the most effective in automatic systems
  - Mel-filter bank outputs, MFCCs and other variants, raw signal processed by trained CNN (transformers)

Pure biometrics vs combination with “password”

Different applications are suited for different approaches

Text-dependent

- Recognition system knows text spoken by person
- Example: fixed or prompted phrases
- Used for applications with strong control over user
- Knowledge of spoken text can improve system performance

Text-independent

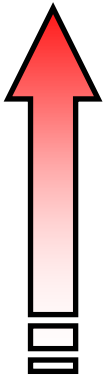
- Recognition system does not know text spoken by person
- Example: User selected phrases, conversational speech
- Used for applications with less or no control over user
- More flexible system but more difficult problem

Features for Speaker Recognition

- Humans use several levels of perceptual cues for speaker recognition

Hierarchy of Perceptual Cues

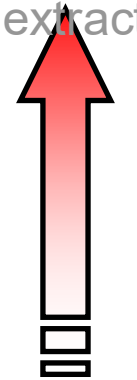
High-level cues (learned traits)



Low-level cues (physical traits)

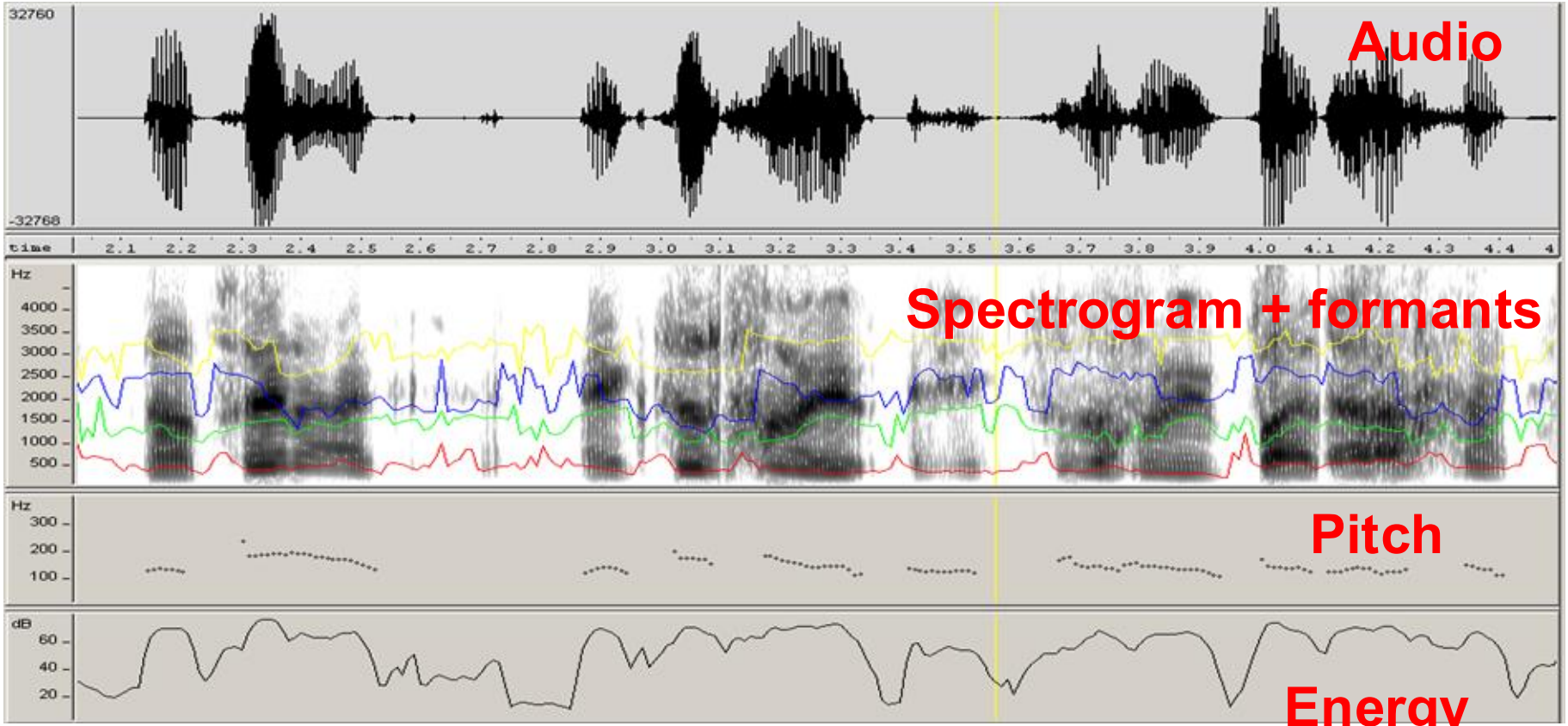
Semantics, idiolect, pronunciations, idiosyncrasies	Socio-economic status, education, place of birth
Prosodics, rhythm, speed intonation, volume modulation	Personality type, parental influence
Acoustic aspects of speech, nasal, deep, breathy, rough	Anatomical structure of vocal apparatus

Difficult to automatically extract

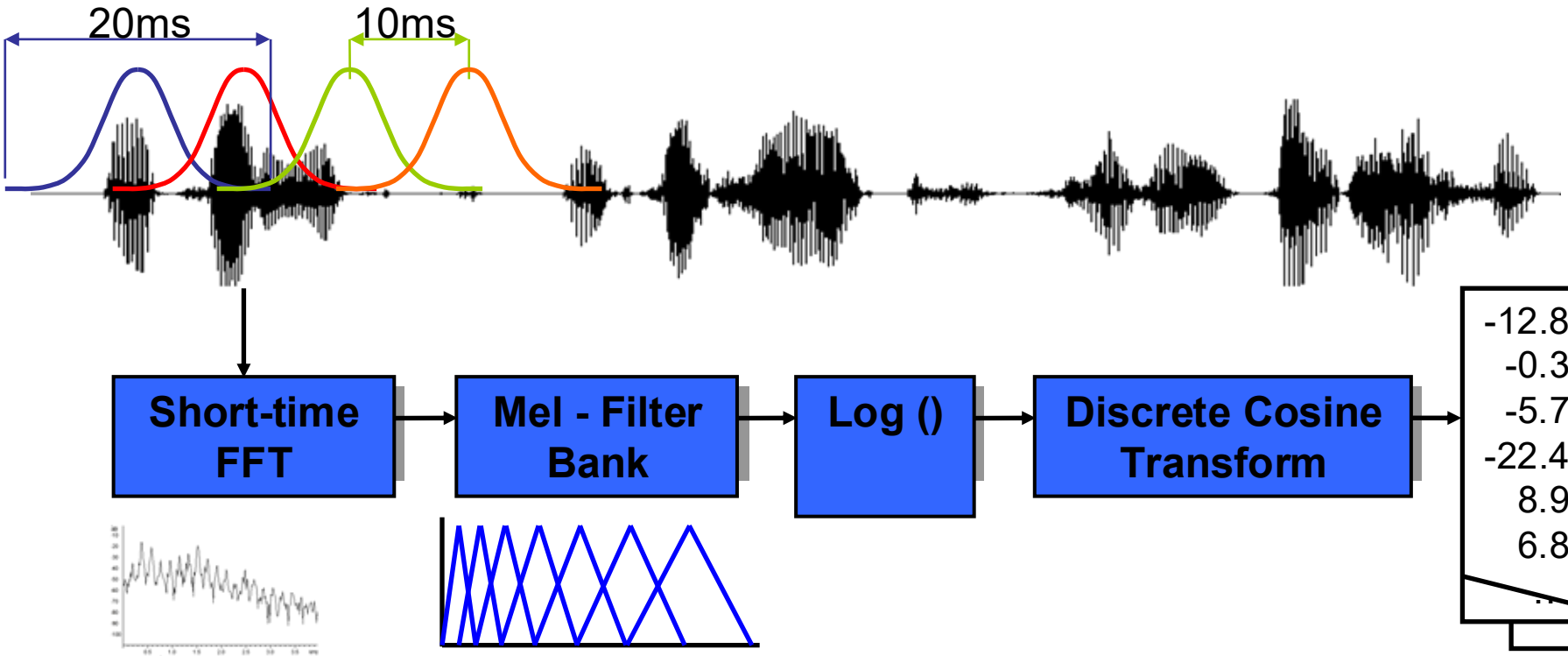


Easy to automatically extract

Audio Analysis



Spectral features – MFCC

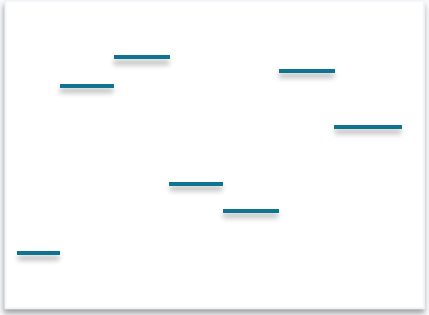


PROSODIC FEATURES IN SPEAKER VERIFICATION

- Leveraging higher-level behavioral dynamics (intonation, energy, timing) alongside anatomy

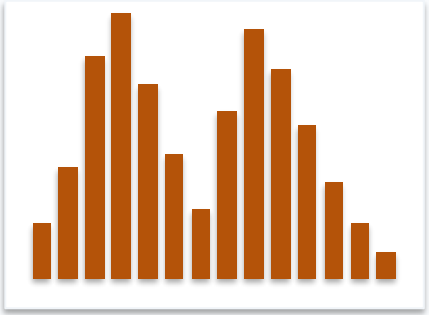
Pitch & Intonation

Fundamental frequency (F_0) patterns over time reflecting speech melody and style.



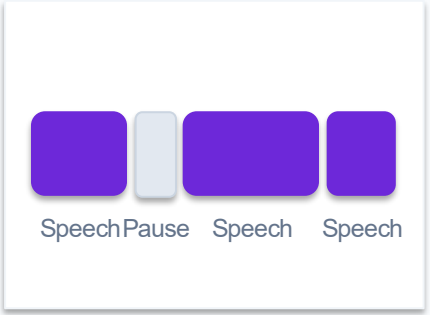
Energy & Stress

Short-time signal intensity and volume accents highlighting behavioral emphasis.



Duration & Timing

Temporal structures including speech rate, syllabic rhythms, and pause profiles.

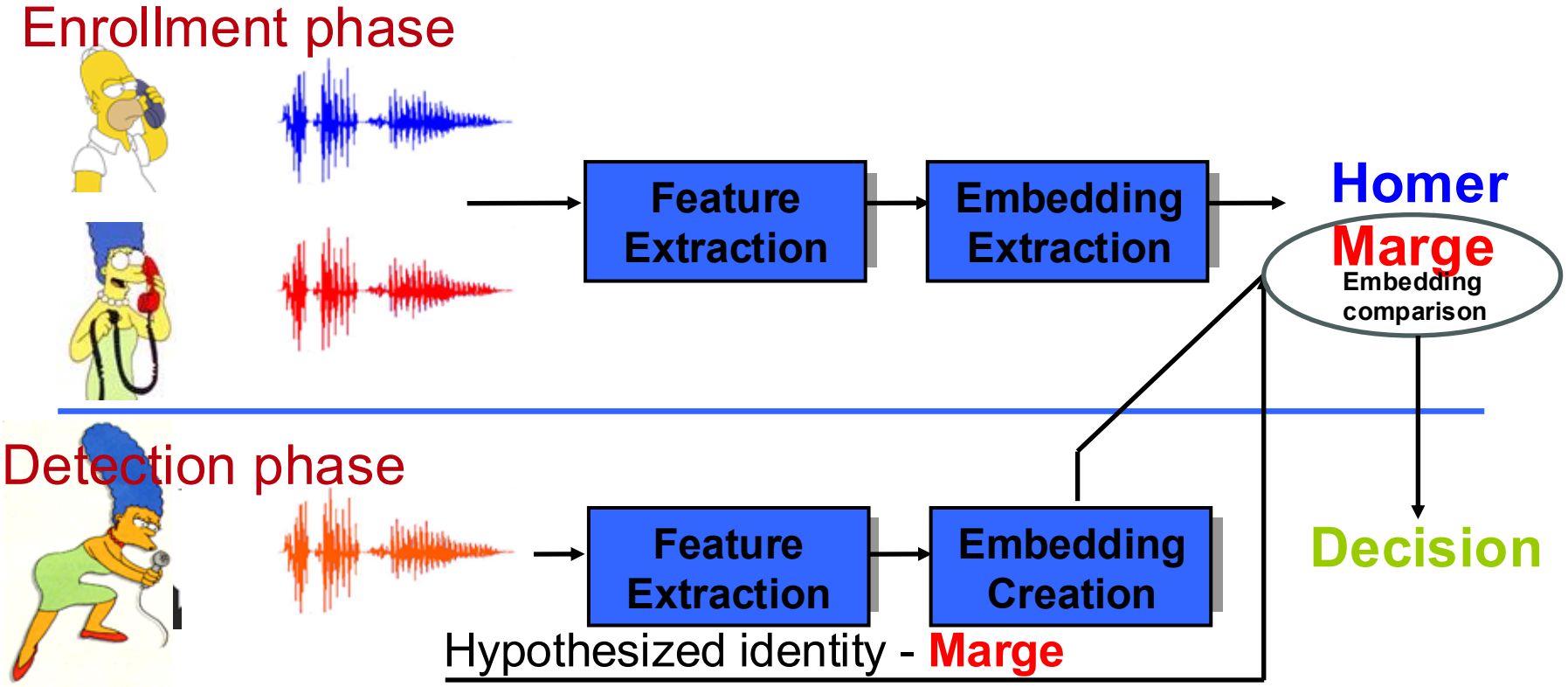


High level features are hidden in the speech content

- **Semantics & Word Choice**
  - **Vocabulary Preferences:** Frequent use of specific synonyms, technical jargon, or regional slang.
  - **Topic Familiarity:** Recurring themes, references, or subject matter expertise that narrow down identity.
- **Idiolect (Individual Language System)**
  - **Syntactic Patterns:** Unique grammatical structures, sentence lengths, or structural habits preferred by the speaker.
  - **Habitual Phrases:** Catchphrases, discourse markers (e.g., "basically," "you know"), and filler words used consistently.
- **Behavioral Idiosyncrasies**
  - **Disfluency Profiles:** Distinct patterns of pausing, stuttering, or self-correction during spontaneous speech.
  - **Expressive Traits:** Tendencies toward irony, emphasis, or specific emotional valences in communication.

Phases of Speaker Detection System

Two distinct phases to any speaker detection system

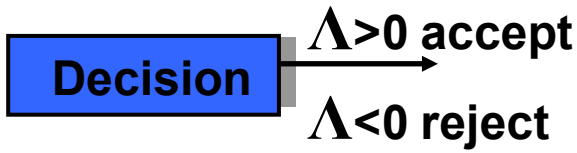


Basic structure of the system – Likelihood ratio

Speaker detection decision approaches have roots in signal detection theory

- **2 class Hypothesis test**
  - H0: the speaker is **not** the target speaker
  - H1: the speaker is the target speaker
- **Statistic computed on test utterance S as likelihood ratio**

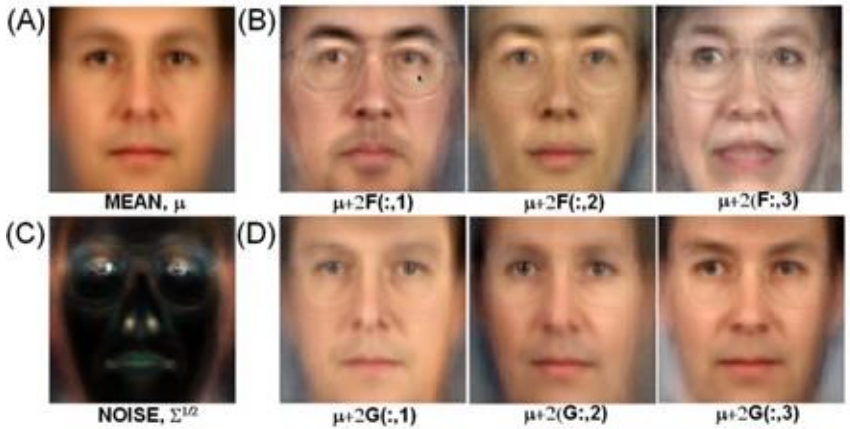
$$\Lambda = \log \frac{\text{Likelihood } \mathbf{S} \text{ came from speaker model}}{\text{Likelihood } \mathbf{S} \text{ did } \mathbf{not} \text{ come from speaker model}}$$



PLDA continued

- PLDA has nice interpretation in face verification where it was introduced by Simon J.D. Prince
- Each face image $\mathbf{i}$ can be constructed by adding
  - (A) mean face $\boldsymbol{\mu}$
  - (B) linear combination of basis $\mathbf{V}$ corresponding to between-individual variability (moving from $\boldsymbol{\mu}$ in these directions gives us images that look like different people)
  - (D) linear combination of basis $\mathbf{U}$ corresponding to within-individual variability (moving from $\boldsymbol{\mu}$ in these directions gives us images that look like the same person under different condition – e.g. lighting)
  - (C) residual noise vector $\boldsymbol{\epsilon}$

$$\mathbf{i} = \boldsymbol{\mu} + \mathbf{V}\mathbf{y} + \mathbf{U}\mathbf{x} + \boldsymbol{\epsilon}$$



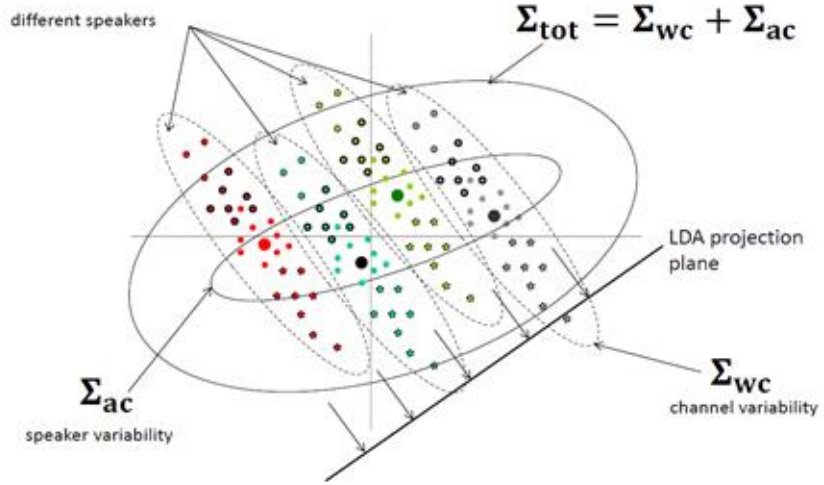
Picture taken from: S.J.D. Prince and J.H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," ICCV, 2007

Simplified PLDA model

Speaker Labelled embeddings are required for training the the probabilistic "backend" (typically PLDA).

The verification score is a log likelihood ratio of the utterances being generated jointly from the same speaker or independently from different speakers

$$s = \log \frac{l\left(\mathcal{X}_{e_1} \ldots \mathcal{X}_{e_n}, \mathcal{X}_{t_1} \ldots \mathcal{X}_{t_m} | H_s\right)}{l\left(\mathcal{X}_{e_1} \ldots \mathcal{X}_{e_n} | H_s\right) l\left(\mathcal{X}_{t_1} \ldots \mathcal{X}_{t_m} | H_s\right)}$$



- David Snyder et al., INTERSPEECH 2017

Deep Neural Network Embeddings for Text-Independent Speaker Verification

David Snyder, Daniel Garcia-Romero, Daniel Povey, Sanjeev Khudanpur

Center for Language and Speech Processing & Human Language Technology Center of Excellence,
The Johns Hopkins University, USA

{david.ryan.snyder, dpovey}@gmail.com, {dgromero, khudanpur}@jhu.edu

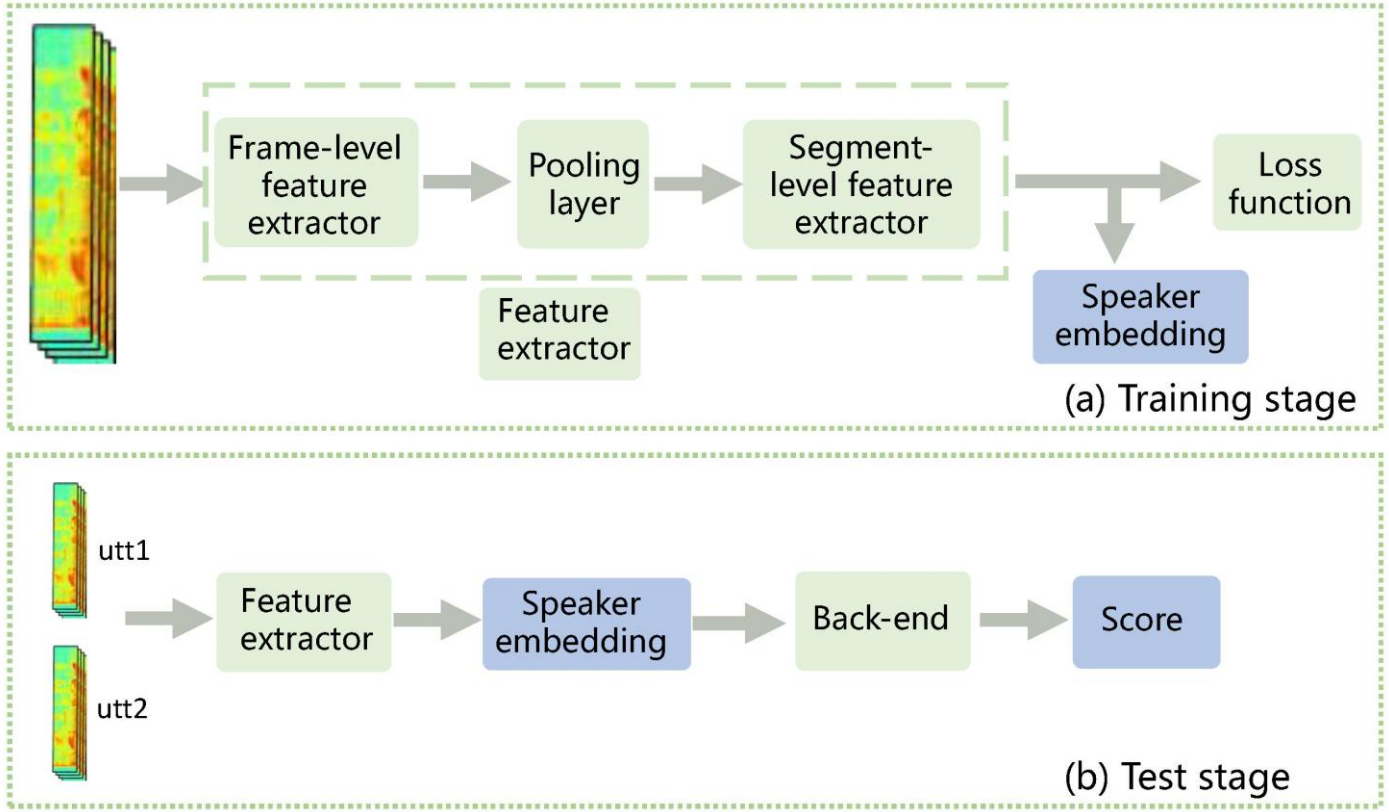
Abstract

This paper investigates replacing i-vectors for text-independent speaker verification with embeddings extracted from a feed-forward deep neural network. Long-term speaker characteristics are captured in the network by a temporal pooling layer that aggregates over the input speech. This enables the network

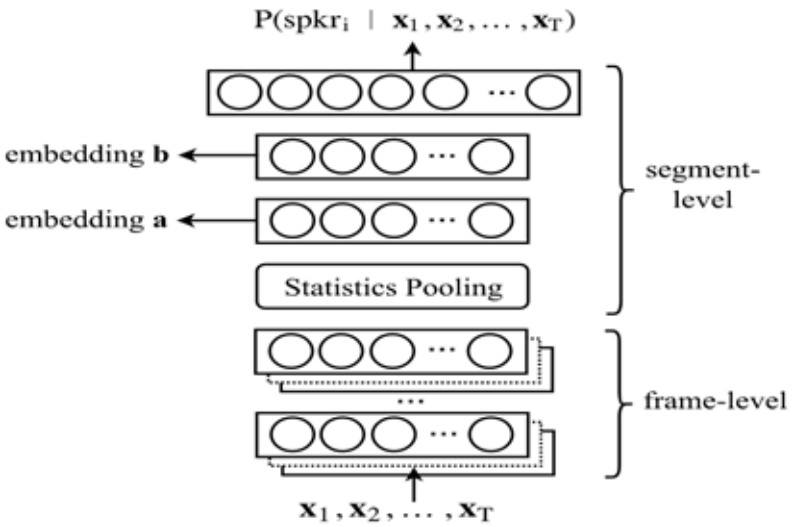
model (UBM) that is used to collect sufficient statistics, a large projection matrix to extract i-vectors, and a probabilistic linear discriminant analysis (PLDA) backend to compute a similarity score between i-vectors [2, 3, 4, 5, 6, 7].

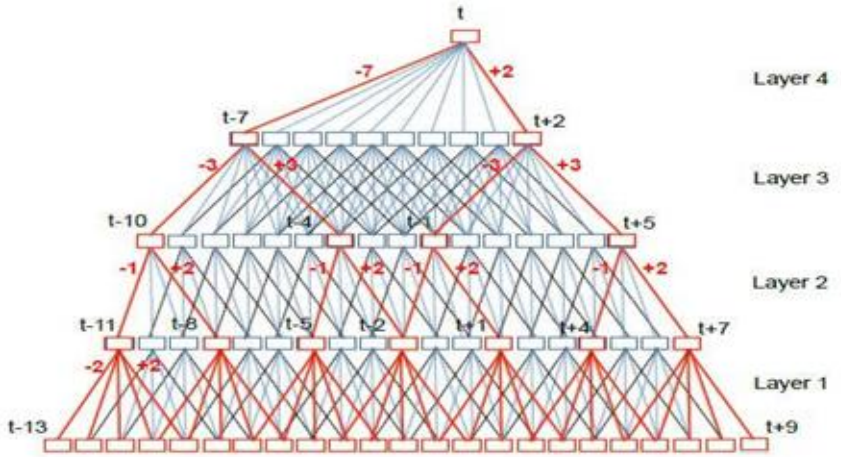
Traditionally, the UBM is a Gaussian mixture model (GMM) trained on acoustic features. Recent work has shown

General pipeline during embedding extractor training



Speaker-discriminative DNNs, x-vectors





gure 1: Computation in TDNN with sub-sampling (red) and ithout sub-sampling (blue+red)

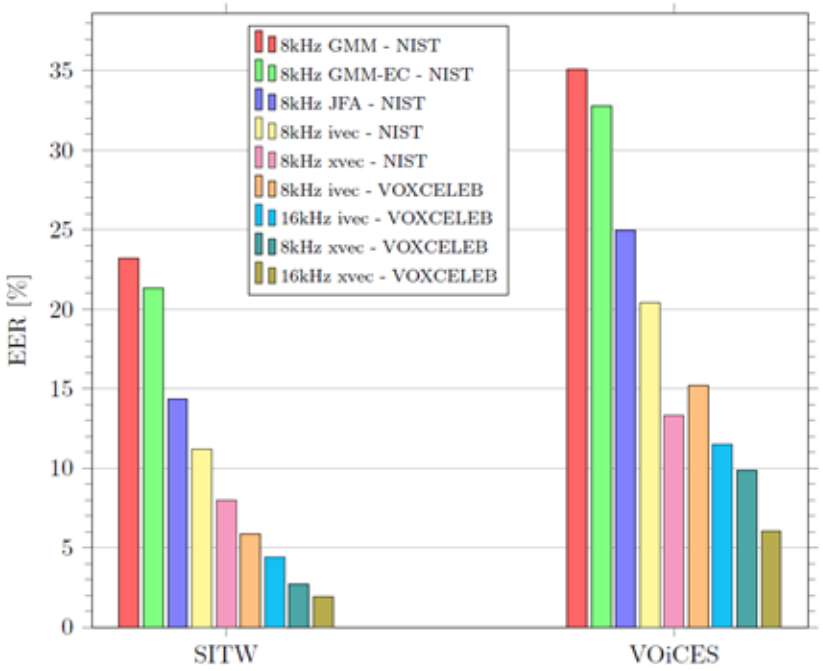
7 layers of Time-delay architecture (TDNN), Stats: mean and std, Embeddings: 512 and 300, Softmax output over supervised set of training speakers (few thousands), Trained with samples between 2 and 10 seconds

Peddinti, V., Povey, D., Khudanpur, S., *"A time delay neural network architecture for efficient modeling of long temporal contexts"* Proc. Interspeech 2015

Snyder, D., Garcia-Romero, D., Povey D., Khudanpur S. *"Deep Neural Network Embeddings for Text-Independent Speaker Verification"* , Interspeech 2017

Longitudinal Analysis

- Both SITW and Voices are 16K
- 2000 GMM-UBM
- 2006 GMM-EC
- 2008 JFA
- 2010 iVectors
- 2017 x-vectors
- Effect of in- vs out-domain data
- 8K vs 16K



Pavel Matějka, et al., *"13 years of speaker recognition research at BUT, with longitudinal analysis of NIST SRE"*, Computer Speech & Language, vol. 63, 2020

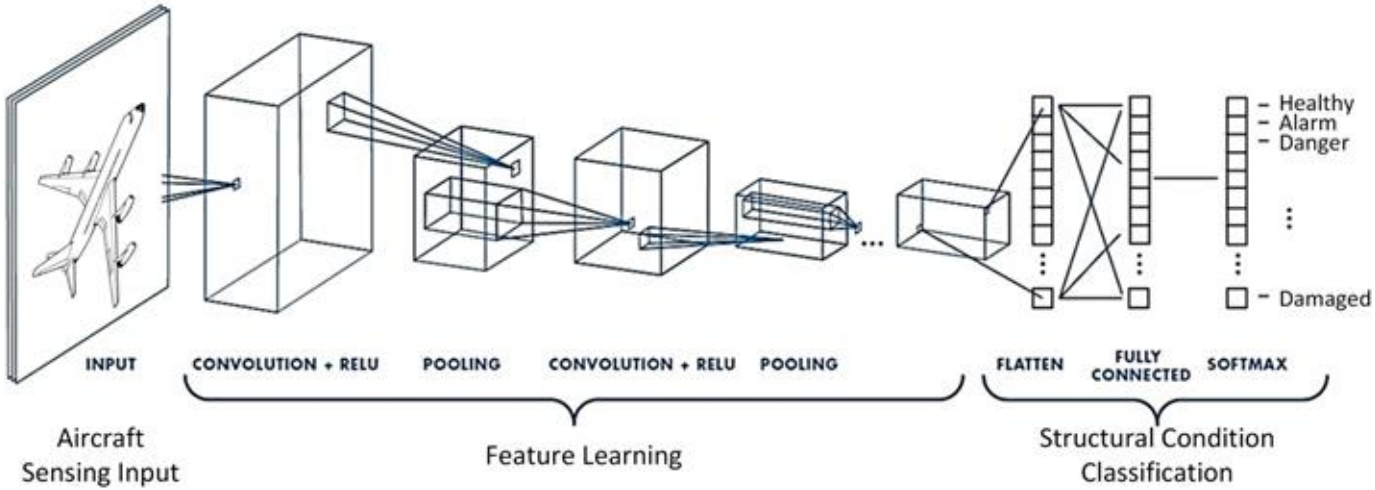
SOTA AND CURRENT RESEARCH DIRECTIONS

SOTA approach for feature learning - CNN

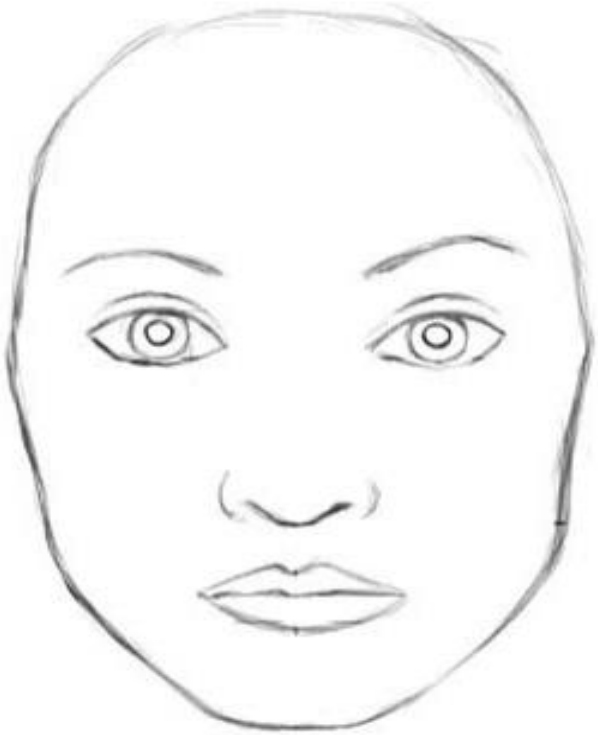
X-vectors with variants of large convolutional networks like ResNet, ECAPA-TDNN, RepVGG….

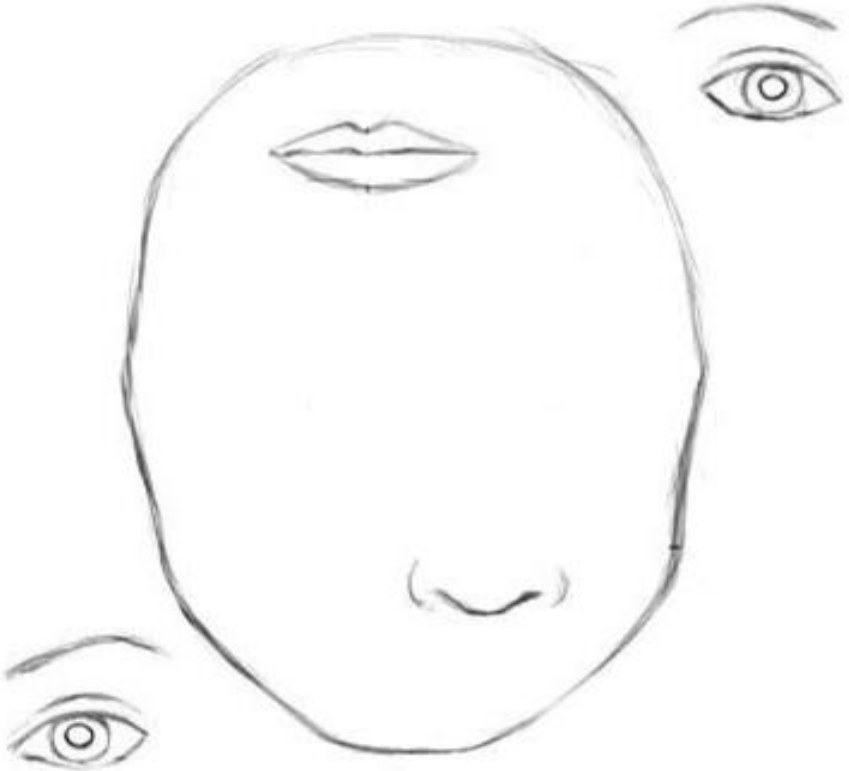
For coherent domains with large amounts of annotated data such as VoxCeleb

Voxceleb1-O is essentially solved, EERs << 1%



Attention vs CNN



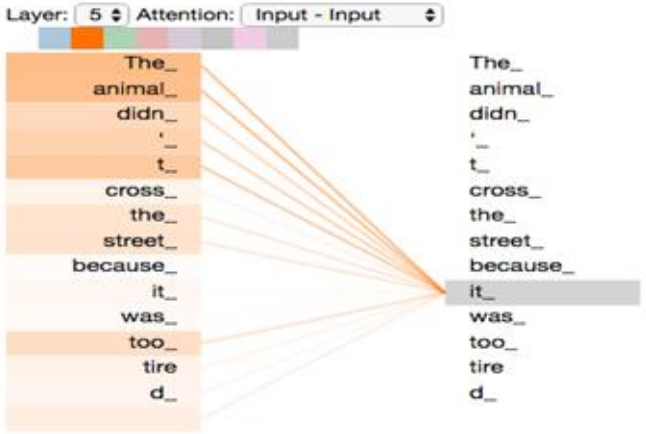


Attention

Instead of processing an entire audio wave or sentence identically, the **network assigns weights to different frames**
Viewed as a matrix, the attention weights show how **the network adjusts its focus according to context**.

General Attention

	I	love	you
Je	0.94	0.02	0.04
t’	0.11	0.01	0.88
aime	0.03	0.95	0.02



- General Attention between Inputs and outputs
- Within the input elements (Self-Attention)

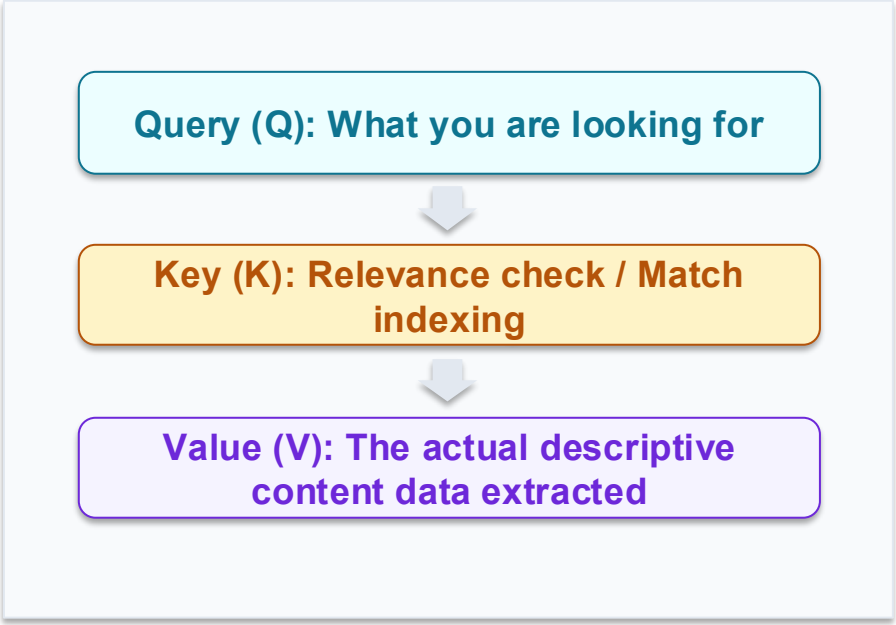
When the model is processing the word “it”, self-attention allows it to associate “it” with “animal”.

- Problems with streaming, memory issues

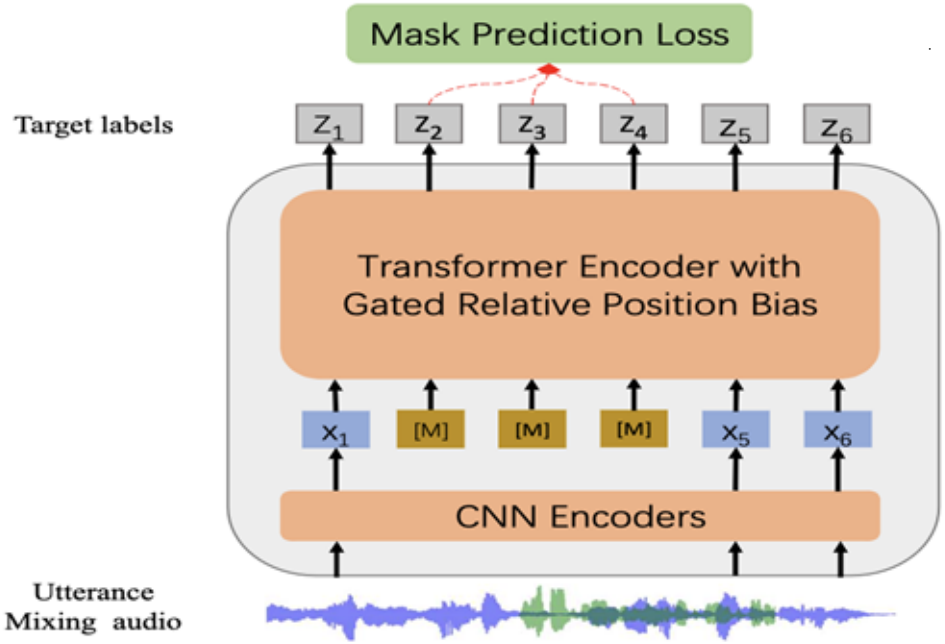
Jay Alammar - Visualizing machine learning one concept at a time., https://jalammar.github.io/illustrated-transformer/

How neural networks dynamically focus on the most relevant information within a sequence.

- **Dynamic Focus:** Instead of processing an entire audio wave or sentence identically, the network assigns weights to different frames.
- **Relevance Scoring:** Matches target demands against context indicators to decide what content to amplify or suppress.
- **Context Generation:** Blends information together mathematically to output an optimized, context-aware representation.



WavLM

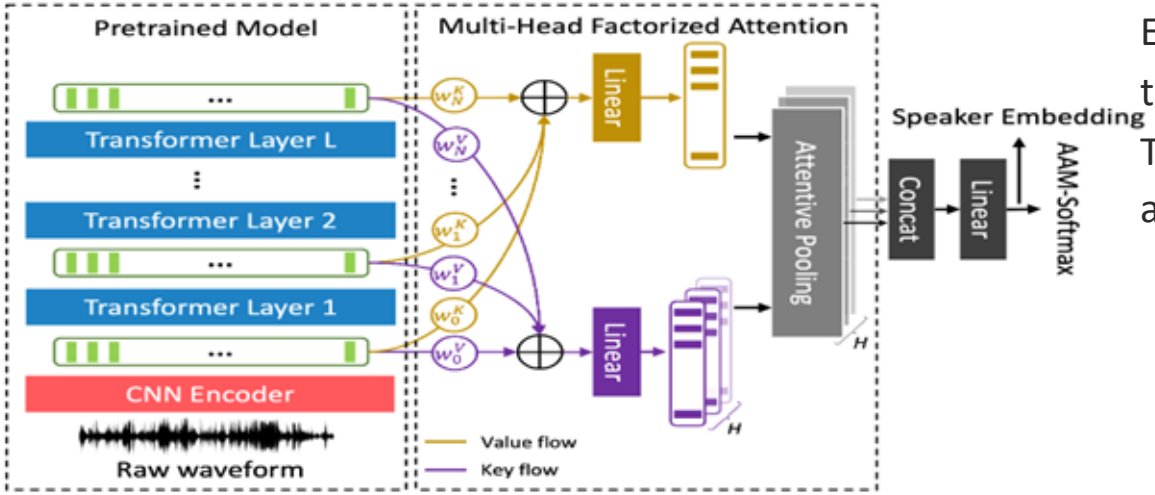


WavLM main characteristics

- Tailored to speaker recognition, diarization, speech separation, a.o.
- Jointly learns masked speech prediction and denoising.
- Simulate noisy/overlapped speech as inputs, and predict the pseudo-labels of original speech on the masked region.
- Replaces standard positional embeddings with gated relative position bias.

Chen, Sanyuan, et al. "WavLM: Large-scale self-supervised pre-training for full stack speech processing." (Microsoft) 2021

Attention-based SV backend on top of WavLM



Simple backend with multihead attention (64 heads).

Each head models an acoustic area via a trainable query vector.

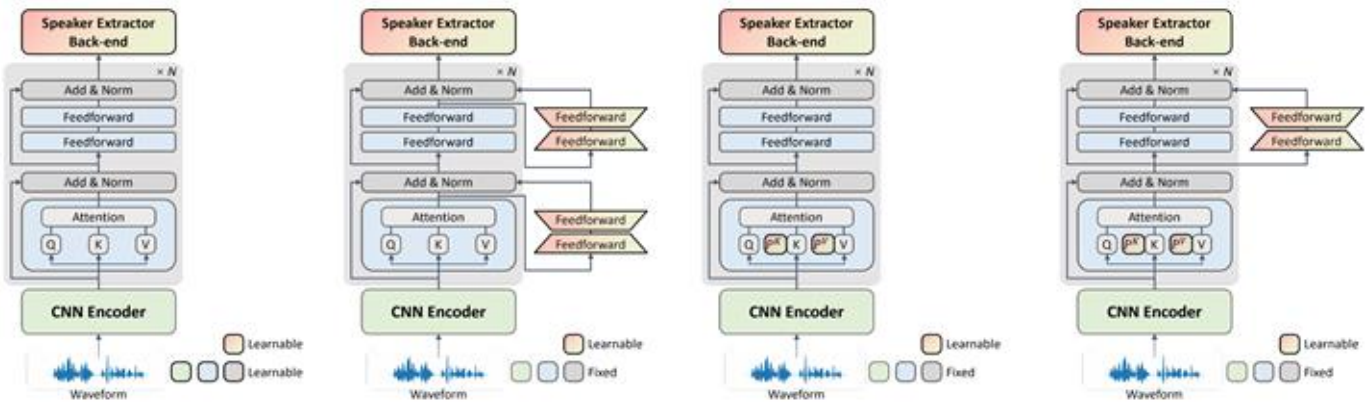
Two different layer-pooling weights (keys and values).

- Values should contain speaker information.
- Keys should not.

Peng, Junyi, et al. "An attention-based backend allowing efficient fine-tuning of transformer models for speaker verification." arXiv preprint arXiv:2210.01273 (2022)., SLT 2022

Transformer models and adapters for SV

- Do we need to finetune the whole SSL network?
- What if there is not enough in-domain data. (Experiments on VoxCeleb and CNCeleb)
- Interesting implications for concurrent deployment of many domain-specific models



(a) Full Fine-tuning (b) Bottleneck Adapter (c) Prefix Tuning (d) Mix-and-Match Adapter

Peng, Junyi, et al. ***"Parameter-efficient transfer learning of pre-trained Transformer models for speaker verification using adapters."*** *IEEE-ICASSP 2023*

Extracting speaker-related representations from

- **Not enough labelled data**
  - Utilize pre-training paradigm that leverages vast amount of unlabeled data (or download the model)
  - Pre-trained model can be easily fine-tuned for target application (domain)
  - In the end, less labeled data are needed w.r.t. CNN or RNN-based models
- **Plenty of labelled data**
  - Train large CNN-based supervised embedding extractors
  - Obtain SOTA results, but perhaps lose some robustness

Agenda

- Speaker recognition applications
  - Why we need speaker recognition
  - Different application domains
- Background and theory
  - Feature extraction
  - DNN embeddings

- Evaluation metrics and Wespeaker toolkit
- Conclusions

Evaluation Metric

- There are two types of errors

  MISS: incorrectly rejected a target trial

  It was target trial, but system said it is not

  also known as a false reject

  FALSE ALARM: incorrectly accept a non-target trial

  It was not target trial, but system said it is

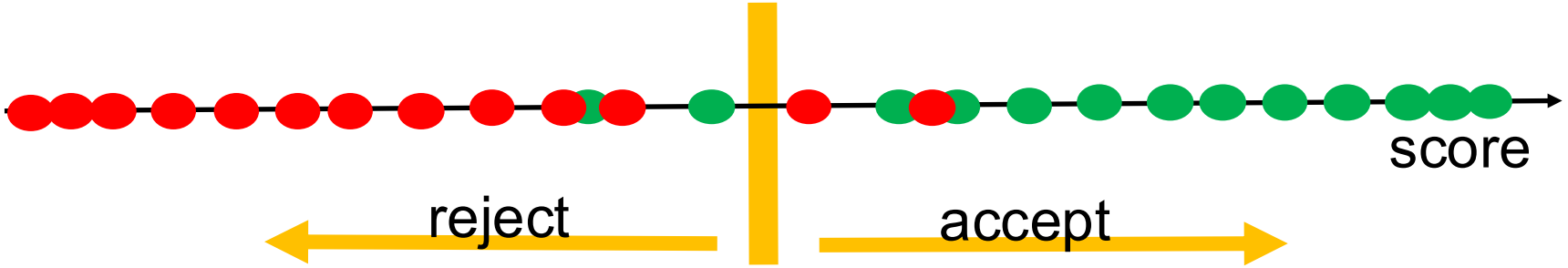
  also known as a false accept

- The performance of a detection system is measure of the trade-off between these two errors – is controlled by adjustment of the decision threshold
- In an evaluation, N_{target} target-trials (test speaker = model speaker) and $N_{non\text{-}target}$ non-target trials (test speaker != model speaker) are conducted and error probabilities are estimated at given threshold

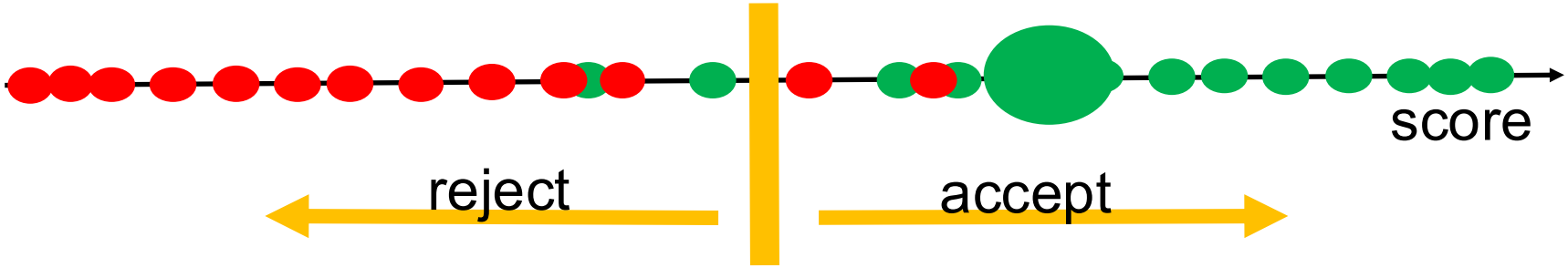
$$\Pr(\text{miss}|\text{thr}) = \frac{\#\text{ target trials scores} < \text{thr}}{N_{target}} \qquad \Pr(\text{false alarm}|\text{thr}) = \frac{\#\text{ non-target trials scores} > \text{thr}}{N_{non\text{-}target}}$$

Does the system work well ?

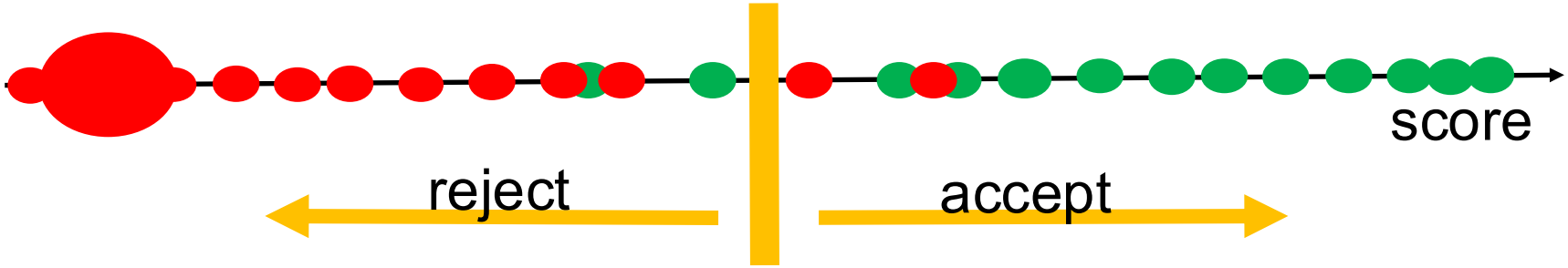
- We need some (lots of) data – pairs model speaker – test speaker (we call these pairs trials).
  - **Target-trials** (test speaker = model speaker)
  - **Non-target-trials** (test speaker ≠ model speaker)
- We run them through the system and record the scores
- We need to set the detection **threshold**



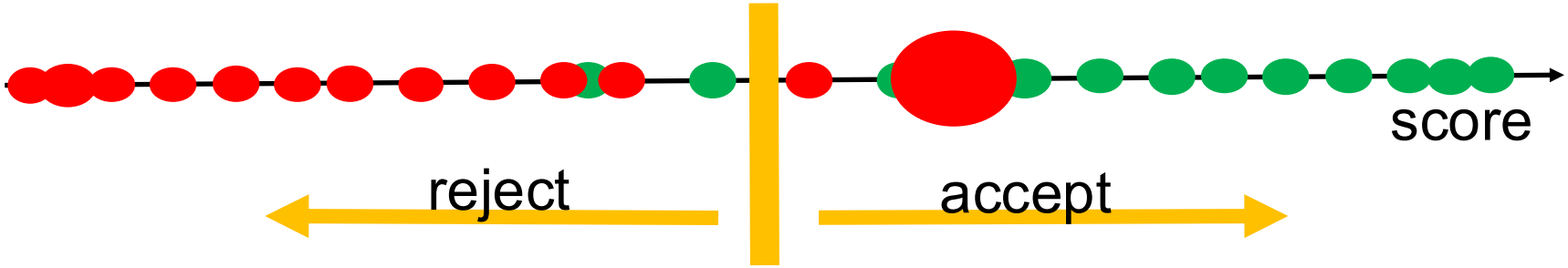
True accept



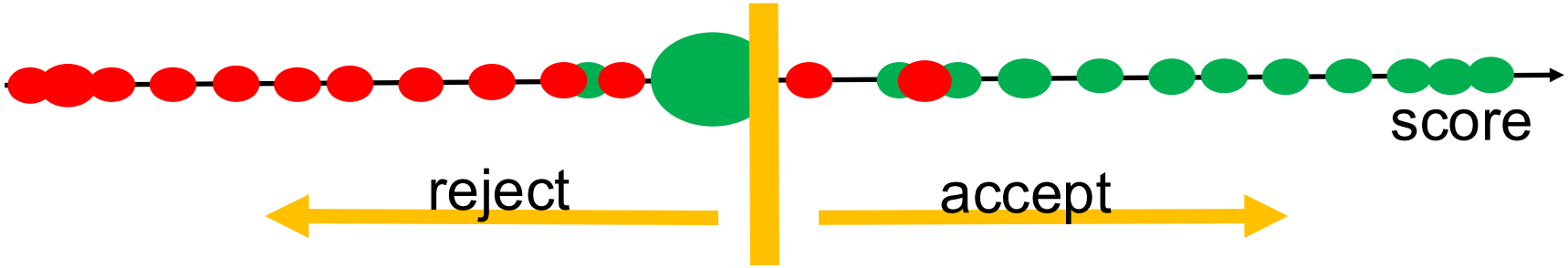
True reject



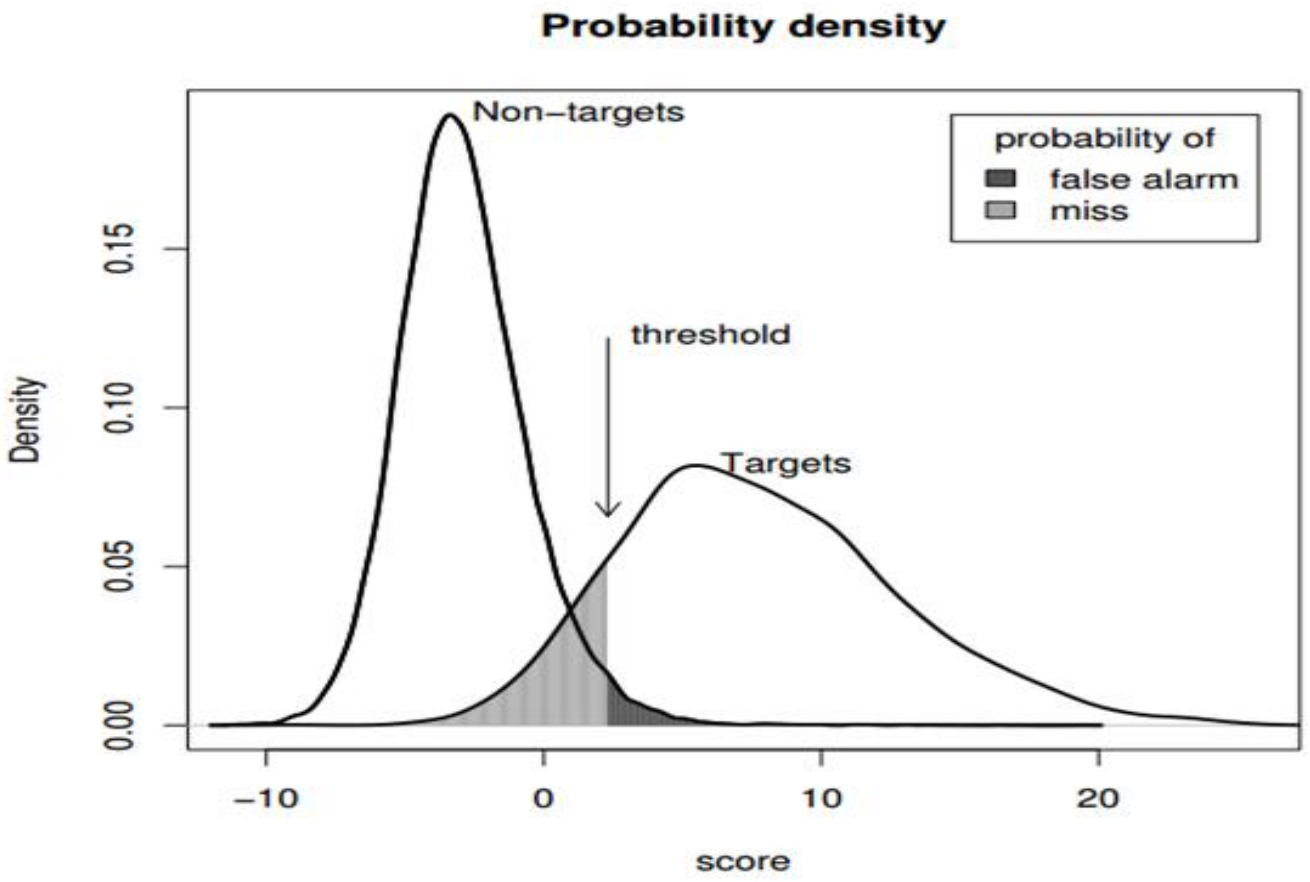
False alarm



Miss

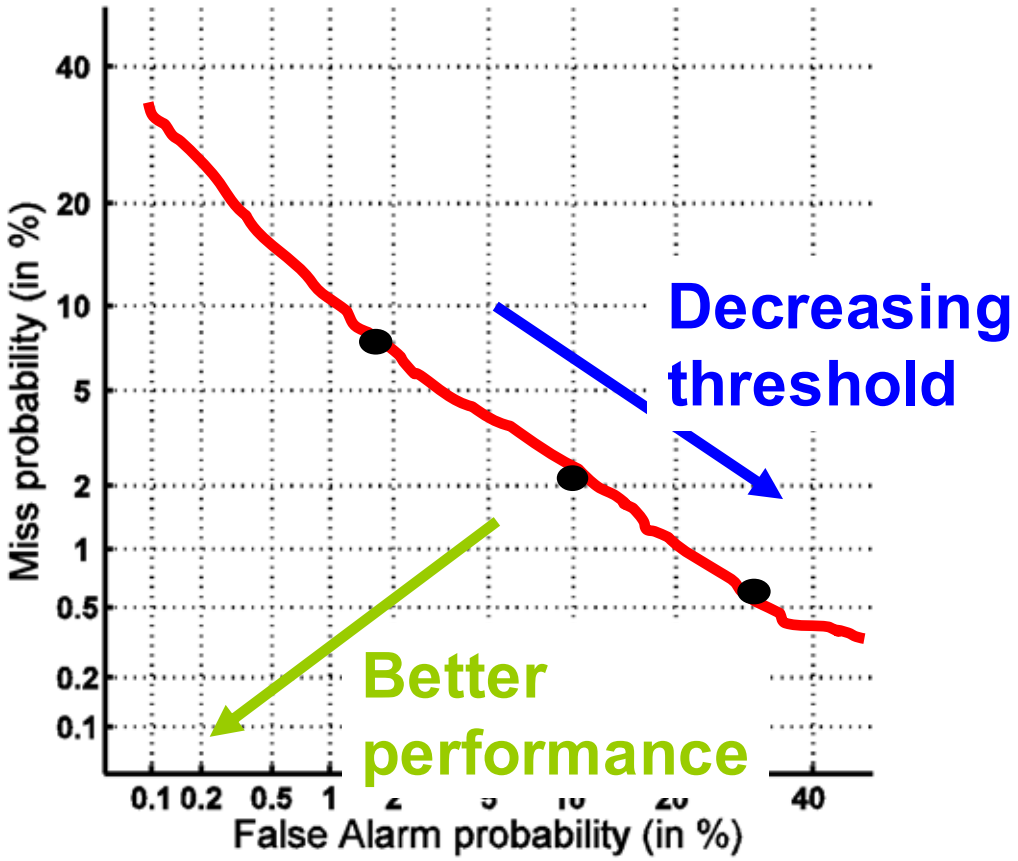


Single threshold, two types of error



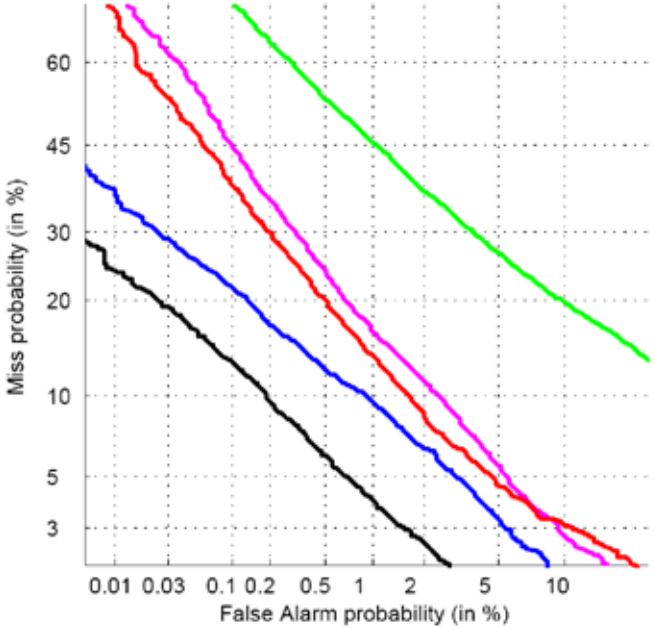
False accept
False reject

DET – Detection Error Tradeoff



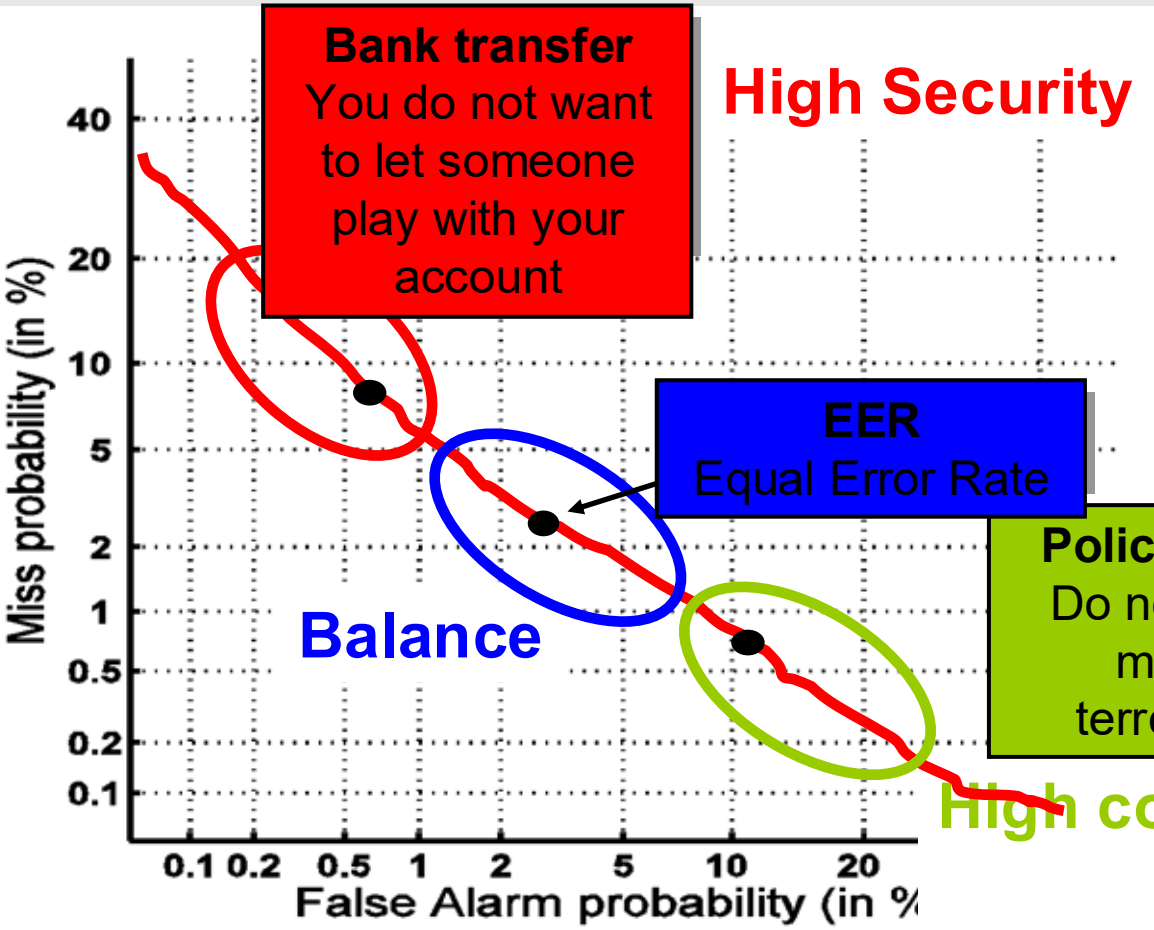
Comparison of different SRE techniques

NIST SRE 2010, tel-tel (cond. 5)



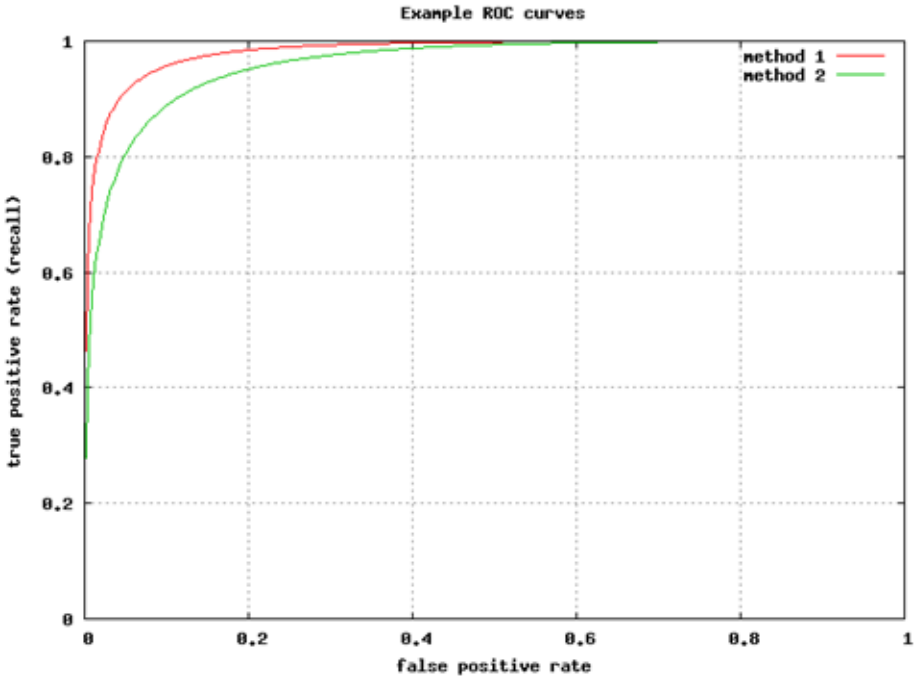
- Baseline (relevane MAP)
- Eigenchannel adapt.
- JFA
- iVector+PLDA

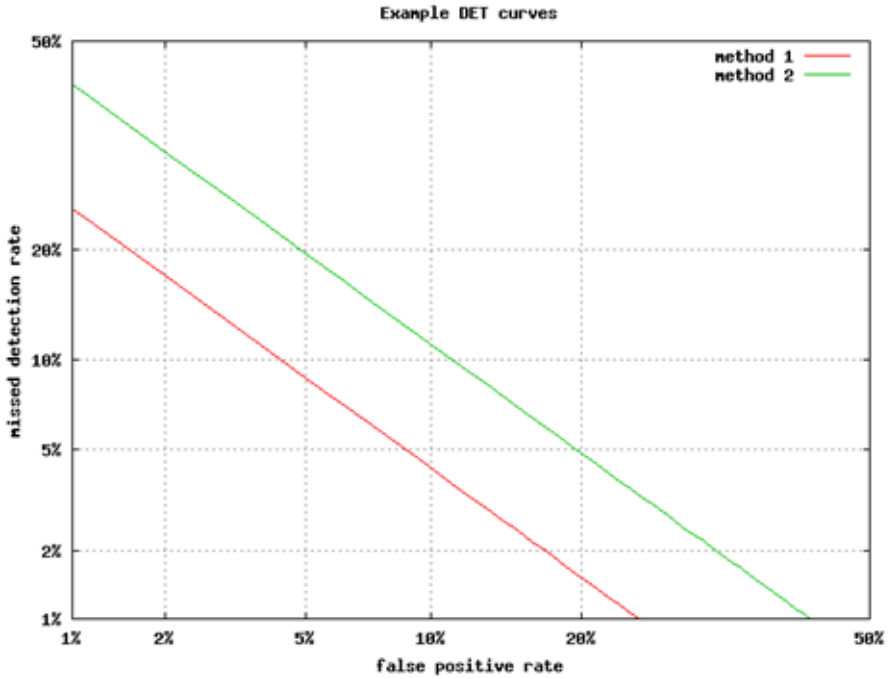
DET – Detection Error Tradeoff



Equal Error Rate (EER) = operating point of the system where it makes the same number of false alarms and misses

ROC curve vs DET curve





Wespeaker – training speaker embedding extractors

- Clone the repository and install the framework dependencies (PyTorch and Torchaudio)

```
git clone https://github.com/wenet-e2e/wespeaker.git
cd wespeaker
pip install -r requirements.txt
```

Wespeaker – prepare data

- Prepare your dataset (such as VoxCeleb or your own audio files) on disk.
- Look around in the recipe, search for wav.scp, utt2spk.scp

- **wav.scp** : Maps a unique utterance ID to its audio file path.
- **utt2spk:** Maps each utterance ID to the matching speaker ID.

Wespeaker – configure the setup

- Navigate to a pre-built directory (like examples/voxceleb/v2/).
- You modify run.sh and the configuration file (config.yaml) to set up your experiment
  - Model backbone (e.g., ECAPA-TDNN or ResNet34)
  - and loss function (e.g., AAM-Softmax).

Wespeaker – train

- Execute the master script (`bash run.sh --stage 1 --stop_stage 3`). Wespeaker executes an on-the-fly "Unified IO" data loop:
  - **On-the-fly Augmentation:** Artificially corrupts raw audio with noise, reverb, or speed perturbation.
  - **Acoustic Features:** Converts raw PCM waveforms directly into log-mel filterbank (Fbank) frames.
  - **Temporal Aggregation:** Passes frames through neural layers to aggregate temporal features into fixed-size speaker embeddings

Wespeaker – test

- Once the training script finishes and saves the model checkpoints, you can use the built-in command-line tools to extract embeddings from new audio
  - You can run more phases in the script to perform final model averaging, or run batch evaluation with predefined test set and compute EER, DCF, etc
  - You can also take the model (checkpoint), two audio files and compare them whether they come from the same speaker or not)
- `wespeaker --task similarity --audio_file voice1.wav --audio_file2 voice2.wav`

  `Or just compute embeddings:`

- `wespeaker --task embedding --audio_file target_voice.wav --output_file embedding.txt`

Conclusions

- Over the past decade
  - error rates have decreased 5 times or more thanks to advances in channel compensation techniques
  - the new techniques have resulted in massive speed-ups
    - i-vector/x-vectors are extracted for each recording (about 100 times faster than RT)
    - With embeddings, **billions** of verification trials can be tested in few seconds
- DNN embeddings have replaced generative embeddings (i-vectors)
- Use of large SSL models is promising when little annotated training data are available for training DNN embedding extractors

- **For more details talk to anybody from Speech@FIT**

Thanks for your attention
and
I hope you enjoyed it ;)