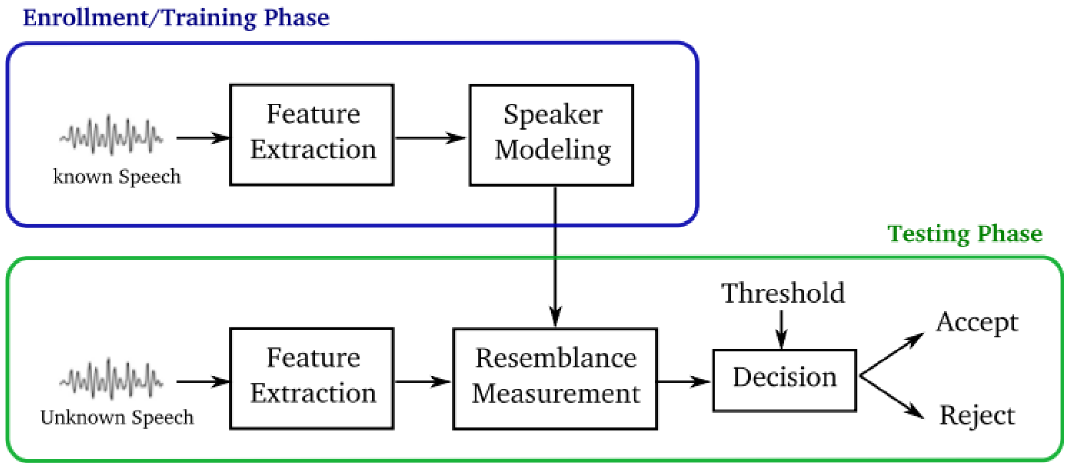
Deep Speaker Recognition: Modular or Monolithic?

Gautam Bhattacharya, Jahangir Alam, Patrick Kenny

Abstract Version

1. Introduction

- Goal
  - In this work, they analyze the performance of end-to-end deep speaker recognizers on two popular text-independent tasks: NIST-SRE 2016 and VoxCeleb.
    - NIST-SRE 2016 → https://www.nist.gov/itl/iad/mig/speaker-recognition-evaluation-2016
    - VoxCeleb → https://www.robots.ox.ac.uk/~vgg/data/voxceleb/



2. Related Works

- Two tasks and approaches
  - X-vector [Snyder *et al*, 2018] model has established as the state-of-the-art in recent NIST evaluations.
    - Speaker embeddings are used to train a probabilistic linear discriminant analysis (PLDA) classifier.

    → Modular

Layer	Layer context	Total context	Input x output
frame1	$[t-2, t+2]$	5	120x512
frame2	$\{t-2, t, t+2\}$	9	1536x512
frame3	$\{t-3, t, t+3\}$	15	1536x512
frame4	$\{t\}$	15	512x512
frame5	$\{t\}$	15	512x1500
stats pooling	$[0, T)$	T	$1500T$x3000
segment6	$\{0\}$	T	3000x512
segment7	$\{0\}$	T	512x512
softmax	$\{0\}$	T	512xN

  - ResNet models have been widely adopted for learning speaker embedding models, especially for the VoxCeleb task [Chung *et al*, 2016] [Cai *et al*, 2018]

    → Monolithic

3. Proposed Methods

- Modular framework
  - They propose a modular approach that draws inspiration from the x-vector/PLDA recipe.
  - Unlike the x-vector model, their approach uses a second neural network instead of PLDA.

  - Procedures
    - Step 1: Train a speaker embedding model by minimizing the softmax loss. → extract speaker embeddings from entire training dataset.

    - Step 2: Train a small classifier using the embeddings extracted in step 1 as input. → extract speaker embeddings from this second model to perform speaker verification.

3. Proposed Methods

- Main factors
  - Deep residual neural network feature extractor (step 1)
  - Self-attention (step 1)
  - Large margin loss function (step 2)
  - Feature normalization (step 2)

- The Technical details and result → in the paper !