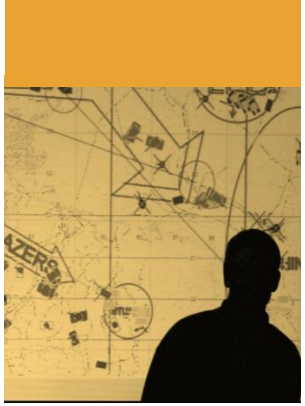
SRI International



Application of Convolutional Neural Networks to Language Identification in Noisy Conditions

Yun Lei, Luciana Ferrer, Aaron Lawson, **Mitchell McLaren**, Nicolas Scheffer

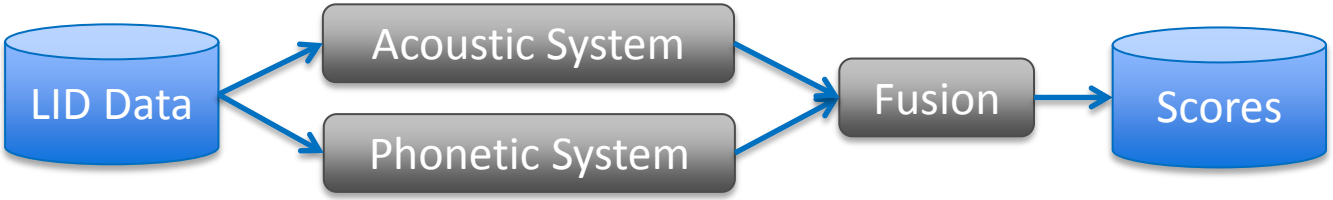
Outline

- Background
- The DNN/i-vector framework (proposed for SID)
- The CNN extension to channel-degraded LID
- The simpler, superior CNN/posterior system for LID
- Experiment setup
- Results

Background

Language identification (LID)

- The UBM/i-vector framework is widely used in LID
- Phone recognizers also achieve nice performance
- The fusion between the acoustic-based approach and phonetic-based approach results in a significant improvement.



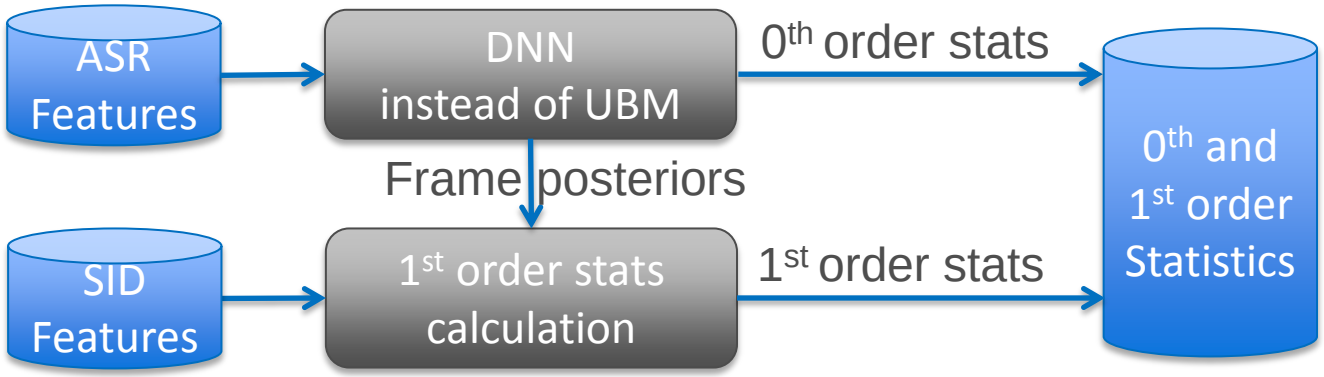
The challenge?

- To find a single system that combines the best of both worlds

We recently solved this challenge for Speaker ID (SID)!

Background: The DNN/i-vector framework

- Combines the Deep Neural Network (DNN) trained for Automatic Speech Recognition (ASR) and the popular i-vector model



- Replaces the UBM with a DNN from ASR

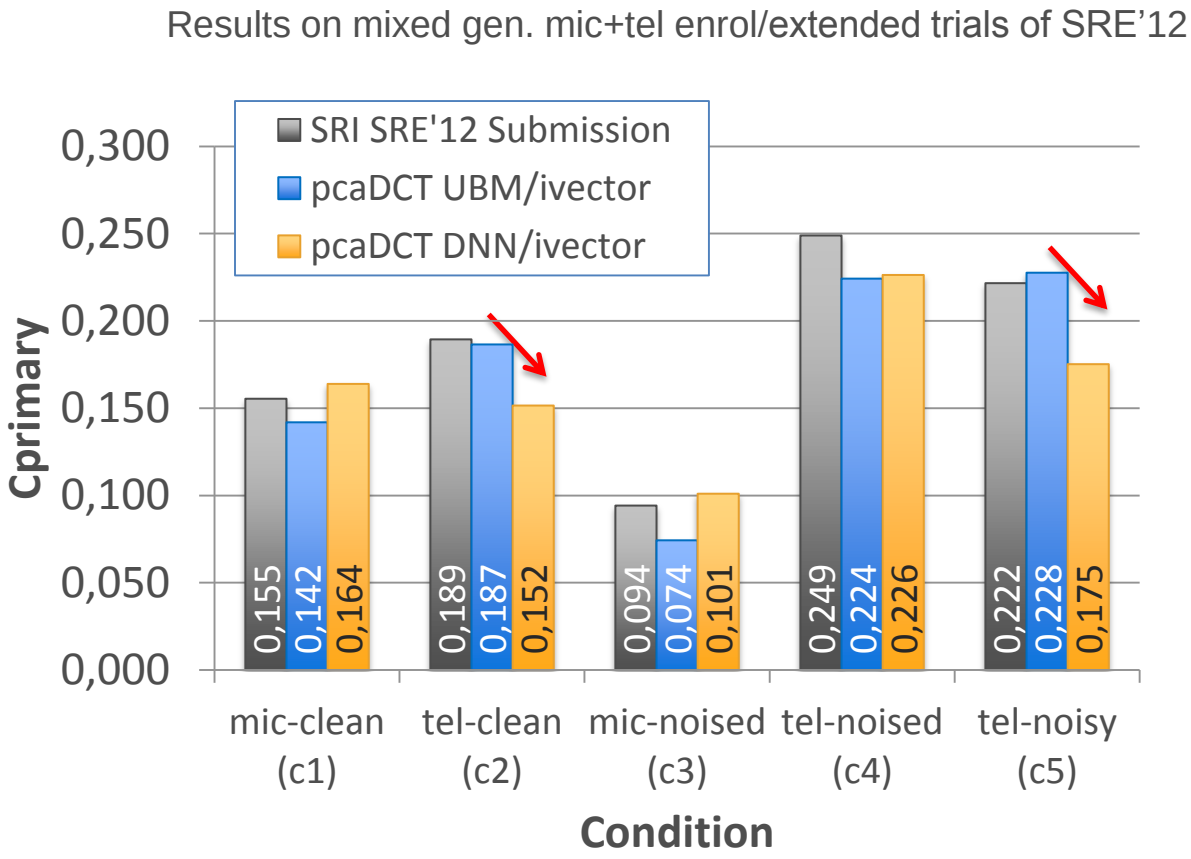
UBM vs DNN:

- UBM trained (unsupervised) and classes (components) assumed to map to phonetic classes
- DNN trained (supervised) to map classes to ‘senones’
  - Senones represent a tied 3-phone state

Background: The DNN/i-vector framework

It's very powerful for SID:

- Achieves a significant 30% relative imp. in NIST SRE12 **telephone** conditions in [Lei'14]
- Work to be done for microphone trials
- A single system that beats our SRE'12 submission



Can it be used for LID?

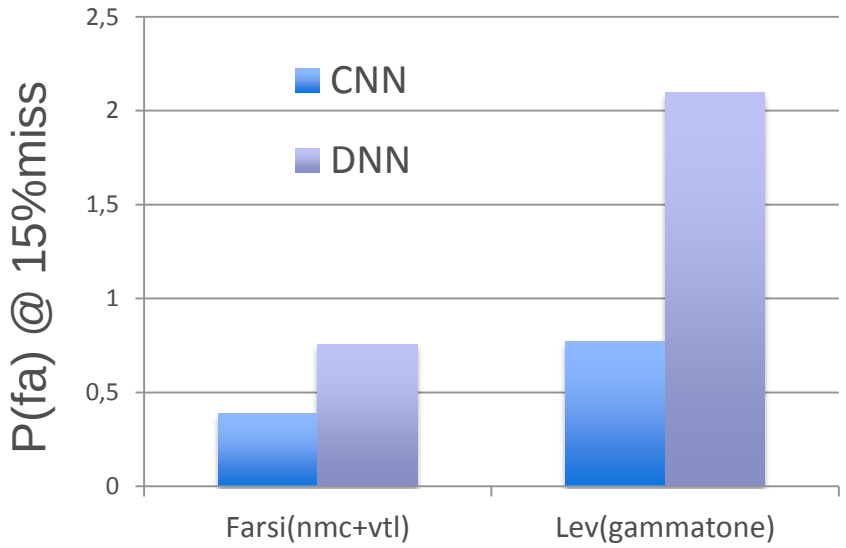
- The output of the DNN should include the language related information and more robust against speaker/noise distortions.

[Lei'14] Y. Lei et al., "A NOVEL SCHEME FOR SPEAKER RECOGNITION USING A PHONETICALLY-AWARE DEEP NEURAL NETWORK", Proc. ICASSP 2014.

DNNs for channel-degraded LID?

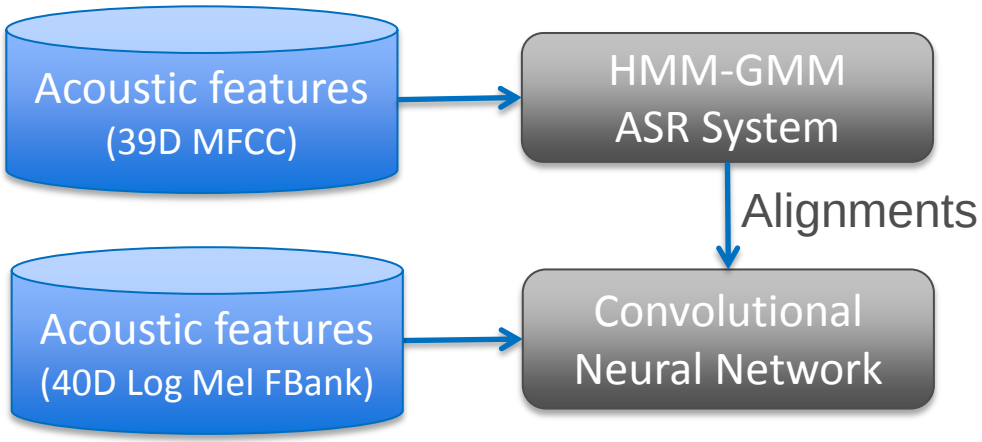
DNN vs CNN:

- Deep **convolutional** neural network (CNN) significantly outperforms DNN in noisy conditions on ASR related tasks
  - Comparison on RATS KWS task:



- We therefore attempt to use a **CNN for channel-degraded LID**

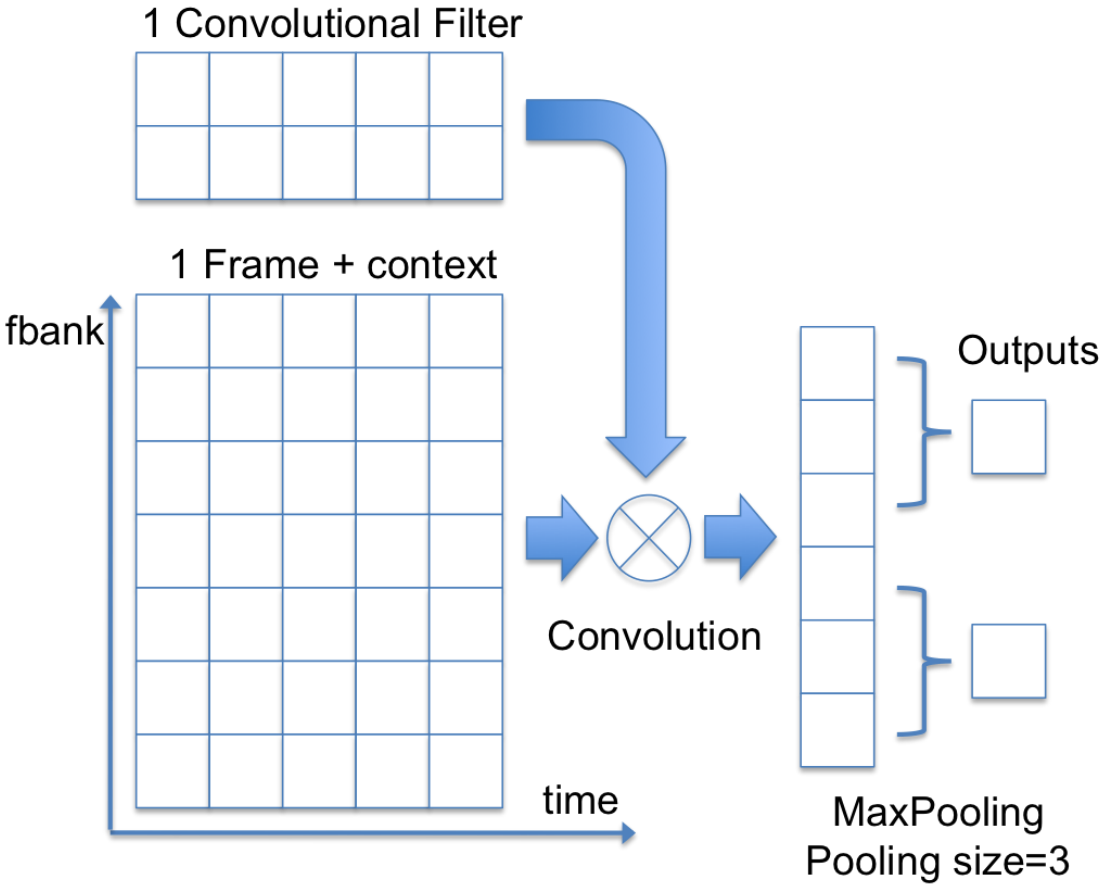
Training an CNN for ASR



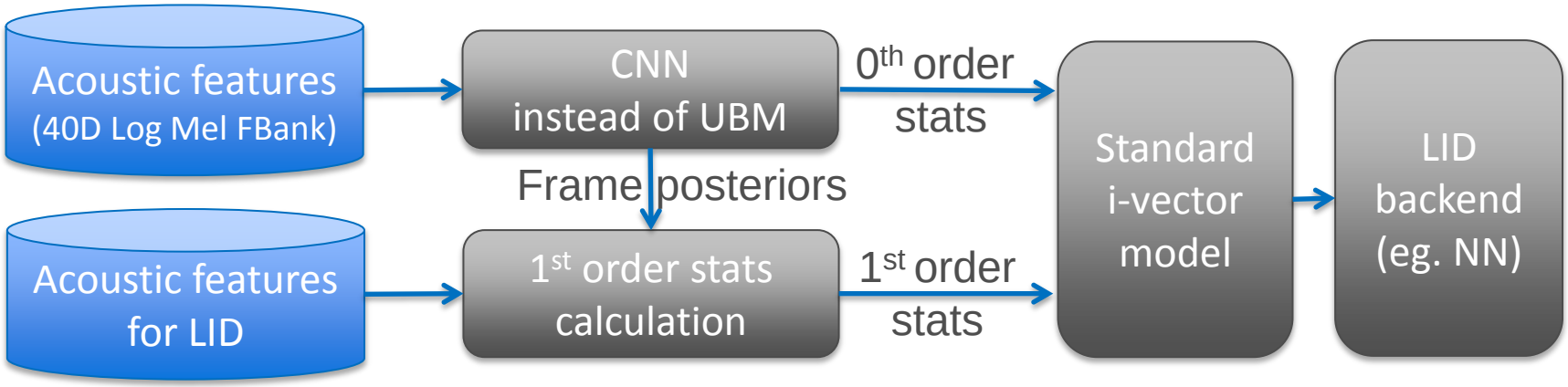
- Same training flow as with DNN training
- The features of 15~30 frames are concatenated as input
- A decision tree (trained for ML) is used to define senones.
- A pre-trained HMM-GMM is used to generate training alignments.

CNN in ASR

- An example of the CNN layer structure



CNN/i-vector system

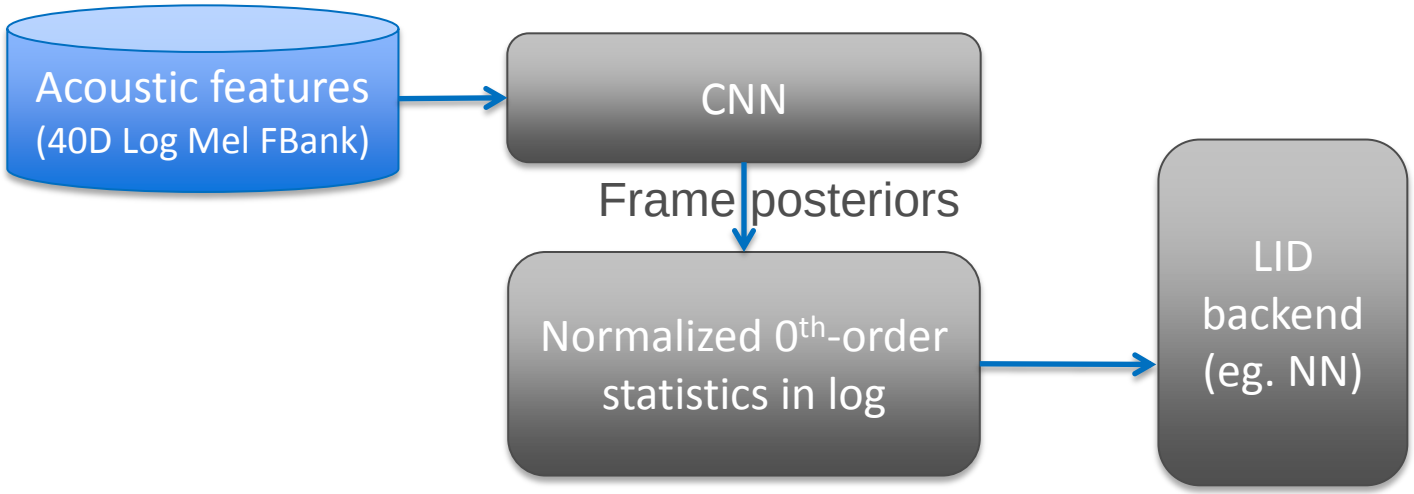


- Senones from the CNN define the classes for posteriors.

The benefit of LID features being independent of CNN features:

- A multi-feature system could use the same posteriors (normally one UBM-specific set of posteriors per feature).
- Tuning of features is no longer a balancing act between stable posteriors and LID performance.

CNN/posterior: An alternative, simpler system



Proposed **CNN/posterior** system, inspired by phonetic systems:

- Count of tied context-dependent state (tri-phone)
- State-level instead of phone-level
- CD tied phone state => Replace N-gram expansion
- Can use standard LID backends

Experiment Setup: Data

Evaluated on DARPA RATS LID task

- 5 target languages (and 10 unknown languages)
  - Farsi, Urdu, Pashto, Arabic Levantine, Dari
- Severe channel degradation:
  - Clean telephony speech retransmitted over 7 channels
  - Signal-to-noise ratio (SNR) between 30dB and 0dB
- The **transcription** used for CNN training is available for only two target languages in KWS task
  - Farsi and Arabic Levantine

Test durations: 3, 10, 30, 120 seconds

Metric: Average EER across target languages

Experiment Setup: Models

HMM-GMM Model

- 39D MFCC, 3353 senones with 200k Gaussians)
- Maximum likelihood training
- Multilingual training on Farsi and Levantine Arabic

CNN Model

- Input: 40 filter banks, 15 stacked frames
- Convolutional layer: 200 filters, pooling size=3, filter size=8
- 5 hidden layers following the convolutional layer
- Multilingual training on Farsi and Levantine Arabic

UBM Model: 2048-component UBM

LID features: 140-Dimensional 2D-DCT log MelSpectra feature optimized for RATS SID task. (Similar to zig-zag DCT proposed in [McLaren'14])

[McLaren'14] M. Mclaren, N. Scheffer, and Y. Lei, *"Effective use of DCT coefficients for contextualizing features for speaker recognition."*, Proc. ICASSP 2014.

Experiment Setup: Vectors and Backend

CNN and UBM i-vectors

- 400D i-vector subspace trained on LID training data

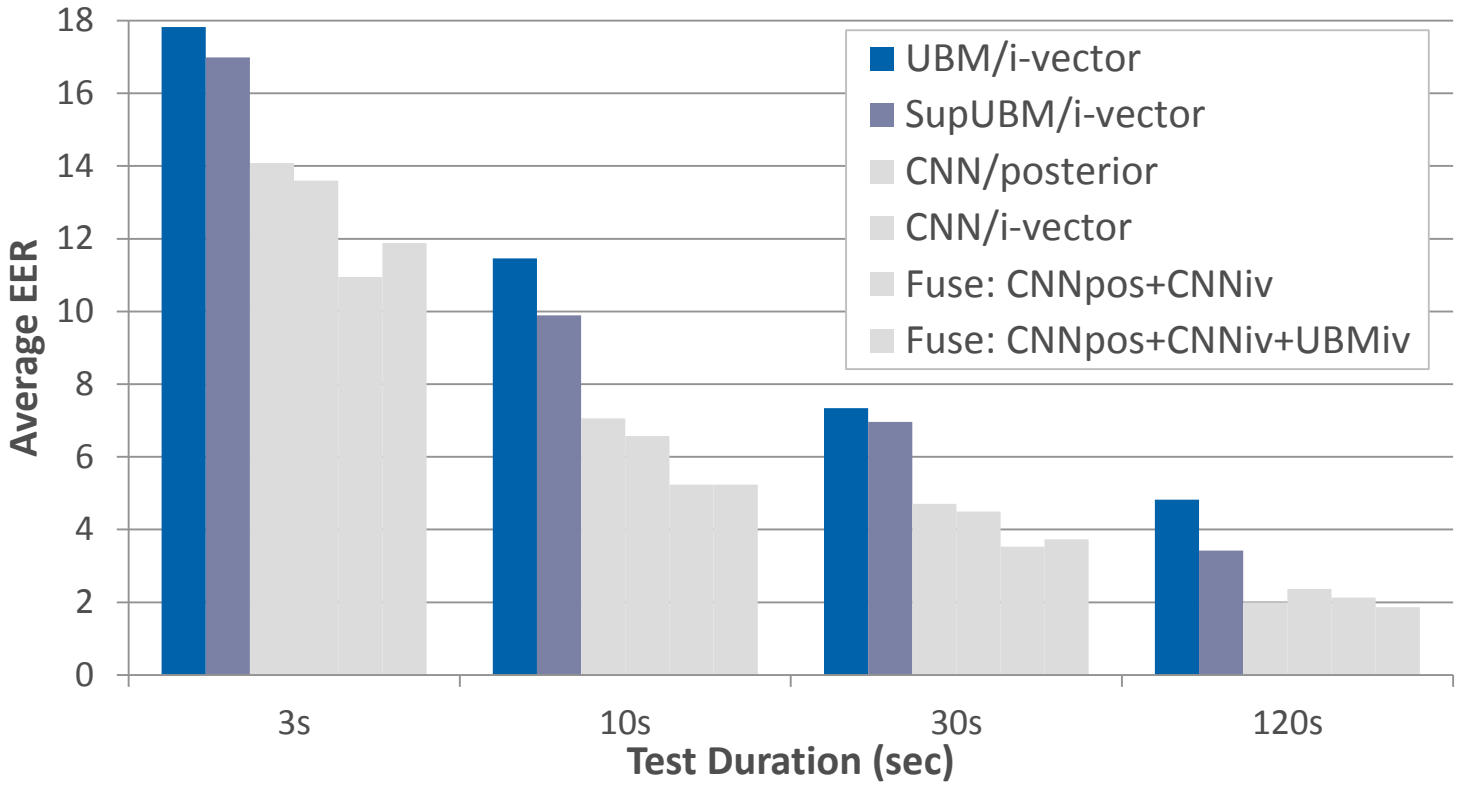
CNN/posterior vectors

- Collect 3353-dimensional average posteriors of the utterance
- Reduce via probabilistic PCA to 400D (same as i-vector dimension)

LID Backend

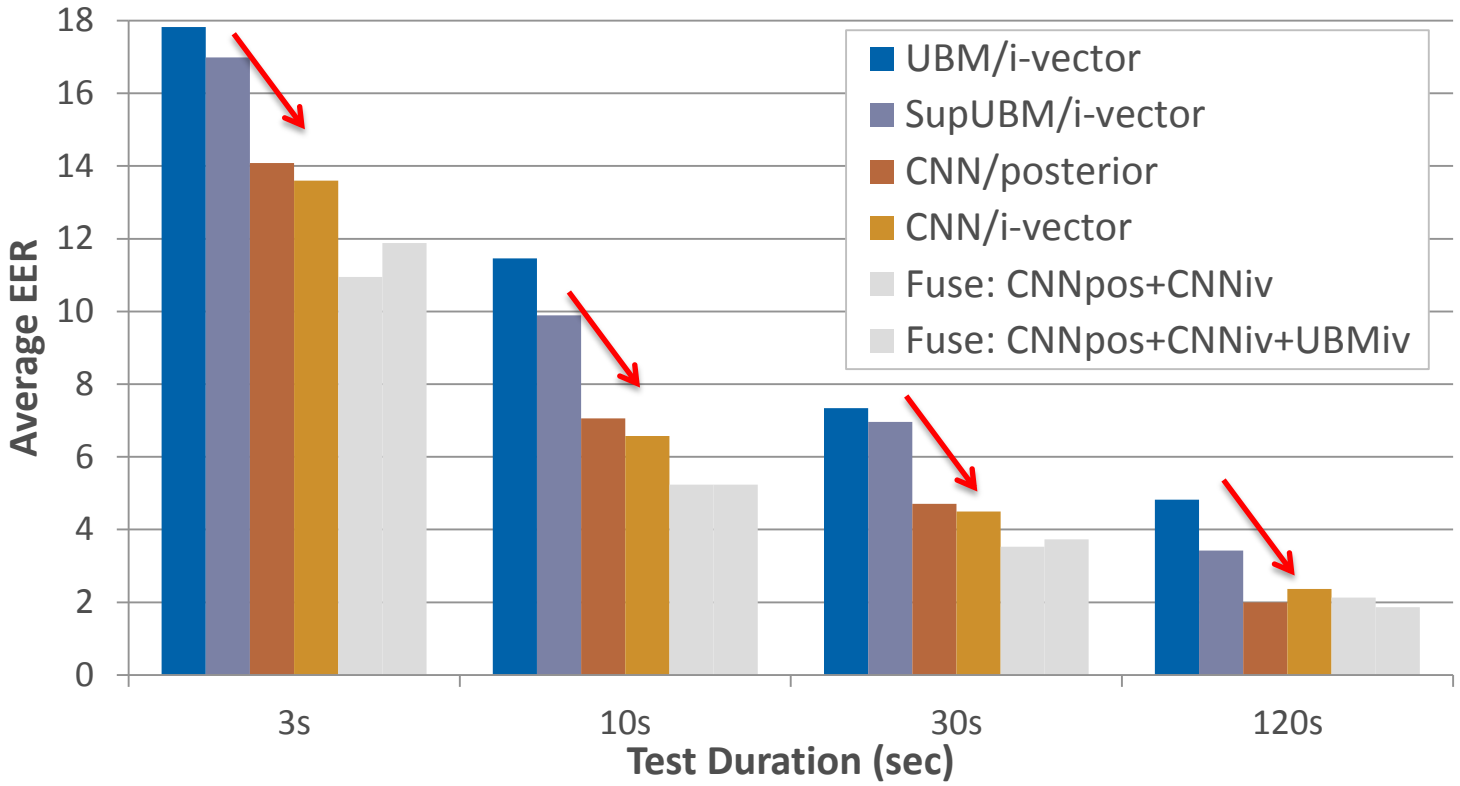
- Simple MLP trained on cross entropy (200-node hidden layer)
- Training data duplicated as 30s and 8s chunks with 50% overlap
  - Improved all systems on all durations
- Input: 400D vectors
- Output: 5 target languages + 1 out-of-set

Performance: UBM approaches



- **SupUBM**: A UBM with 3353 components each trained on the frames aligned to each senone of the CNN.
  - Provides a fair comparison to the CNN systems

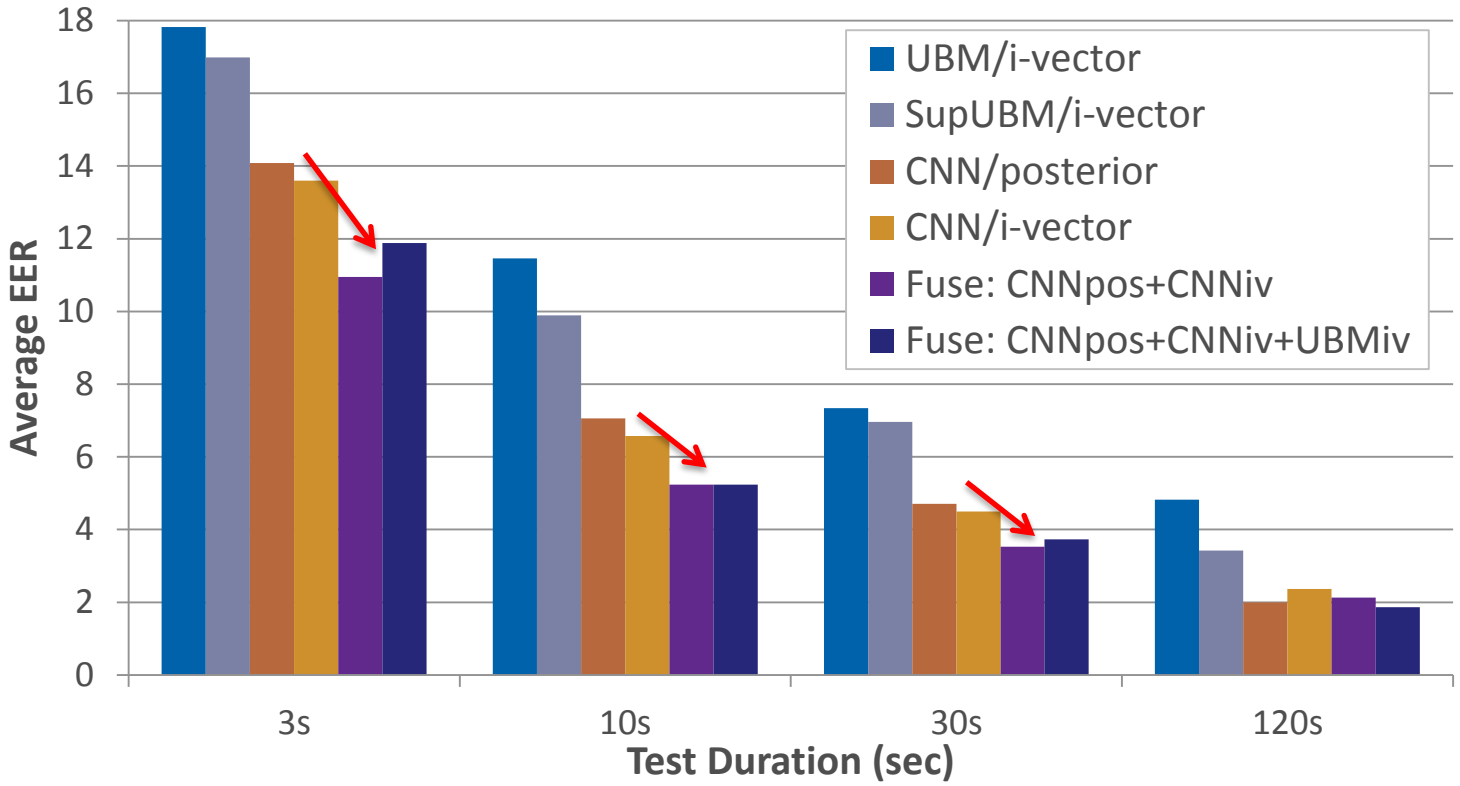
Performance: CNN approaches



CNN/i-vector greater than **30% relative improvement** over SupUBM for 10s+ tests

- CNN approaches: 20% relative improvement at 3sec
- Comparable performance from simple CNN/posterior system

Performance: Fusion



20% relative improvement fusing
CNN/posterior and CNN/i-vector for tests less than 120s

- No additional gain when adding UBM/i-vector system to fusion (except in 120s case)

Conclusions

- Confirmed the robustness of the CNN on noisy conditions
- Showed the **CNN/i-vector** framework to be effective for RATS LID
- Proposed a new kind of phonotactic system, **CNN/posterior**, that competes with acoustic system
- High complementarity between CNN/posterior and CNN/i-vector.

The Extension

- The CNN/posterior system can be improved by removing the PCA dimension reduction and fusion of language dependent models.
- Deep bottleneck feature are good alternatives of the direct usage of the DNN/CNN output.