Bottleneck Features for Speaker Recognition

Odyssey 2012:
The Speaker and Language Recognition Workshop

Sibel Yaman[1], Jason Pelecanos[1], and Ruhi Sarikaya[2]

[1] IBM T. J. Watson Research Labs, Yorktown Heights, NY

[2] Microsoft Corporation, Redmond, WA

Roadmap

- **Introduction**
- **Bottleneck feature extraction**
  1) A conversation level training criterion
  2) Incorporating a separate system in training
- **Experiments**
- **Summary**

The Big Picture

- **In the speech recognition literature:**

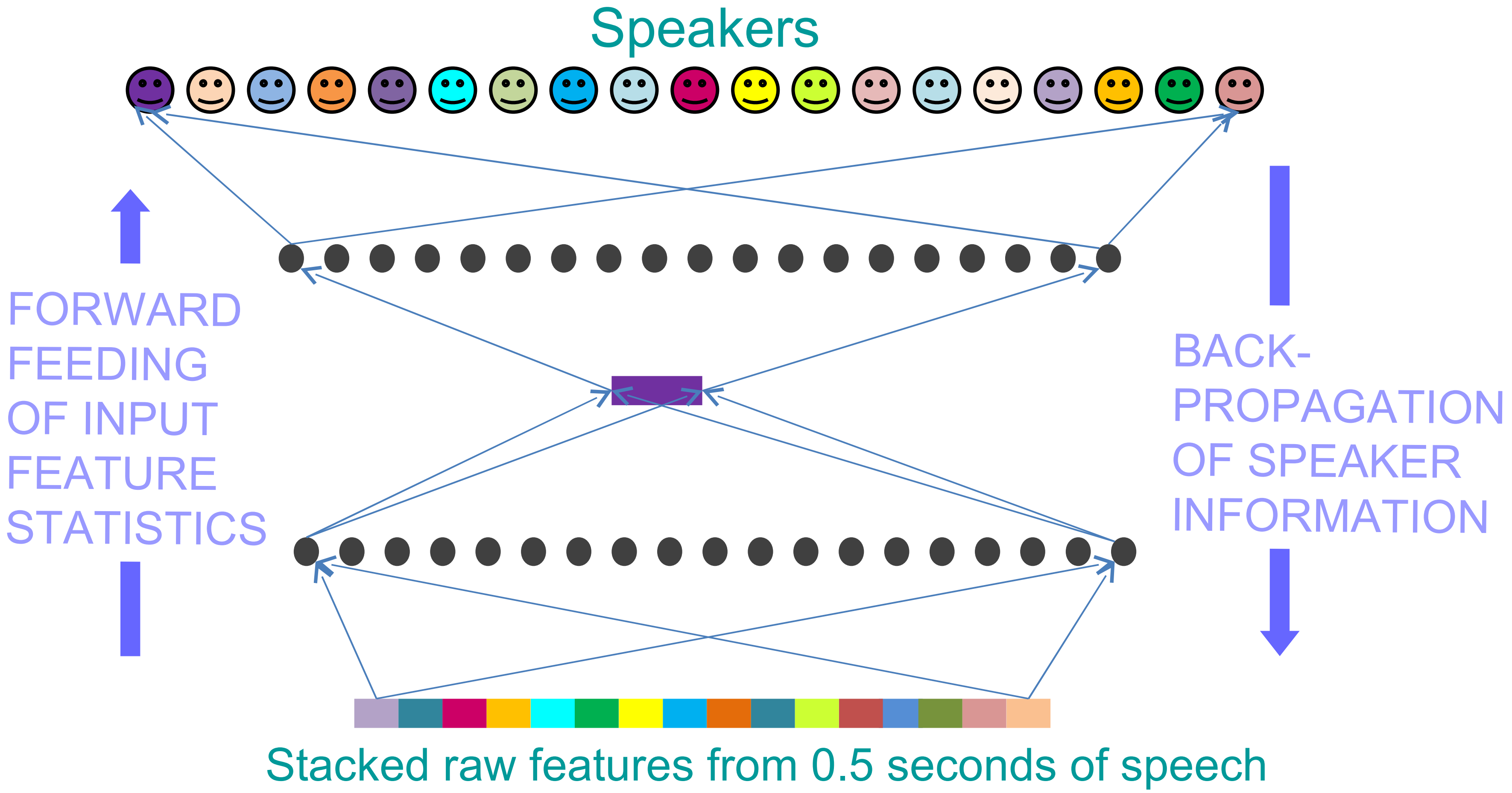
Deep networks are shown to outperform HMMs (Seide 2012, etc.).

- **In the speaker recognition literature:**

Many sites report ever-improving performance figures (Konig 1998, Garimella 2012).

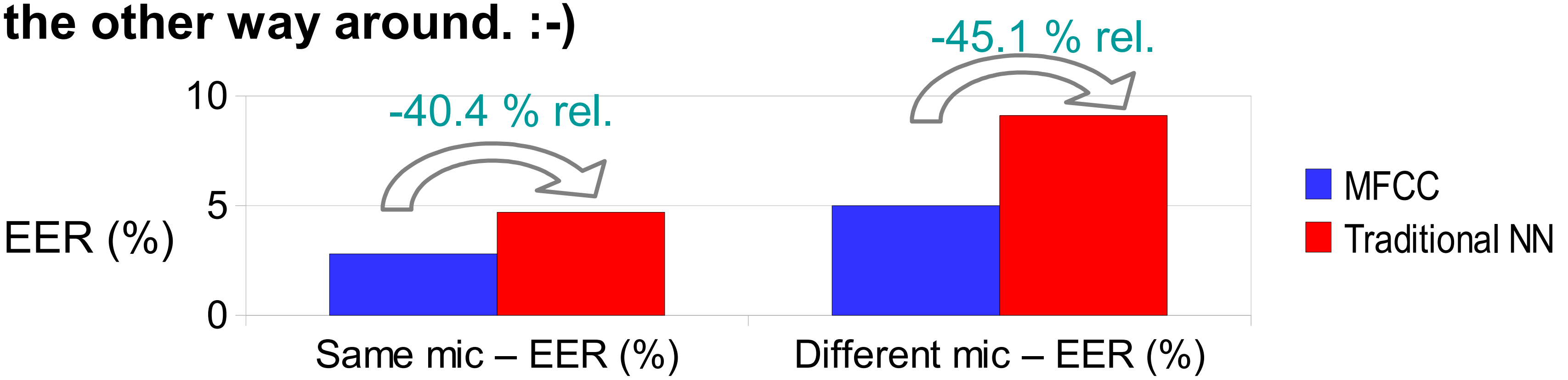
Bottleneck Network Architecture

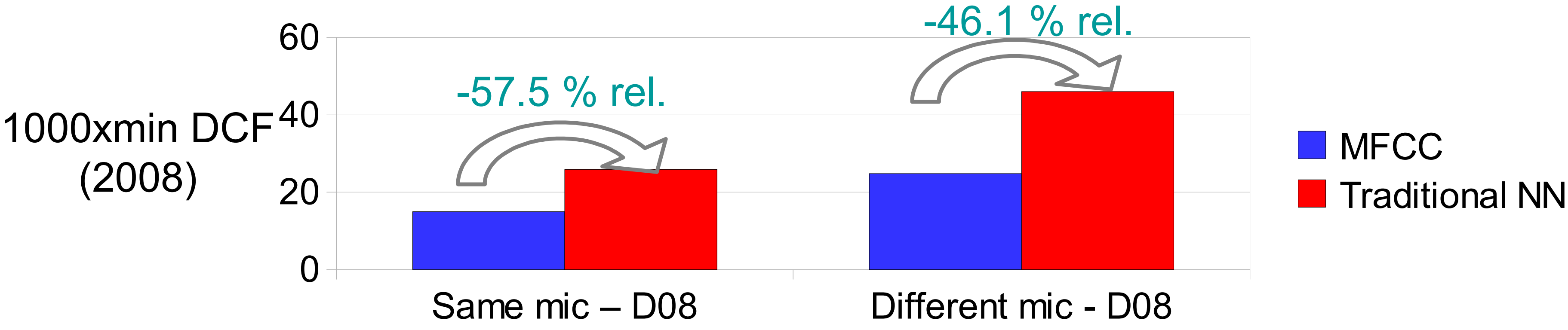
An *information bottleneck* acts as a feature compressor (Konig 1998).


Speakers
FORWARD FEEDING OF INPUT FEATURE STATISTICS
BACK-PROPAGATION OF SPEAKER INFORMATION
Stacked raw features from 0.5 seconds of speech

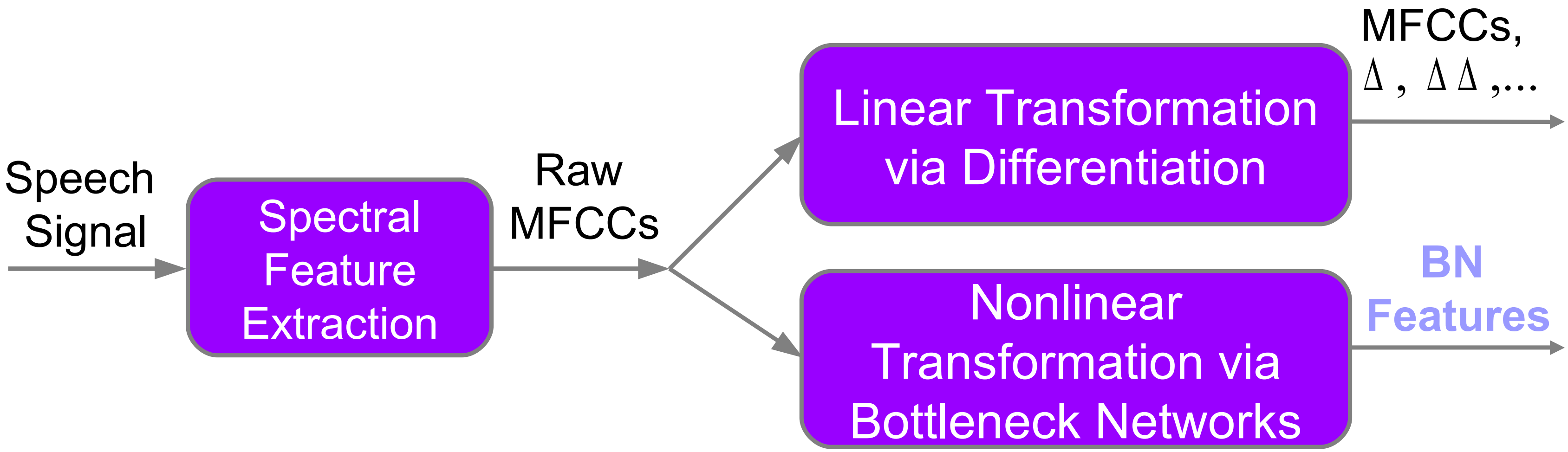
Using Neural Networks for Speaker Recognition

- **Feature extraction with neural networks traditionally performs relatively poorly.**
- **We investigate approaches to make the performance comparison the other way around. :-)**





An Overview


Speech Signal
Spectral Feature Extraction
Raw MFCCs
Linear Transformation via Differentiation
MFCCs, Δ , Δ Δ ,...
Nonlinear Transformation via Bottleneck Networks
BN Features

We demonstrate two ways of exploiting the expressive power of deep networks:

1) The training is adjusted to the targeted performance evaluation metric.

2) Information from a separate system is incorporated in training.

Roadmap

- Introduction
- **Bottleneck feature extraction**
  1) A conversation level training criterion
  2) Incorporating a separate system in training
- Experiments
- Summary

Frame vs. Conversation Level Training

- **Frame level training has limitations:**
  - Learning the speaker is constrained to the context around the current frame.
  - A long context would explode the number of free parameters.

- **Conversation level training offers solutions:**
  - The frames coming from one conversation are tied together so that a single decision is made.
  - The network size can be kept relatively small.

(1) A Speaker Recognition Training Criterion

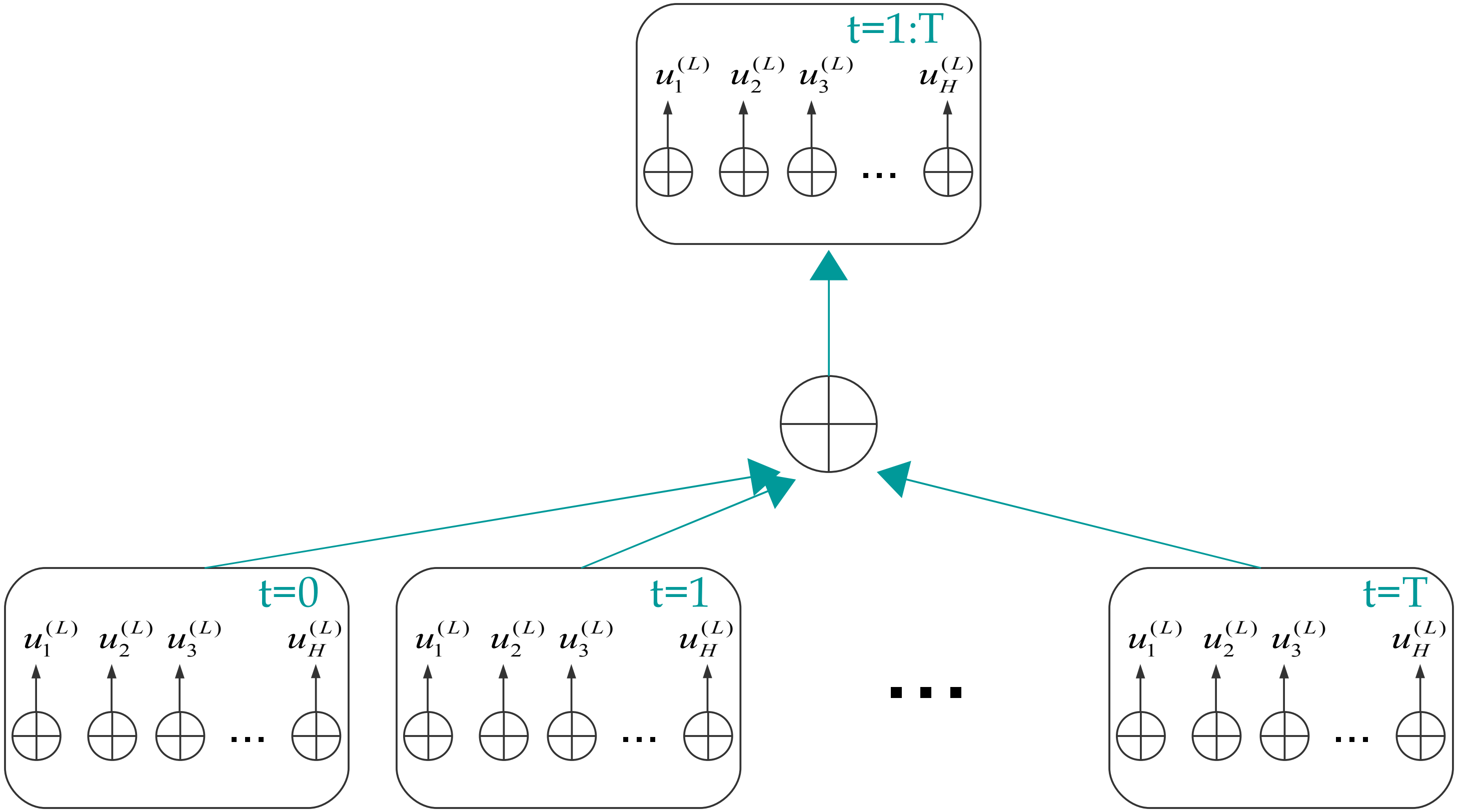
- **A log-likelihood ratio-based training criterion (Brummer 2005) is optimized**

$$J_{LLR}(\Theta) = \alpha \sum_{T:\text{target}} \underbrace{\log(1+e^{-u_T-c})}_{\text{Cost associated with target trials}} + \beta \sum_{N:\text{nontarget}} \underbrace{\log(1+e^{+u_N+c})}_{\text{Cost associated with nontarget trials}}$$

- **There is one target and (*S-1*) nontarget scores at the output layer.**

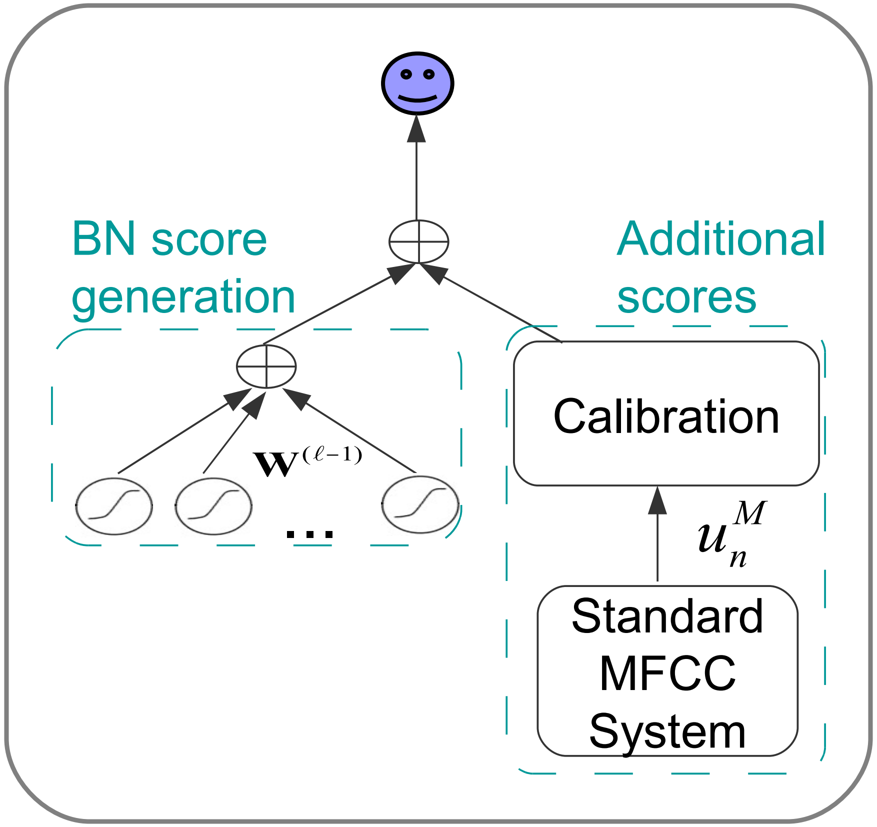
Conversation Level Training

- We need a global constraint on the decision for the entire recording.
- The scores are averaged at the output layer before the nonlinearity.



(2) Using a Separate System in Training

Scores from a separate system are incorporated in training.



The term

$$\mathbf{u}^{(\ell)}(\Theta) = \mathbf{W}^{(\ell-1)}\sigma^{(\ell-1)}$$

in the training objective is replaced with

$$u'_n(\Theta) = \omega_1 \mathbf{W}^{(\ell-1)}\sigma^{(\ell-1)} + \omega_2 u_n^M + \kappa$$

Score Calibration

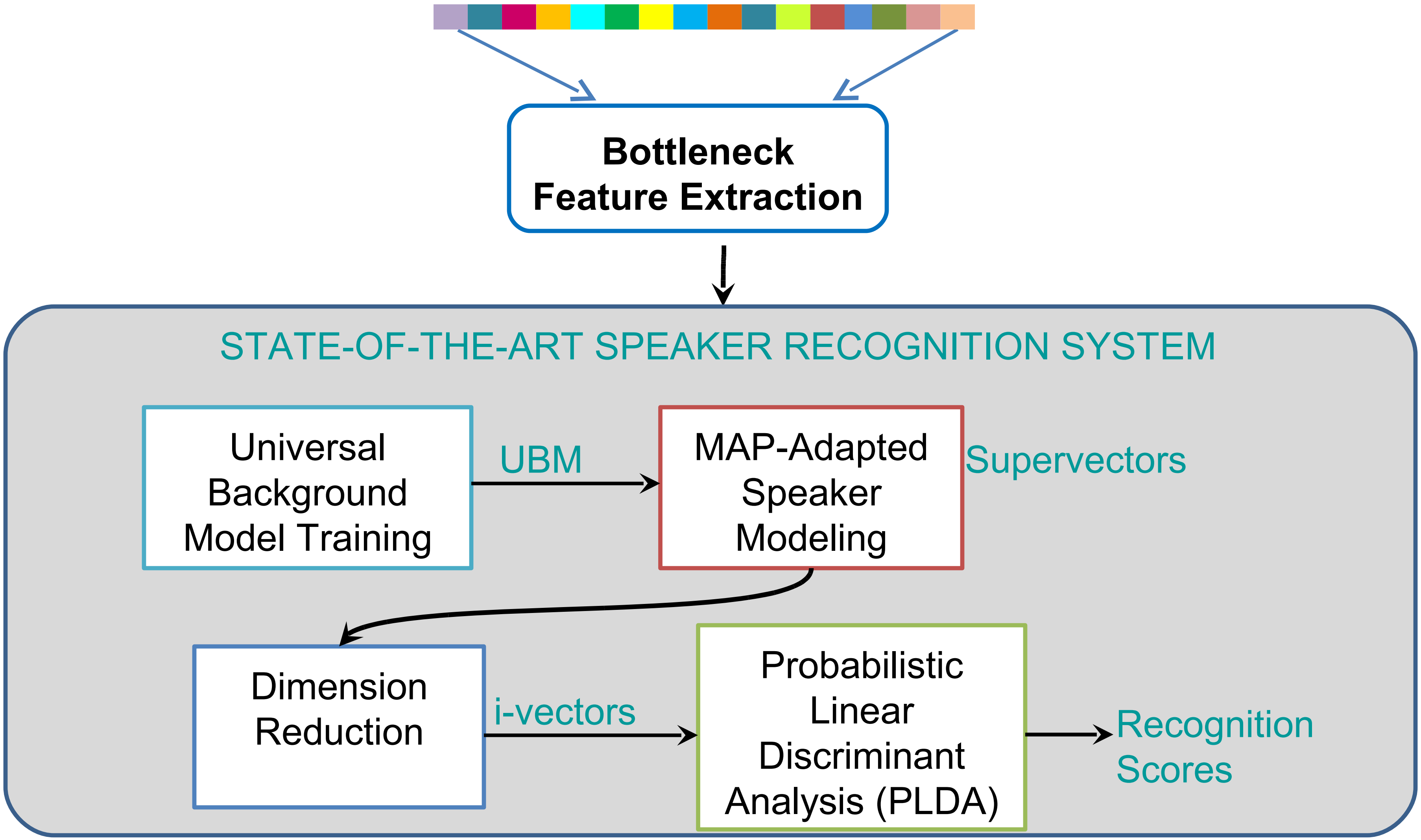
- **The additional scores should have a log-likelihood ratio interpretation.**
- **The score calibration is achieved by solving**

$$\{\omega_1^*, \omega_2^*, \kappa^*\} = \arg\min_{\omega_1, \omega_2, \kappa} J_{LLR}(\omega_1, \omega_2, \kappa \mid \Theta \;\; \text{fixed})$$

- **The network is trained by solving**

$$\Theta^* = \arg\min_{\Theta} J_{LLR}(\Theta \mid \omega_1^*, \omega_2^*, \kappa \;\; \text{fixed})$$

The Back-End System


Bottleneck
Feature Extraction
STATE-OF-THE-ART SPEAKER RECOGNITION SYSTEM
Universal Background Model Training
UBM
MAP-Adapted Speaker Modeling
Supervectors
Dimension Reduction
i-vectors
Probabilistic Linear Discriminant Analysis (PLDA)
Recognition Scores

Roadmap

- **Introduction**
- **Bottleneck feature extraction**
  1) A conversation level training criterion
  2) Incorporating a separate system in training
- **Experiments**
- **Summary**

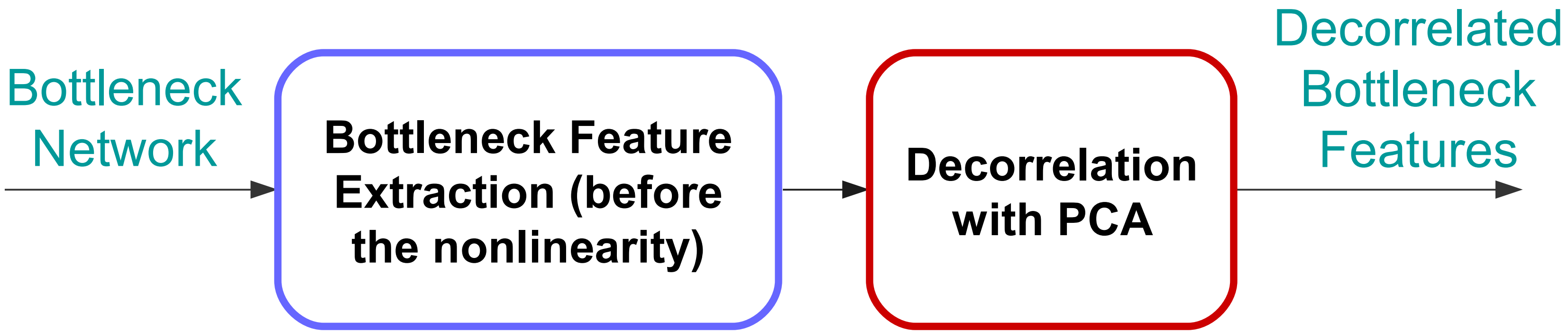
Experiments

- **We ran experiments on the same and different microphone tasks of NIST SRE 2010.**
- **Microphone recordings were used in bottleneck network training.**
  - 173 speakers in the training and validation sets
  - 4341 recordings in training and 865 recordings in validation
- **Network architecture:**

294 dimensional input → 1000 x 42 x 500 → 173 speakers

Processing of the Input and Output Features of the Network

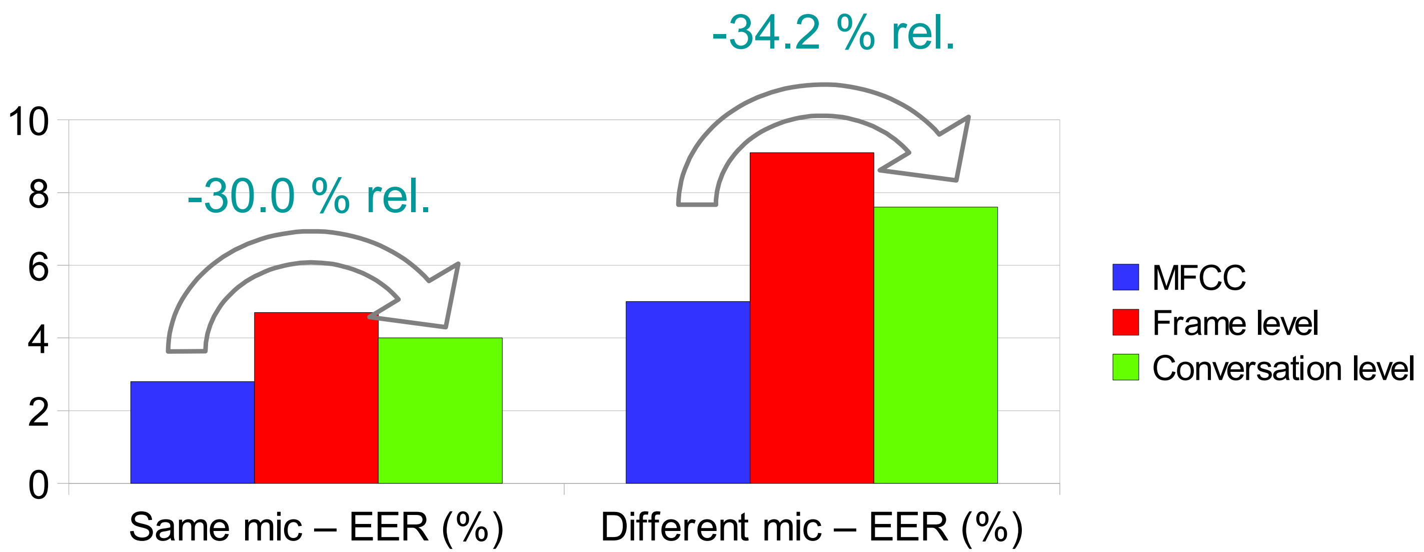
- **Input features are mean and variance normalized to better condition the network.**


Bottleneck Network
Bottleneck Feature Extraction (before the nonlinearity)
Decorrelation with PCA
Decorrelated Bottleneck Features

- **The bottleneck features are decorrelated for modeling with diagonal covariance GMMs.**

Effect of the Training Criterion

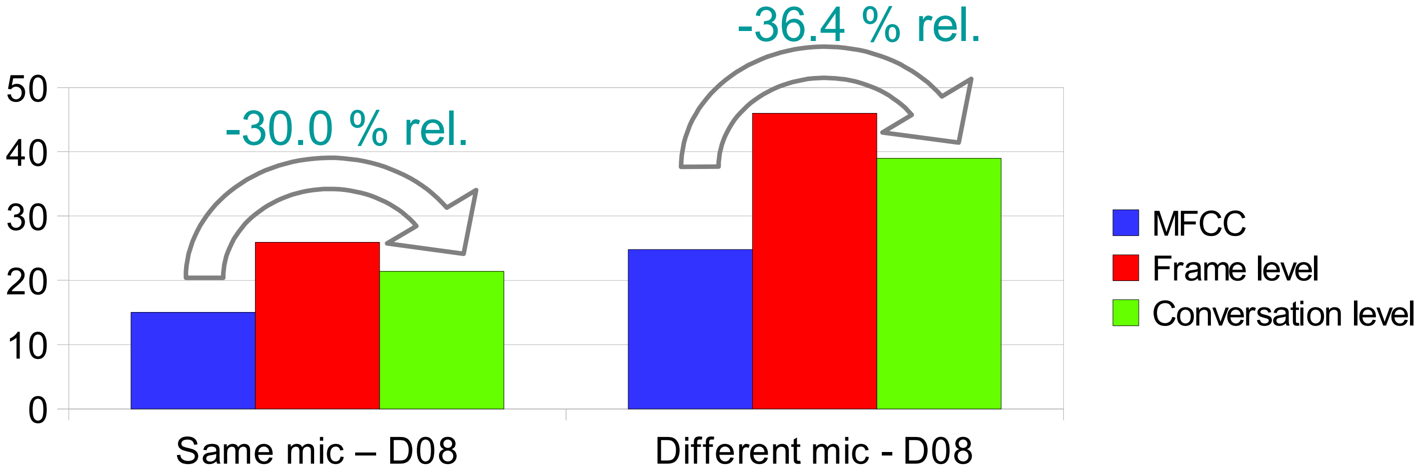
EER (%)



-30.0 % rel.

-34.2 % rel.

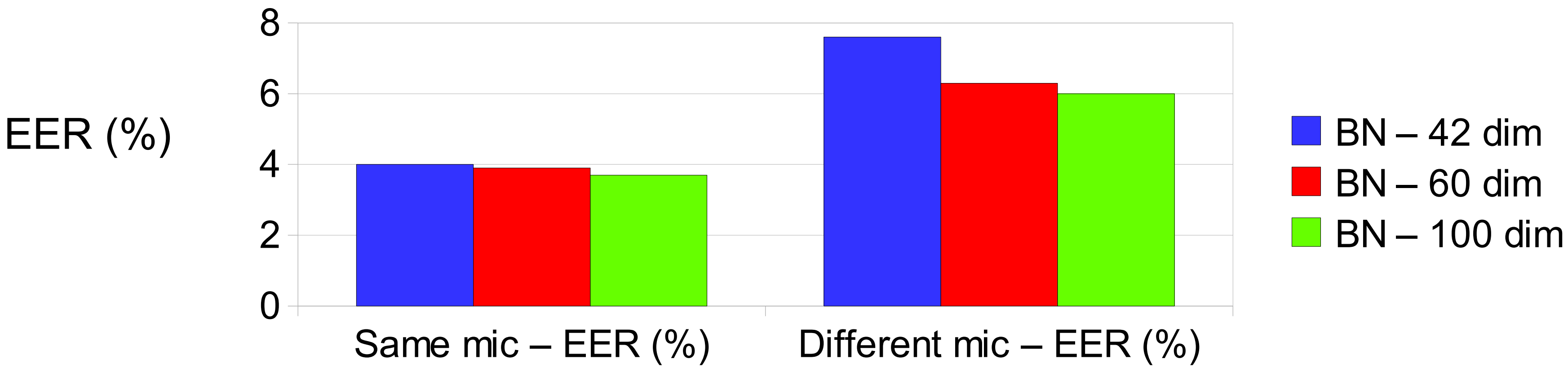
1000xmin DCF (2008)

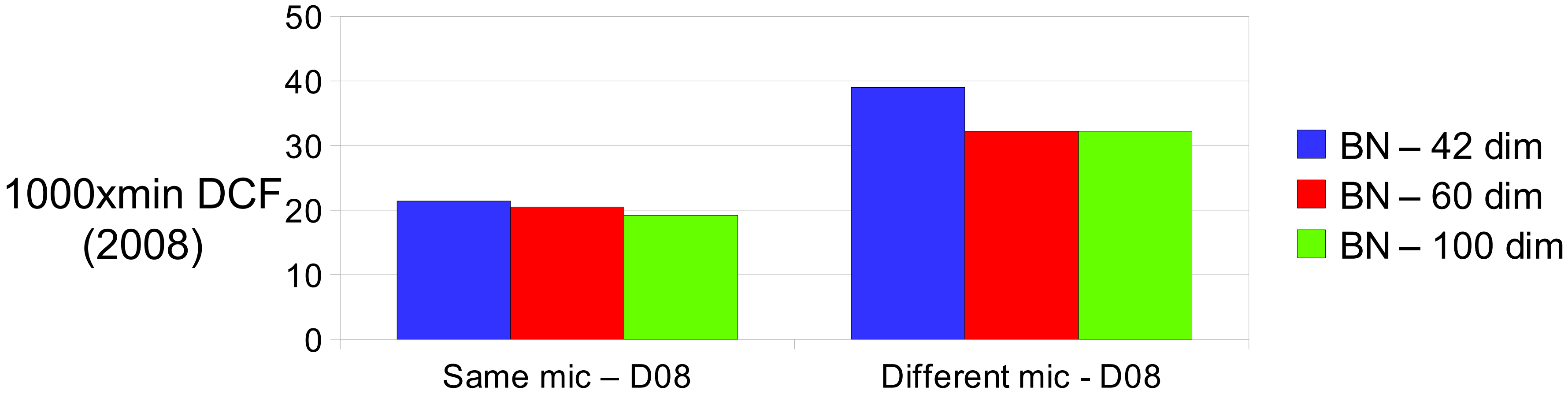


-30.0 % rel.

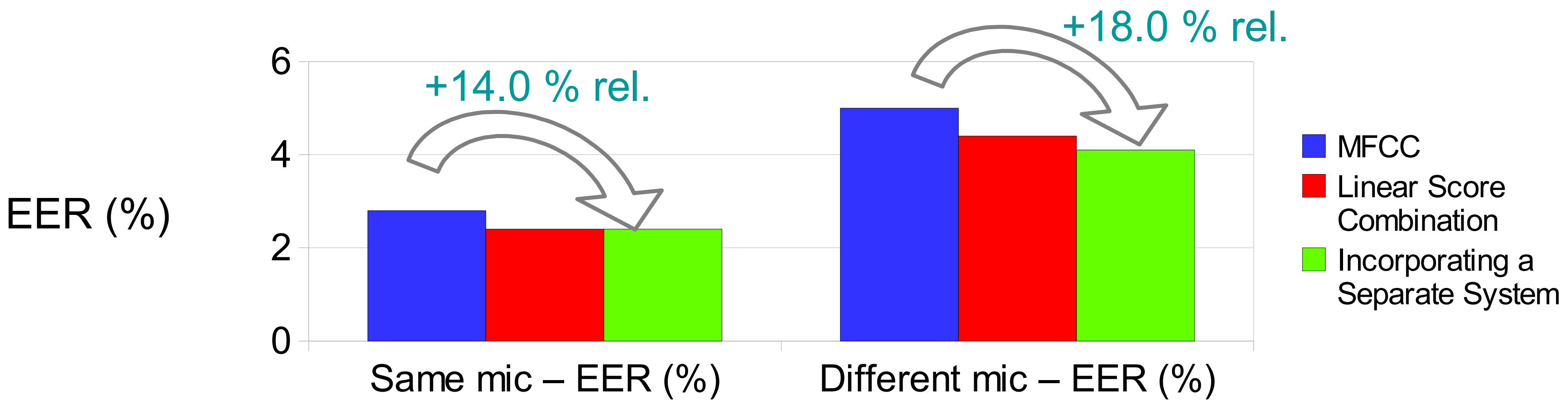
-36.4 % rel.

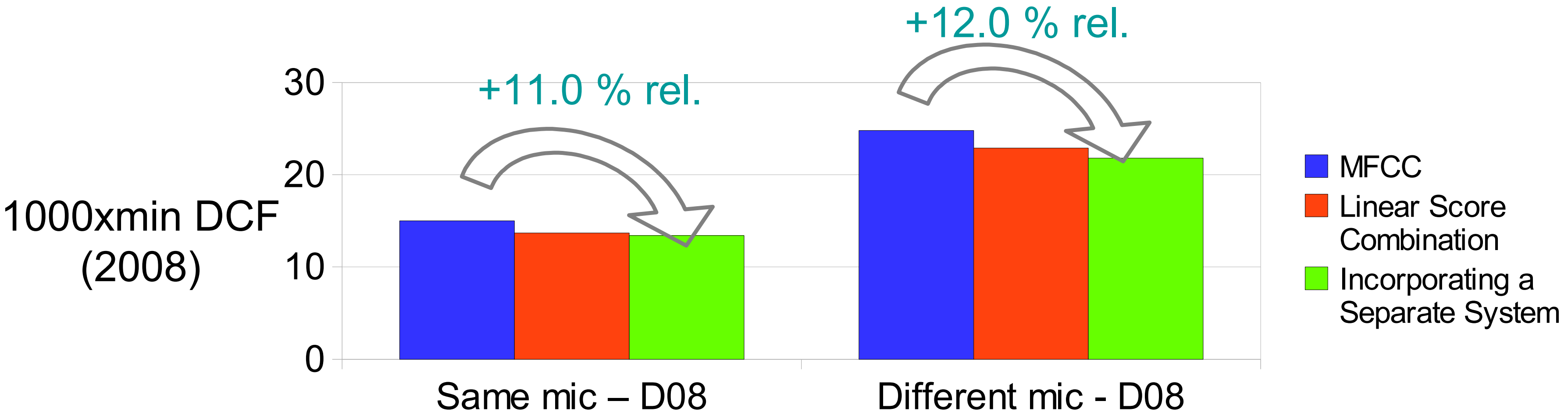
Dependence on Feature Size





Performance when Trained with Information from a Separate System





Summary

1) **We showed how to train a neural network for use in the front-end of a speaker recognition system.**

- A conversation level training criterion that minimizes a log-likelihood ratio score-based cost function is developed.

2) **We also showed how to use neural networks to exploit information from a separate system.**

Thank you!