INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD
OUTREACH DIVISION

talent sprint

# AI & ML in speech systems

Dr. Anil Kumar Vuppala,
Speech Processing Laboratory,
IIIT Hyderabad

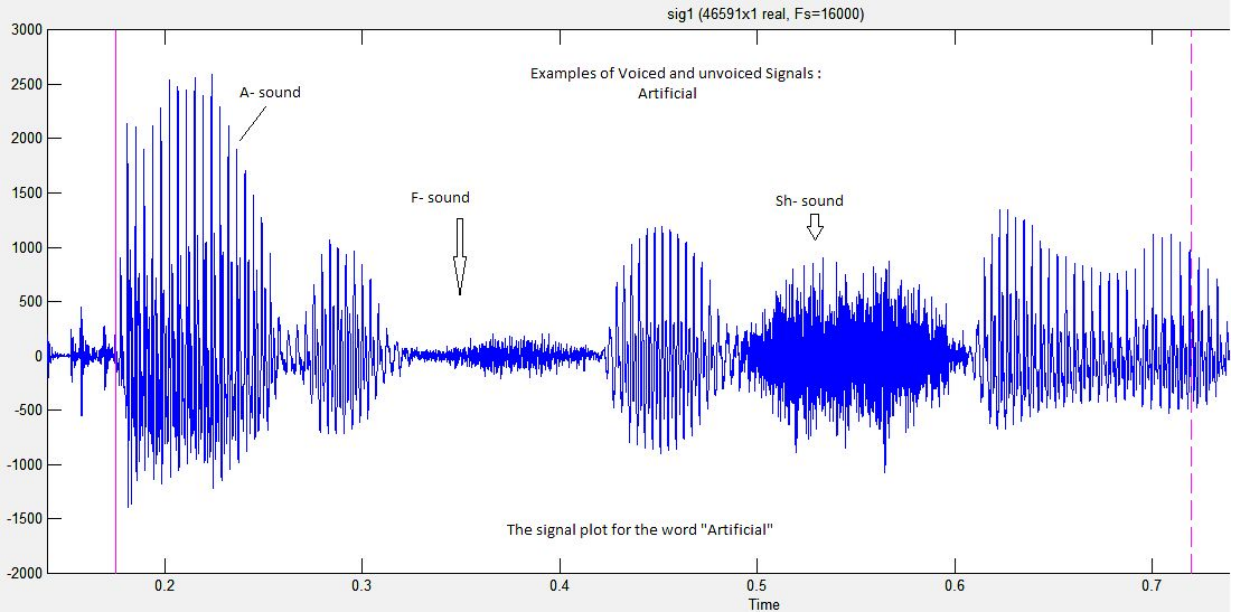
# Contents

- ➔ Introduction
- ➔ Feature representation
- ➔ Gaussian Mixture Modelling (GMM) for Speech technologies
  - ◆ Speaker recognition/verification
  - ◆ Language Identification
  - ◆ Emotion recognition
- ➔ GMM-Hidden Markov Modelling for sequential information modelling
  - ◆ Automatic speech recognition (ASR)
- ➔ DNN based speech applications

# Introduction contd.

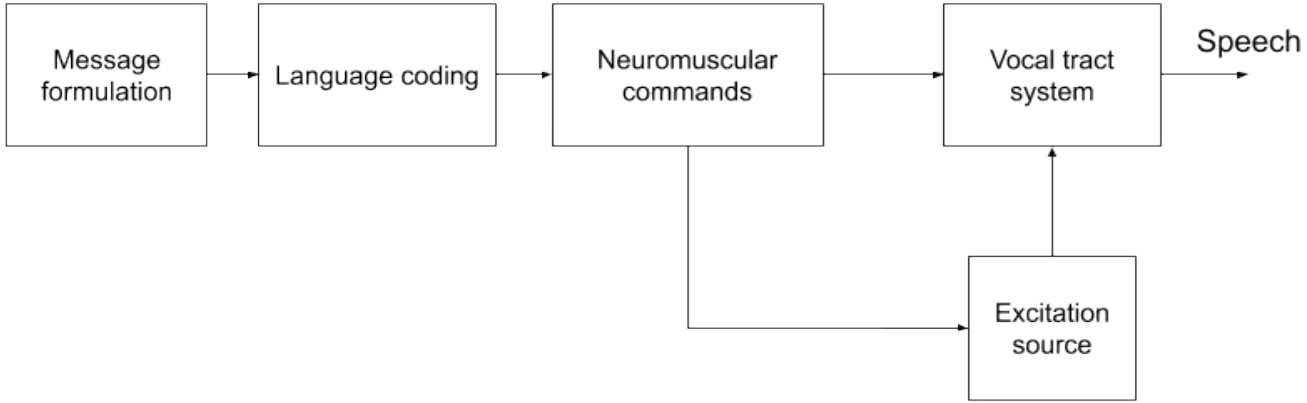
**Information in speech**

- Message
- Language
- Gender
- Age
- Speaker identity
- Emotional state
- Cognitive behaviours of
  - Depression
  - Autism
  - Language dis
- Abnormalities in speech production etc.




Typical speech signal for a word “Artificial”

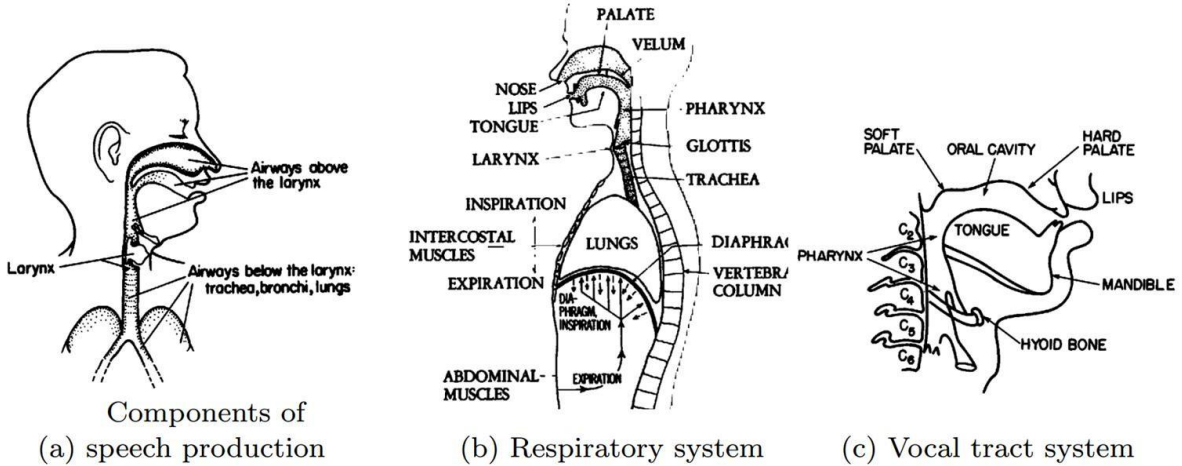
# Introduction




**Speech** is a unique, complex, and dynamic motor activity through which we express our thoughts and emotions.

Natural mode of communication for human beings.

Important in human computer interaction




Components of
(a) speech production (b) Respiratory system (c) Vocal tract system

# Introduction contd.

**Applications of speech processing:**

- Automatic speech recognition
- Text-to-speech synthesis
- Spoken language dialog systems
- Speaker recognition/verification
- Emotion recognition
- Language identification
- Pronunciation evaluation for clinical tools
- Forensic tools
- Speech coding

# Introduction contd.

**Applications of speech processing:**

- Automatic speech recognition
- Text-to-speech synthesis
- Spoken language dialog systems
- Speaker recognition/verification
- Emotion recognition
- Language identification
- Pronunciation evaluation for clinical tools
- Forensic tools
- Speech coding

**Importance of speech technologies:**

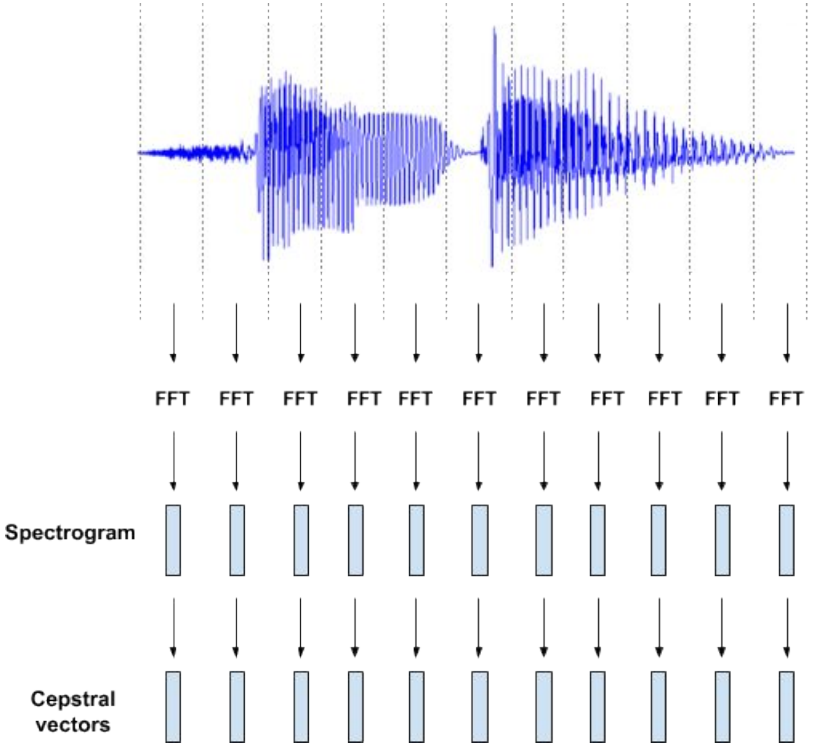
- For better human machine interaction
- For information security and authentication
- Speech based smart appliances
- Multilingual speech systems (speech to speech translation)
- As a navigator for disabled people
- As a personalized speech therapist
- As a personalized assistant in education, agriculture, health care, and other service sectors.

# Feature extraction

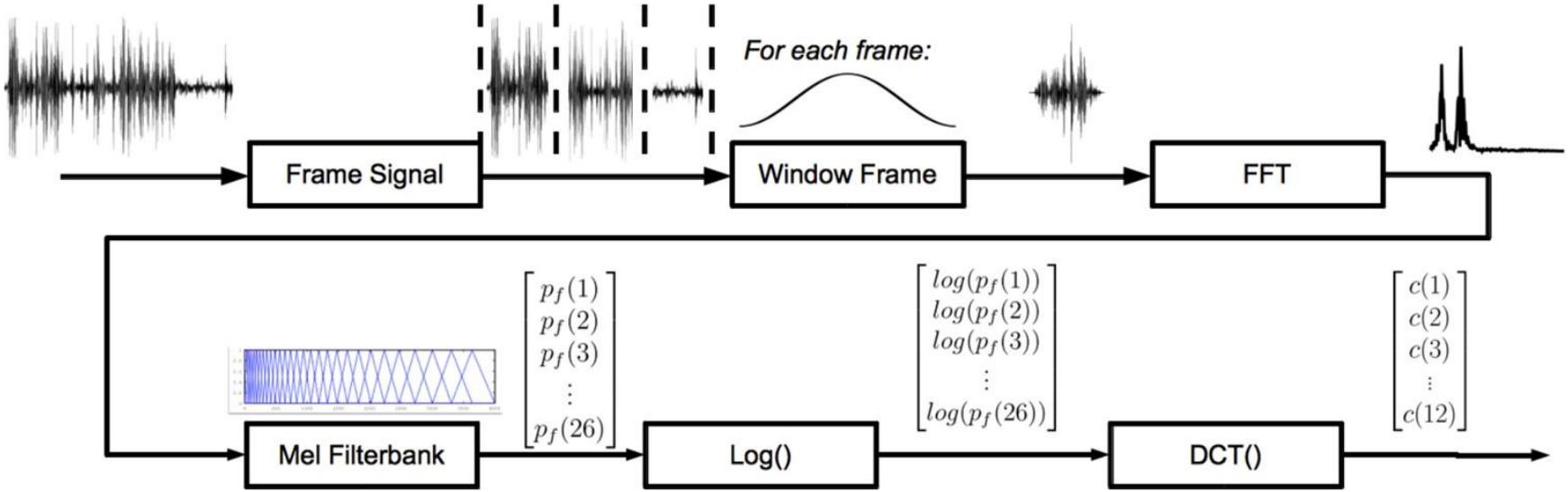
- Primary step in the development of speech systems
- Fourier transform
- Short time Fourier analysis due to non-stationarity
- 20-30 ms framesize of speech
- Window type: Hamming, Hanning, Rectangular etc.
- Cepstral coefficients are most common feature representation the which captures the vocal tract shape and dynamic information
- Mel-frequency cepstral coefficients (MFCC), Linear prediction prediction cepstral coefficients (LPCC), Perceptual linear prediction cepstral coefficients (PLPCC), etc.

# Feature extraction

- Primary step in the development of speech systems
  Fourier transform
- Short time Fourier analysis due to non-stationarity
- 20-30 ms framesize of speech
- Window type: Hamming, Hanning, Rectangular etc.
- Cepstral coefficients are most common feature representation the which captures the vocal tract shape and dynamic information
- Mel-frequency cepstral coefficients (MFCC), Linear prediction prediction cepstral coefficients (LPCC), Perceptual linear prediction cepstral coefficients (PLPCC), etc.



# Feature extraction contd.




- Including the energy, total 13-dimensional static cepstral features
- 39-dimensional MFCC features: [static -- delta -- delta-delta]

# Speech systems

**Speaker recognition:** is a technique to recognize the identity of a speaker from a speech utterance.

**Speaker verification** aims to verify whether an input speech corresponds to the claimed identity.

**Emotion recognition** is a process of identifying the emotional state (Anger, Happy, Sad, Neutral, Fear, etc) of a speaker from the spoken utterance.

**Spoken Language identification** is a task of recognising the language identity/information from the speech signal.

**Automatic speech recognition** is the process of deriving the transcription (word sequence) of an utterance, given the speech waveform.

**Text to speech synthesis** aims to converts the given text (word sequence) information into speech.

# Speech systems contd.

Speaker identification/verification
Language identification

Emotion recognition

} Simple classification problems

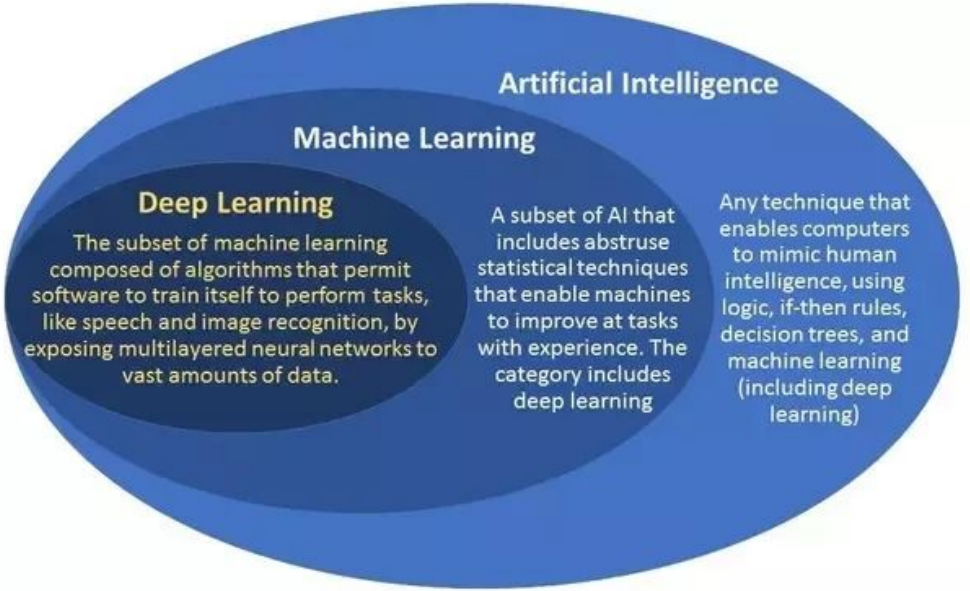
---

Automatic speech recognition (ASR)

Text to speech synthesis

Are advanced speech systems require: classification, duration/pronunciation modelling, sequence-sequence mapping etc.

# Deep learning in AI




Ref:-
https://www.quora.com/What-are-the-main-differences-between-artificial-intelligence-and-machine-learning-Is-machine-learning-a-part-of-artificial-intelligence

# Why Third wave? 1950's, 1990's and now

More data from systems and sensors (IoT)

More compute power : GPU's, multi-core CPU's

Can train deep architectures

**Some more applications** of DL are:

Speech recognition, Image classification, natural languge processing, chat bots, personalized recommendations, prediction, anomaly detection, fraud detection, drug discovery, autonomous cars, video analytics etc...

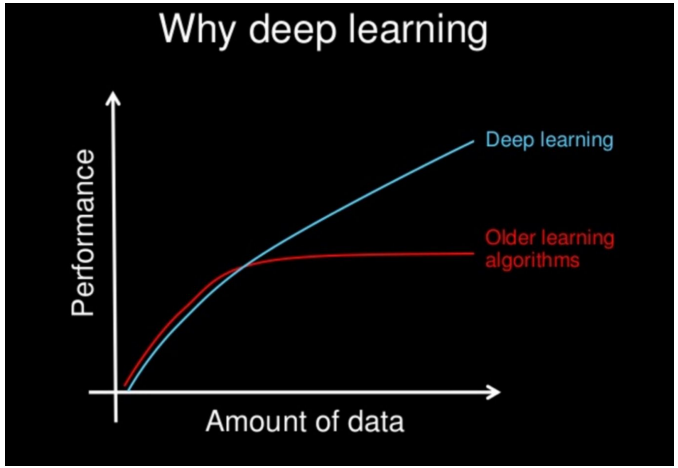
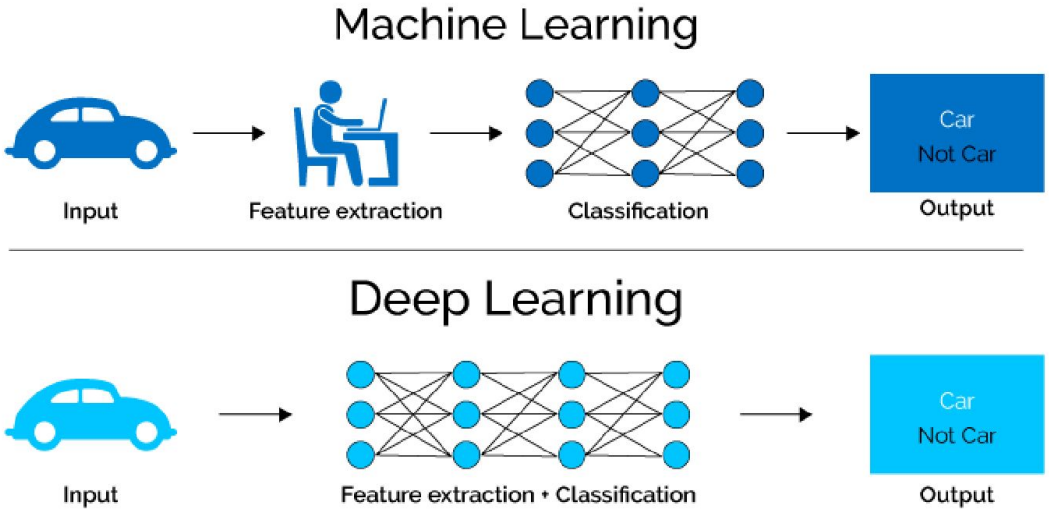



Fig Ref:-
https://www.linkedin.com/pulse/how-artificial-intelligence-revolutionizing-finance-del-toro-barba/

# Machine learning vs Deep learning




Ref:- https://www.xenonstack.com/blog/log-analytics-with-deep-learning-and-machine-learning

# Parameters to vary for tuning

- Number of layers
- Number of neurons in each layer
- Activation function in each layer
- Number of epochs
- Error/loss functions
  - Iteration (equivalent to when a weight update is done)
- Learning rate (α)
  - Size of the step in the direction of the negative gradient
- Batch size
- Momentum parameter (weightage given to earlier steps taken in the process of gradient descent)
- Kernels
- Number of features
- Gradient descent methods

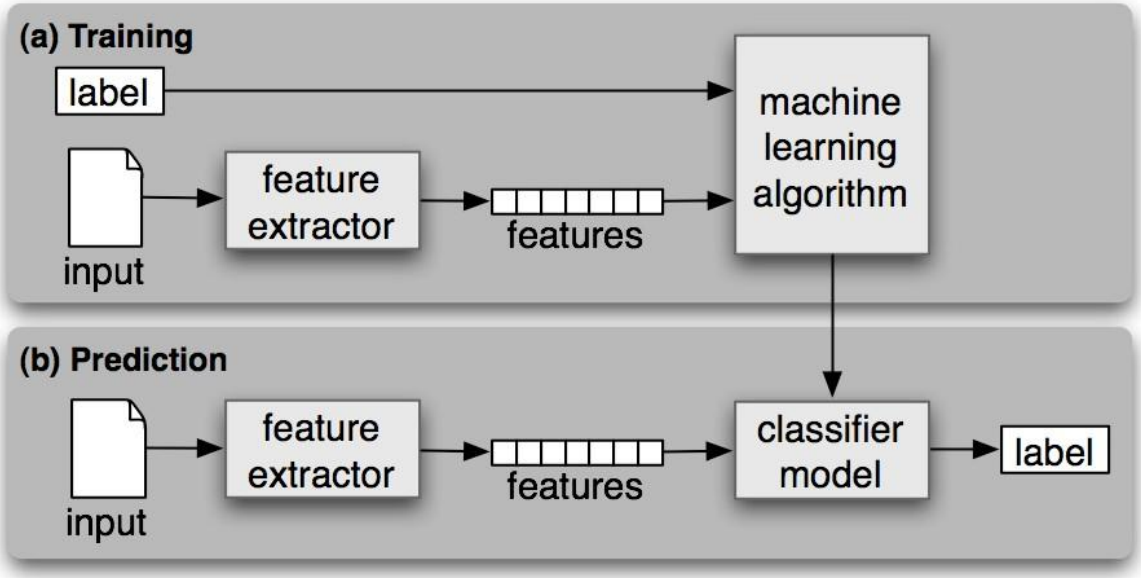
# Speech systems contd.

**Input:** speech signal

**Features:** 39-dimensional MFCCs

**Class labels:**
Language ID (Ex: Telugu-1, Hindi-2, English-3, Tamil-4, etc.) in language identification task. Similarly, speaker ID will be the class label in speaker identification.

**Machine learning algorithms:**
Gaussian Mixture Modelling (GMM), GMM with universal background modelling, I-vector modelling.




Classification system example

# Gaussian Mixture Modelling (GMM)



Parameter estimation:

Assume we have a collection of acoustic frames $X = \{\mathbf{x}_t\}_{t=1}^{T}$ for estimation of model parameters $\boldsymbol{\theta}$

Maximum likelihood (ML) estimation

$$\boldsymbol{\theta}_{\text{ML}} = \arg\max_{\boldsymbol{\theta}} p(X|\boldsymbol{\theta})$$

Maximum *a posteriori* (MAP) estimation

$$\boldsymbol{\theta}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|X) = \arg\max_{\boldsymbol{\theta}} p(X|\boldsymbol{\theta})p(\boldsymbol{\theta})$$

where $p(\boldsymbol{\theta})$ denotes the prior distribution of $\boldsymbol{\theta}$

# Gaussian Mixture Modelling (GMM) contd.



Formulation of GMM:

Gaussian mixture model (GMM) is a weighted sum of Gaussians

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{i=1}^{M} \pi_i b_i(\mathbf{x})$$

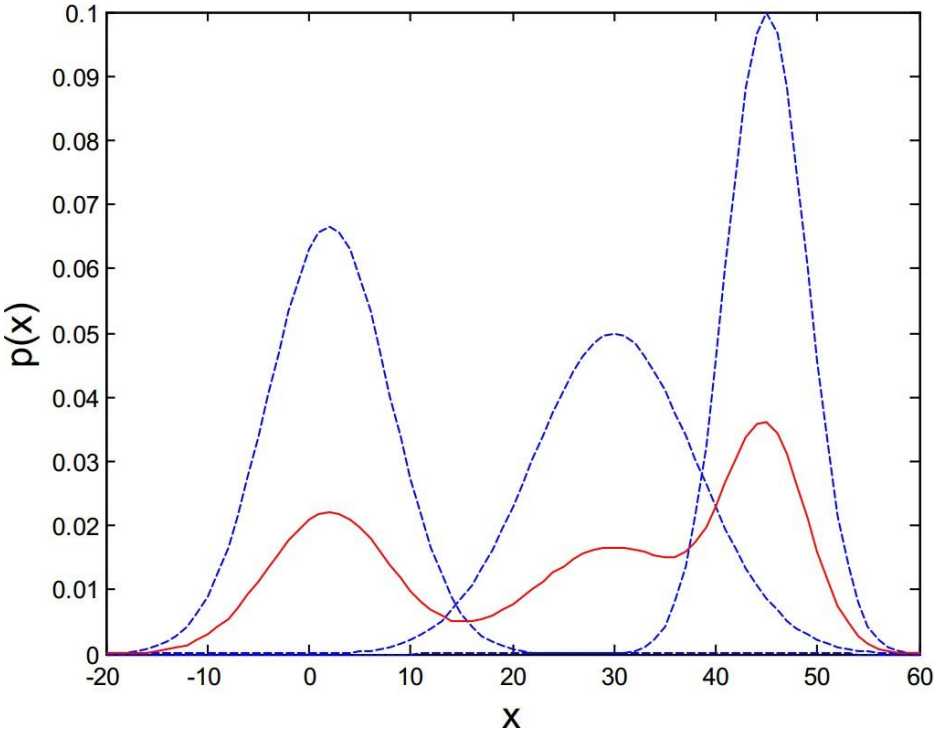
$$\boldsymbol{\theta} = \{\pi_i, \mathbf{u}_i, \Sigma_i\}$$

$\pi_i$: mixture weight
$\mathbf{u}_i$: mixture mean vector
$\Sigma_i$: mixture covariance matrix

$$b_i(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{u}_i)^\top \Sigma_i^{-1}(\mathbf{x}-\mathbf{u}_i)\right)$$

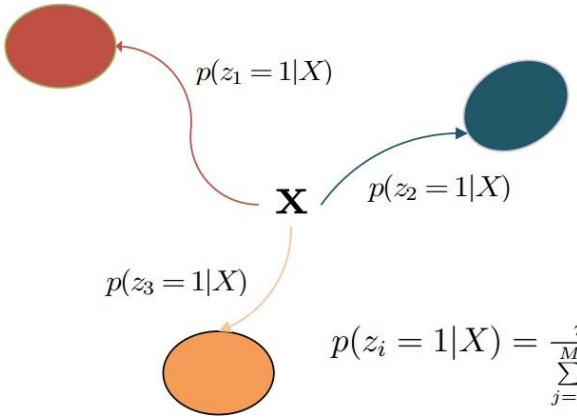
Mixture component $z_i$ is a latent variable which is either zero or one

# Gaussian Mixture Modelling (GMM) contd.




GMM distribution from three Gaussians

# Gaussian Mixture Modelling (GMM) contd.




$$p(z_i = 1|X) = \frac{\pi_i b_i(\mathbf{x})}{\sum_{j=1}^{M} \pi_j b_j(\mathbf{x})}$$

Accumulate sufficient statistics

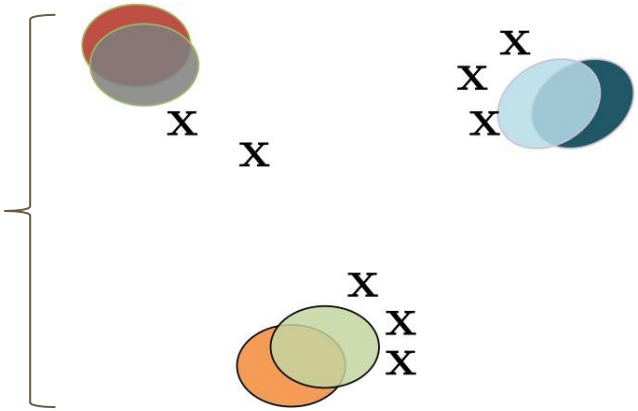
$$T_i = \sum_{t=1}^{T} p(z_{ti} = 1|X)$$

$$\mathbb{E}_i(\mathbf{x}) = \sum_{t=1}^{T} p(z_{ti} = 1|X)\mathbf{x}_t$$

$$\mathbb{E}_i(\mathbf{x}\mathbf{x}^T) = \sum_{t=1}^{T} p(z_{ti} = 1|X)\mathbf{x}_t\mathbf{x}_t^T$$

E-step

M-step




GMM parameters

$$\pi_i^{\text{new}} = \frac{T_i}{T}$$

$$\boldsymbol{\mu}_i^{\text{new}} = \frac{1}{T_i}\mathbb{E}_i[\mathbf{x}]$$

$$\Sigma_i^{\text{new}} = \frac{1}{T_i}\mathbb{E}_i[\mathbf{x}\mathbf{x}^T] - \boldsymbol{\mu}_i\boldsymbol{\mu}_i^T$$

# GMM with universal background model (GMM-UBM)



GMM are used for both **target** and **background models**

- Target model of a class is trained using features corresponding to that class
- **Universal background model** is trained using features from many classes

Target model is adapted from universal background model (UBM)

- good with limited target training data

# GMM with universal background model (GMM-UBM)

GMM are used for both **target** and **background models**

- Target model of a class is trained using features corresponding to that class
- **Universal background model** is trained using features from many classes

Target model is adapted from universal background model (UBM)

- good with limited target training data

**Maximum a posteriori (MAP) adaptation:**

- align target training vectors to UBM
- accumulate sufficient statistics
- update target model parameters with smoothing to UBM parameters

# GMM with universal background model (GMM-UBM)

GMM are used for both **target** and **background models**

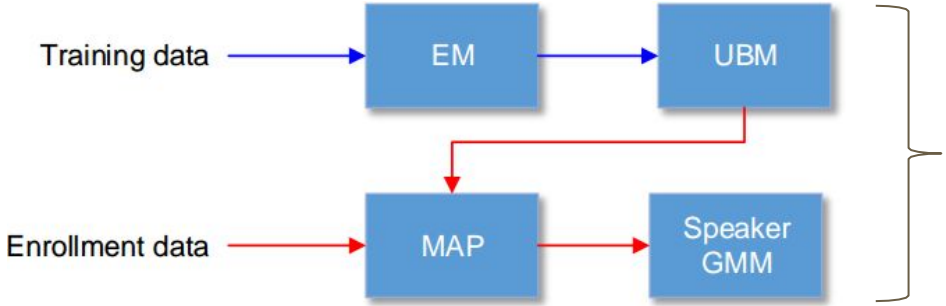
- Target model of a class is trained using features corresponding to that class
- **Universal background model** is trained using features from many classes

Target model is adapted from universal background model (UBM)

- good with limited target training data

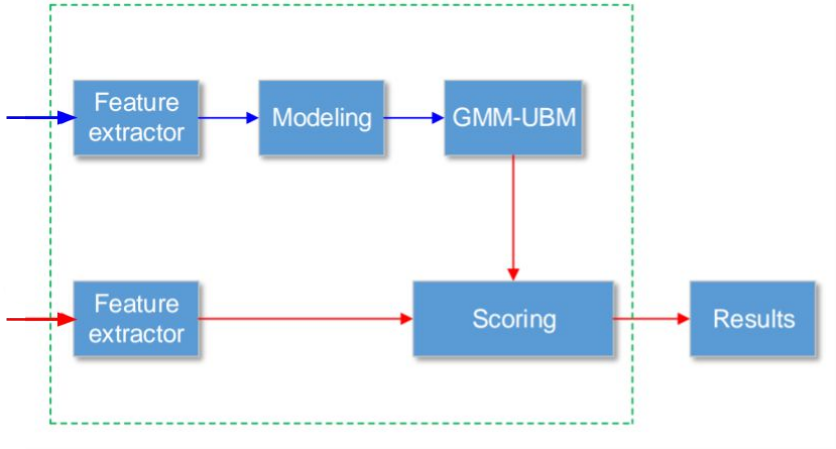
**Maximum a posteriori (MAP) adaptation:**

- align target training vectors to UBM
- accumulate sufficient statistics
- update target model parameters with smoothing to UBM parameters




**GMM-UBM Example:** Adjustment of enrolled speaker's GMM using UBM.

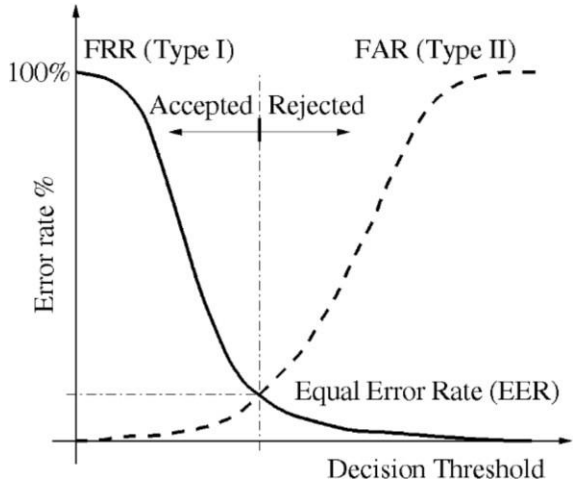
# GMM-UBM based classification

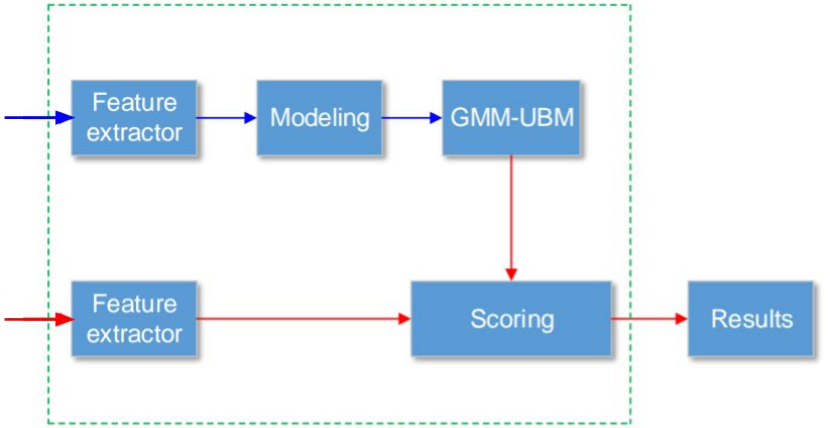



Procedure for **speaker recognition, language identification, emotion recognition** using GMM-UBM modelling

# GMM-UBM based classification

**Metrics:** Accuracy, and Equal error rate (EER). EER is function of false acceptance ratio (FAR) and False rejection ratio (FRR).




Procedure for **speaker recognition, language identification, emotion recognition** using GMM-UBM modelling

# Joint factor analysis and I-vector modelling

- Factor analysis is a statistical method which is used to describe the variability among the observed variables in terms of potentially lower number of unobserved variables called factors.
- Joint factor analysis (JFA) was the initial paradigm for **speaker recognition**
- Later, it is used in Language identification, Emotion recognition, Automatic speech recognition, etc.

# Joint factor analysis and I-vector modelling

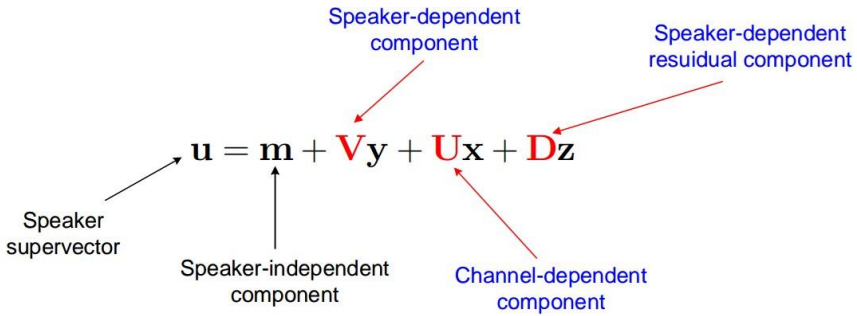
- Factor analysis is a statistical method which is used to describe the variability among the observed variables in terms of potentially lower number of unobserved variables called factors.
- Joint factor analysis (JFA) was the initial paradigm for **speaker recognition**
- Later, it is used in Language identification, Emotion recognition, Automatic speech recognition, etc.

**Intuition and interpretation**

- A supervector for a **speaker (language / emotion)** should be decomposable into speaker independent, speaker dependent, channel dependent, and residual components
- Each component is represented by low-dimensional factors, which operate along the principal dimensions of the corresponding component
- Speaker (language/emotion) dependent component, known as the eigenvoice, and the corresponding factors

# Joint factor analysis and I-vector modelling

GMM supervector u for a speaker (language/emotion/ any other) can be decomposed as:




where

- **m** is a speaker-independent supervector from UBM
- **V** is the eigenvoice matrix
- $\mathbf{y} \sim \mathcal{N}(0, \mathbf{I})$ is the speaker factor vector
- **U** is the eigenchannel matrix
- $\mathbf{x} \sim \mathcal{N}(0, \mathbf{I})$ is the channel factor vector
- **D** is the residual matrix, and is diagonal
- $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ is the speaker-specific residual factor vector

# Joint factor analysis and I-vector modelling

- I-vectors gives the utterance level representation. Cosine distance can be used to find similarity between I vectors.
- Variable length sequence to fixed dimension representation
- I-vectors with **SVM** or **DNN** or **PLDA** scoring are used for speaker/emotion/language identification

# I-vector modelling in Language identification

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD OUTREACH DIVISION

talent sprint

**Motivation**

- I-vector models are the state-of-art baseline models in NIST 2009 Language Recognition Evaluation (**NIST 2009 LRE**) and Oriental language recognition (**OLR**) challenge.
- Allows low dimensional speech representation based on the Factor analysis
- Each speech recording is mapped on low dimensional vector (Ex: 400)
- Factor analysis as feature extractor
- Modeling the inter-language variability between different language classes

# I-vector modelling in Language identification

- $C_{avg}$ is the average pairwise loss between miss rate and false alarm rate;
- **EER** refers to equal error rate.
- Low values of these metrics implies better system performance

$C_{avg}$ AND EER RESULTS OF VARIOUS I-VECTOR BASELINE SYSTEMS ON 3 TEST CONDITIONS.

| | 1 second | | 3 second | | Full-Length | |
|---|---|---|---|---|---|---|
| System | $C_{avg}$ | EER% | $C_{avg}$ | EER% | $C_{avg}$ | EER% |
| i-vector | 0.1672 | 15.28 | 0.0695 | 7.59 | 0.0522 | 6.224 |
| i-vector + LDA | 0.1238 | 13.30 | 0.0494 | 5.95 | 0.0362 | 4.704 |
| i-vector (Linear SVM) | 0.1485 | 14.43 | 0.0624 | 6.07 | 0.0469 | 4.58 |
| i-vector (Poly SVM) | 0.1242 | 12.43 | 0.0470 | 4.83 | 0.0351 | 3.58 |
| i-vector (RBF SVM) | 0.1313 | 12.16 | 0.0495 | 4.59 | 0.0352 | 3.39 |

# I-vector modelling in Emotion recognition

**INTERSPEECH 2009 Emotion recognition Challenge:**
uses **FAU-AIBO** Emotion corpus, It is two class problem:
**Positive** or **Negative**

**GMM-MFCC systems (s1)**
MFCC feature (12-MFCC+E+delta+double delta)
GMM uses 512 Gaussian Components

**GMM-MFCC systems (s2)**
MFCC feature (12-MFCC+E+delta+double delta)
UBM with 512 Gaussian Components
I-vector dimension of 150, Fisher Discriminant Analysis

**GMM-Prosodic system (s3)**
Prosody features: Pitch+ energy + duration features
GMM-UBM with 256 components.

# I-vector modelling in Emotion recognition

**INTERSPEECH 2009 Emotion recognition Challenge:**
uses **FAU-AIBO** Emotion corpus, It is two class problem:
**Positive** or **Negative**

**GMM-MFCC systems (s1)**
MFCC feature (12-MFCC+E+delta+double delta)
GMM uses 512 Gaussian Components

**GMM-MFCC systems (s2)**
MFCC feature (12-MFCC+E+delta+double delta)
UBM with 512 Gaussian Components
I-vector dimension of 150, Fisher Discriminant Analysis

**GMM-Prosodic system (s3)**
Prosody features: Pitch+ energy + duration features
GMM-UBM with 256 components.

| System | Unweighted recall |
|---|---|
| S1 : GMM w/ MFCC | 69.72% |
| S2 : Ivector w/ MFCC | **69.81%** |
| S3 : GMM-UBM w/ long-term feature | 66.61% |
| S1+S2+S3 | **70.54%** |

# I-vector modelling in Speaker recognition

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD OUTREACH DIVISION

talent sprint

**Baseline results:** Comparison of JFA and i-vector systems on the common subset of the **2008 NIST SRE** database. **WCCN**: Within-class Covariance Normalisation, **LDA**: Linear discriminant analysis, **SDNAP**: scatter-difference Nuisance Attribute Projection, **GPLDA:** Gaussian Probabilistic LDA

| Utterance Length | JFA System | | LDA + WCCN | | SDNAP + WCCN | | GPLDA System | |
|---|---|---|---|---|---|---|---|---|
| (training-testing) | EER | DCF | EER | DCF | EER | DCF | EER | DCF |
| 2 sec - 2 sec | **35.25%** | 0.0988 | 35.35% | **0.0986** | 35.67% | 0.0999 | 36.16% | 0.0999 |
| 4 sec - 4sec | 30.48% | **0.0934** | 31.05% | 0.0966 | **30.23%** | 0.0968 | 31.30% | 0.0991 |
| 8 sec - 8 sec | **23.39%** | 0.0803 | 23.95% | **0.0800** | 23.56% | 0.0801 | 23.56% | 0.0837 |
| 10 sec - 10sec | 21.17% | **0.0738** | 21.56% | 0.0741 | 20.84% | 0.0737 | **20.34%** | 0.0762 |
| 20 sec - 20sec | 12.79% | 0.0533 | 13.41% | 0.0530 | 12.84% | **0.0528** | **11.87%** | 0.0532 |
| 50 sec - 50 sec | 6.51% | **0.0266** | 6.44% | 0.0310 | 6.42% | 0.0299 | **5.77%** | 0.0272 |
| full (2.5min) - full | 3.37% | **0.0149** | 3.54% | 0.0179 | 3.62% | 0.0166 | **3.13%** | 0.0168 |

EER: Equal error rate, DCF: Decision Cost Function is equivalent to pairwise loss function (C*avg*)

# Automatic speech recognition

**Automatic Speech recognition (ASR)**: is a transduction of spoken acoustic sequence to text sequence.

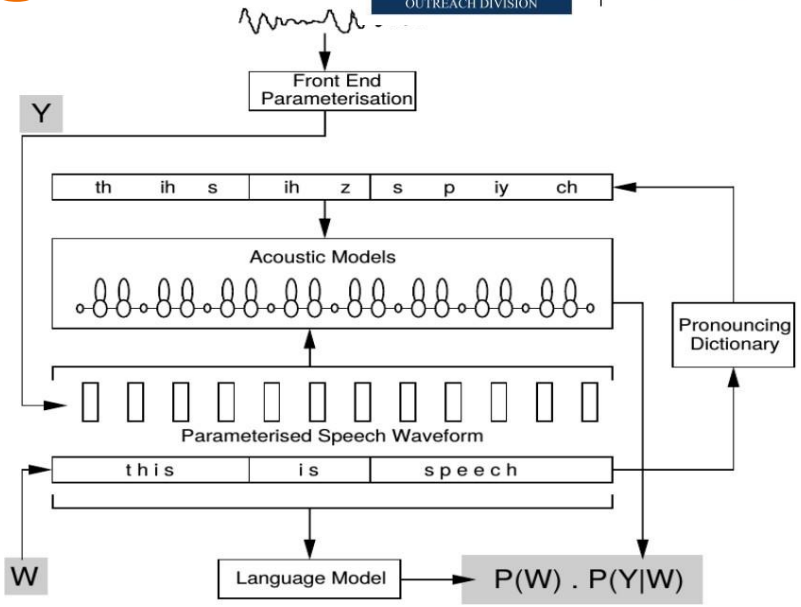
# Automatic speech recognition

**Building of ASR system required various knowledge sources:**

- **Acoustics** - knowledge about variability in speech
- **Phonetics** - knowledge about characteristics of speech sounds
- **Phonology** - knowledge about variability of speech sounds
- **Prosodics** - knowledge about stress and the intonation patterns
- **Lexical** - knowledge about patterns of language
- **Syntax** - knowledge about the grammatical structure of language
- **Semantics** - knowledge about the meaning of the words
- **Pragmatics** - knowledge about the context of conversion

**Automatic Speech recognition (ASR)**: is a transduction of spoken acoustic sequence to text sequence.

# Automatic speech recognition

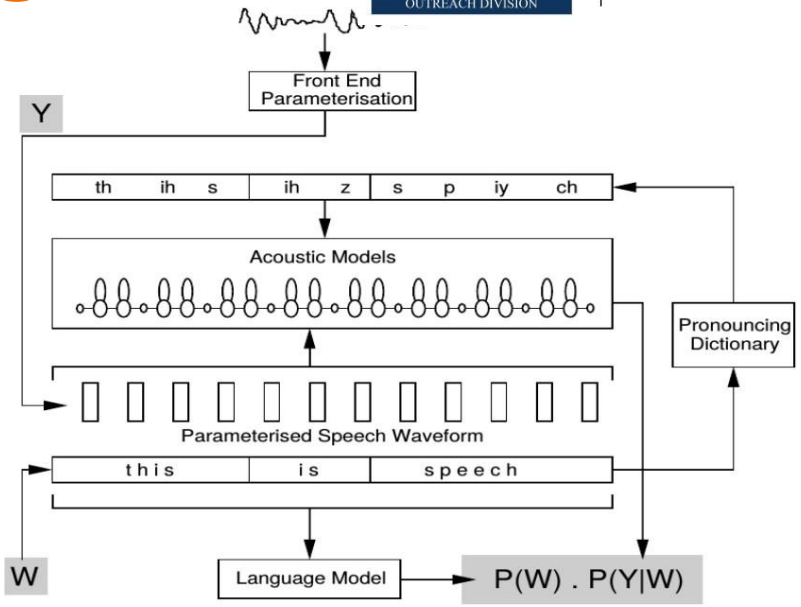


# Automatic speech recognition

Speech Recognition Problem: **P(W |Y )**

Y represents sequence of observation symbols (acoustic features MFCC), W represents the sequence of words.

Objective: **maximize P(W |Y )** during training



# Automatic speech recognition

Speech Recognition Problem: **P(W |Y )**

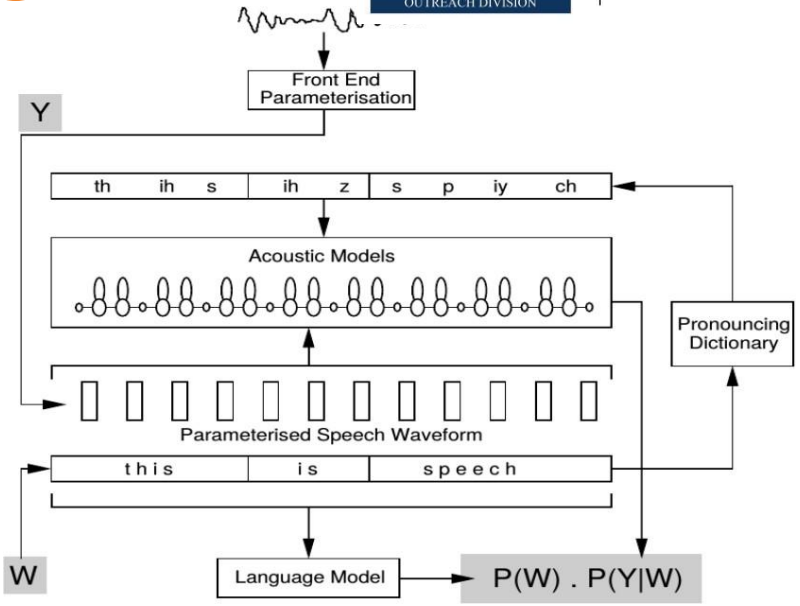
Y represents sequence of observation symbols (acoustic features MFCC), W represents the sequence of words.

Objective: **maximize P(W |Y )** during training

Bayesian formulation for speech recognition:

$P(W|Y) = P(Y|W)P(W)/P(Y)$

$$\hat{W} = \arg\max_W P(W|Y) = \arg\max_W \frac{P(W)P(Y|W)}{P(Y)}$$



# Automatic speech recognition

Speech Recognition Problem: **P(W |Y )**

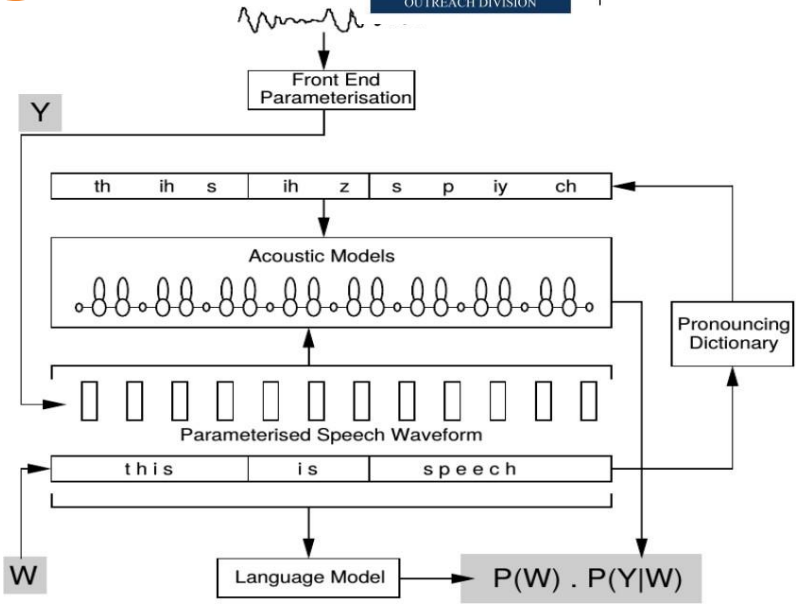
Y represents sequence of observation symbols (acoustic features MFCC), W represents the sequence of words.

Objective: **maximize P(W |Y )** during training

Bayesian formulation for speech recognition:

$P(W|Y) = P(Y|W)P(W)/P(Y)$

$$\hat{W} = \arg\max_W P(W|Y) = \arg\max_W \frac{P(W)P(Y|W)}{P(Y)}$$




P(Y |W ): likelihood function, P(W ): a priori probability distribution

- Performance of Speech Recognition Systems: **word error rate (WER) = (S+I+D)/N**
  Here S, I, D, C are the number of **substitutions**, **insertions**, **deletions** and **correct** words and N is (S + D + C)

# Automatic speech recognition



Widely used acoustic models:

- GMM-HMM based acoustic models
- DNN-HMM based acoustic models
- RNN-CTC based acoustic models
- Encoder-Decoder acoustic models

# Automatic speech recognition

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD
OUTREACH DIVISION

talent sprint

**Hidden markov models:**

A Markov chain is useful when we need to compute a **probability for a sequence** of observable events.

# Automatic speech recognition

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD OUTREACH DIVISION

talent sprint

**Hidden markov models:**
A Markov chain is useful when we need to compute a **probability for a sequence** of observable events.

**Three basic problems in HMM modelling:**

- **The Evaluation Problem:** Given an HMM and a sequence of observations O=O1, O2, O3, …… OT, what is the probability that the observations are generated by the model, P(O|λ)?

# Automatic speech recognition

**Hidden markov models:**
A Markov chain is useful when we need to compute a **probability for a sequence** of observable events.

**Three basic problems in HMM modelling:**

- **The Evaluation Problem:** Given an HMM and a sequence of observations O=O1, O2, O3, …… OT, what is the probability that the observations are generated by the model, P(O|λ)?

- **The Decoding Problem:** Given a model λ and a sequence of observations O=O1, O2, O3, …… OT, what is the most likely state sequence in the model that produced the observations?
  i,.e $Q* = q1, q2, q3, …….. qT$ ; $Q* = \arg\max_Q P(Q|O; \lambda)$; Here, q1, q2 … are referred to states.

# Automatic speech recognition

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD OUTREACH DIVISION

talent sprint

**Hidden markov models:**
A Markov chain is useful when we need to compute a **probability for a sequence** of observable events.

**Three basic problems in HMM modelling:**

- **The Evaluation Problem:** Given an HMM and a sequence of observations O=O1, O2, O3, …… OT, what is the probability that the observations are generated by the model, P(O|λ)?

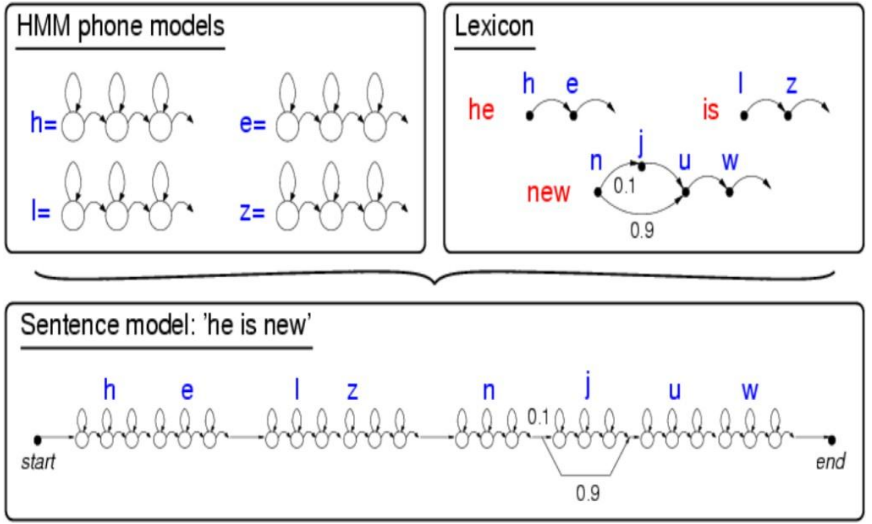
- **The Decoding Problem:** Given a model λ and a sequence of observations O=O1, O2, O3, …… OT, what is the most likely state sequence in the model that produced the observations?
  i,.e $Q* = q1, q2, q3, …….. qT$ ; $Q* = \arg\max_Q P(Q|O; \lambda)$; Here, q1, q2 … are referred to states.

- **The Learning Problem** Given a model λ and a sequence of observations O=O1, O2, O3, …… OT, how should we adjust the model parameters **λ = (A, B, π )** in order to maximize P(O|λ)

# Automatic speech recognition

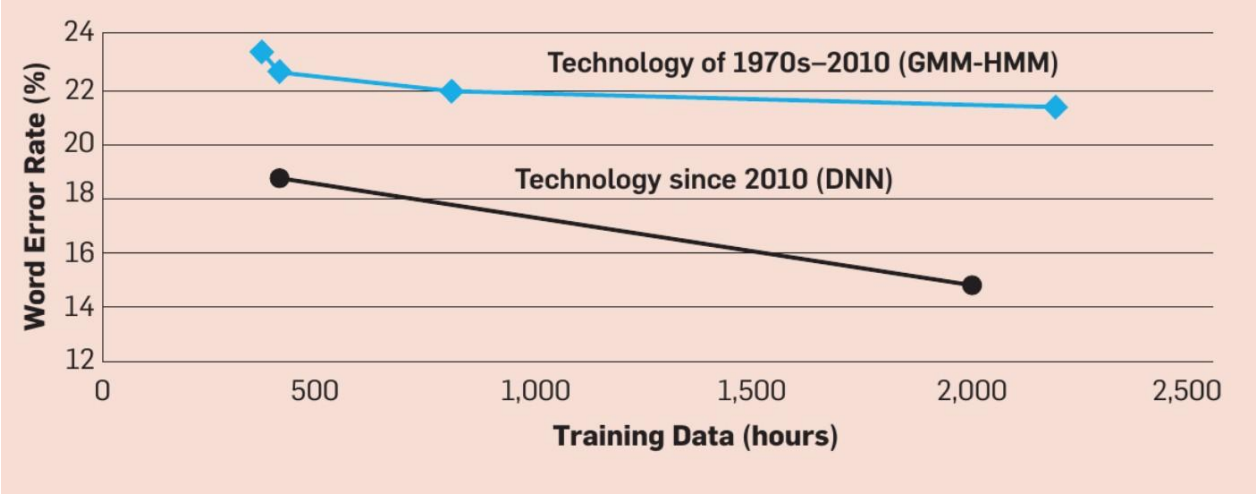
**HMM-GMM based acoustic models**

- Spoken word can be decomposed into sequence of $\mathbf{K_w}$ basic phones (units) and the sequence called **pronunciation sequence**.
- HMMs model the temporal variability of speech and GMMs model how well each frame or a short window of frames fits a state of HMM
- In practice each phone is represented by a HMM with **left to right topology** and **three hidden states.**




Blockdiagram describing HMM-GMM based acoustic modeling.

# Automatic speech recognition




Improvements in the performances of speech recognition systems with relevance to acoustic models (GMM-HMM, and DNN-HMM) as function training data

# Automatic speech recognition

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD OUTREACH DIVISION

talent sprint

Performances of speech recognition systems developed using the WSJ corpus.

| Acoustic modeling approaches | LM | %WER (14 Hrs) | %WER (81 Hrs) |
|---|---|---|---|
| CD-HMM-GMM [15] | Trigram | - | 12.42 (4) |
| CD-HMM-DNN [15] | None | - | - |
| | Dictionary | 56.1 | 51.1 |
| | Monogram | 23.4 | 19.9 |
| | Bigram | 11.6 | 9.4 |
| | Trigram | 9.4 | 7.8 (3.5) |

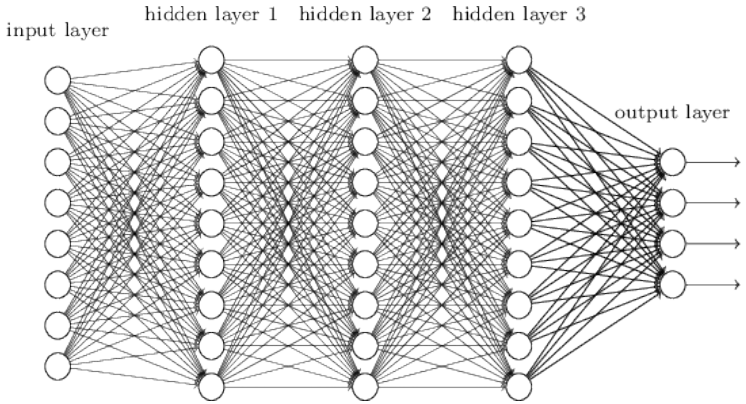
# Thank you

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD
OUTREACH DIVISION

talent sprint

# Part-II
# Advanced AI & ML in Speech Systems

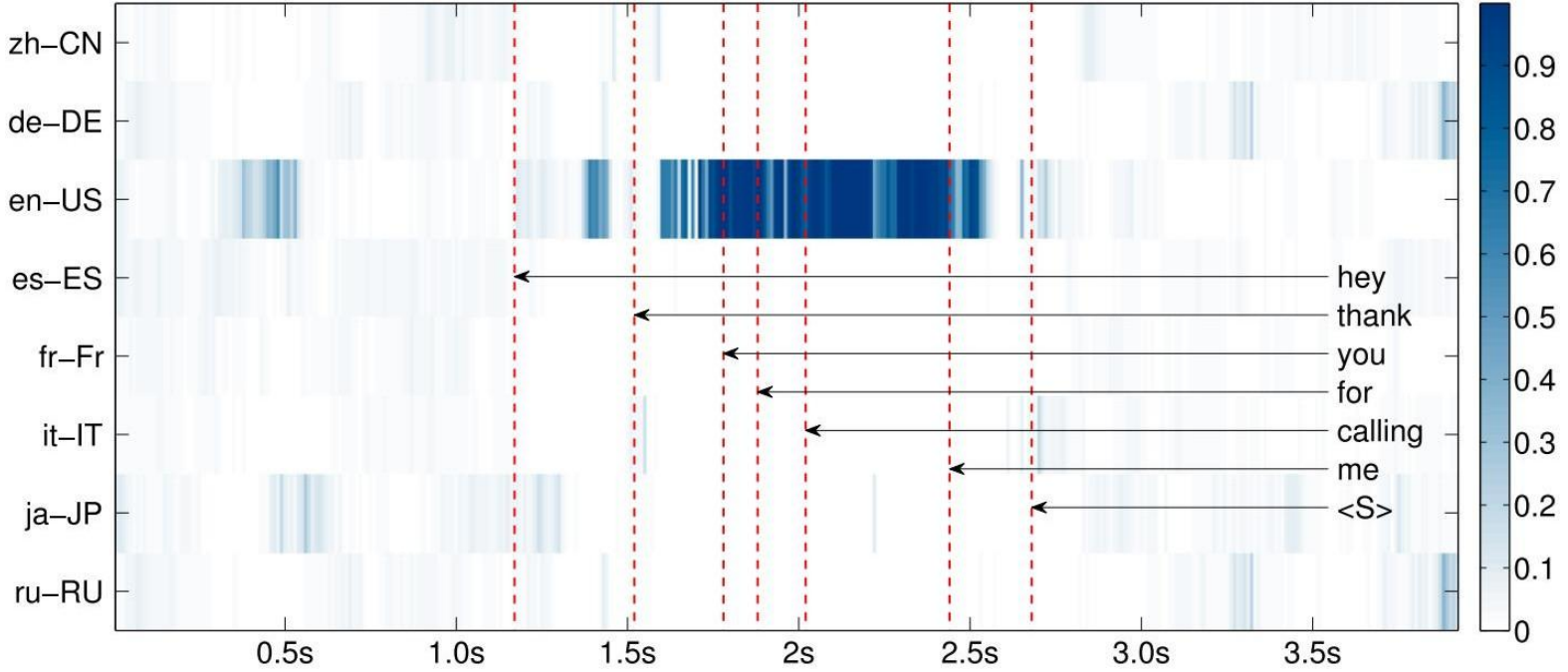
# Deep Neural Network




Deep neural network

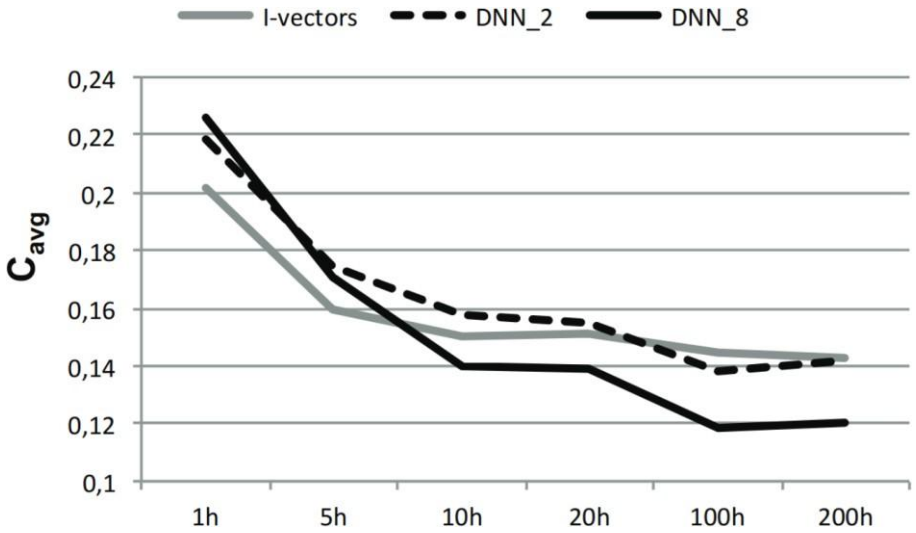
- Decision is taken at frame level.
- The frame level decisions are averaged to get utterance level decision.

# Deep Neural Network



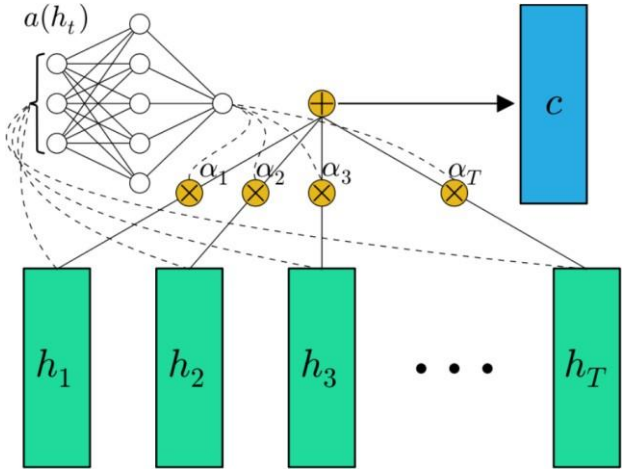

Frame level probabilities of a DNN-based LID system (8 languages selected) evaluated over an English-USA (4s) test utterance.

# Deep Neural Network




i-vector vs DNN performance on LRE09 database.
$C_{avg}$=average cost.

# DNN with attention

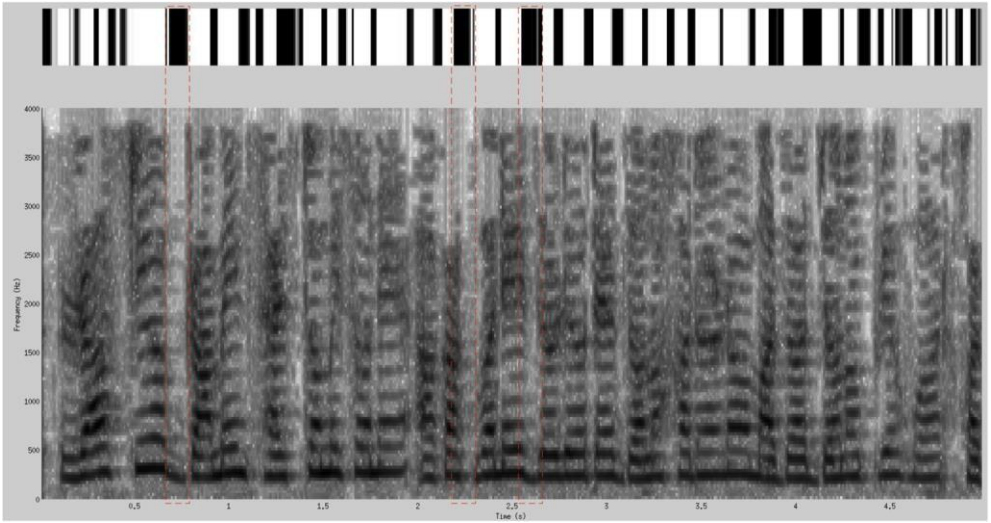



DNN with attention architecture

Context vector (c) is weighted average of hidden representations

- All frames may not contribute for decision / classification
- Decision is taken at utterance level.
- Weighted average of hidden representations

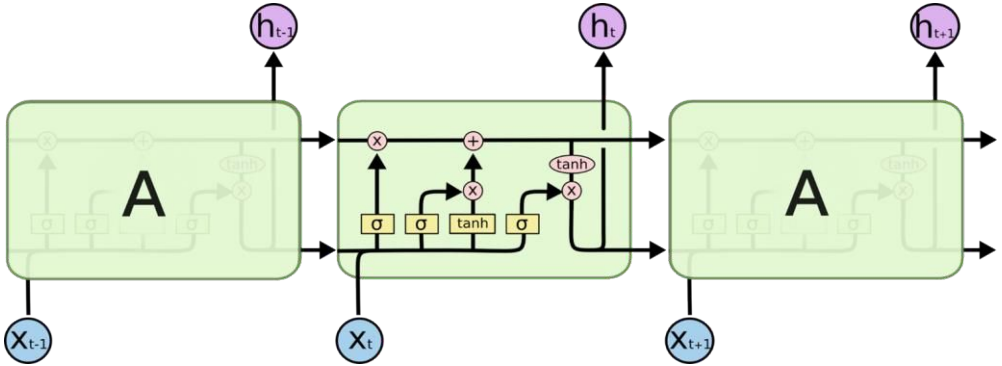
# DNN with attention




An example of spectrogram with attention

- Attention weights are low for silence frames

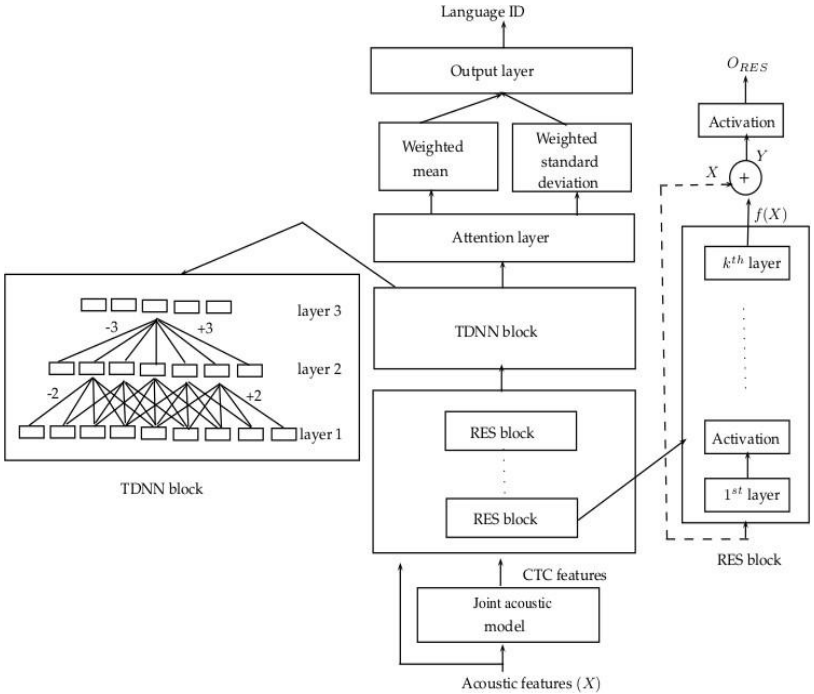
# Sequential Networks




LSTM cell

- Long temporal dependencies
- Sequence to sequence mapping

# Attention based residual time delay neural network




Residual blocks allows skip connections which provide smooth flow of gradients

In TDNN each layer captures temporal dependencies at different context

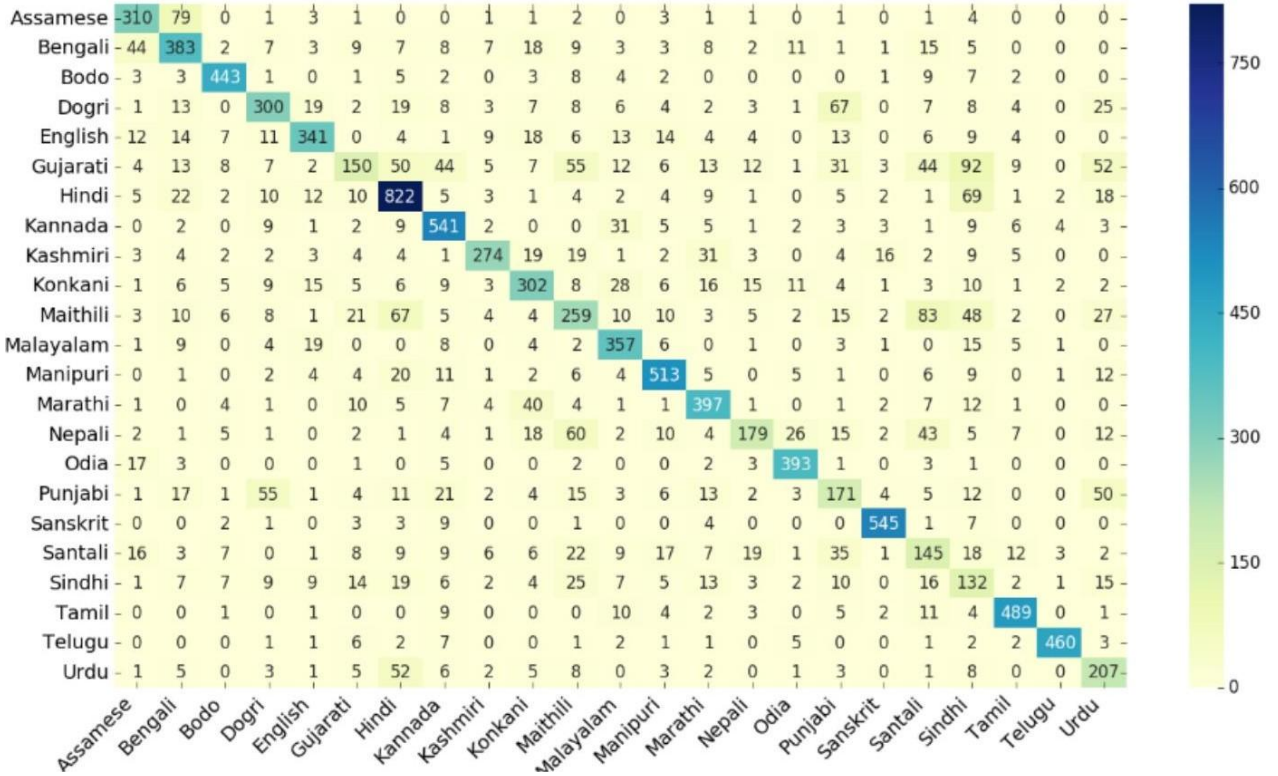
Attention aggregates the whole input sequence information

# Performance of Indian LID system using DNNs

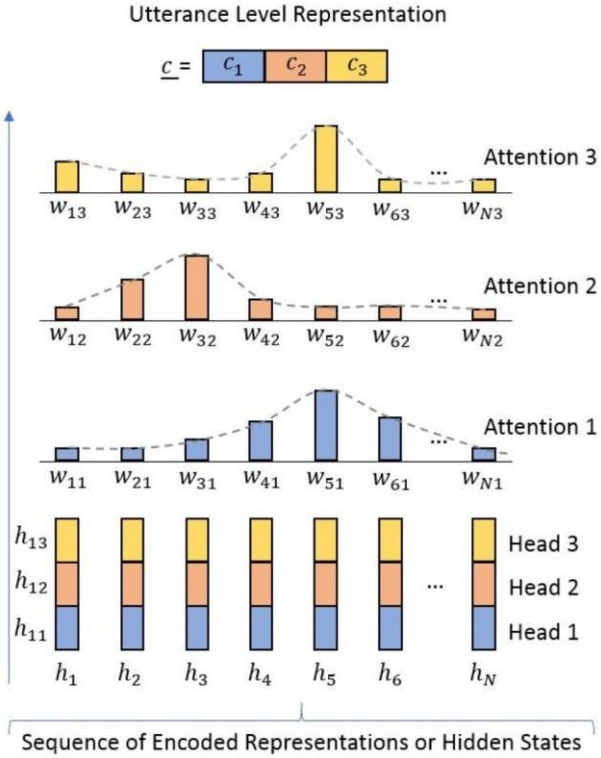
| Network | Features | | | |
|---|---|---|---|---|
| | MFCC | SDC | i-vector | Phonetic |
| DNN | 22.42 | 17.95 | 14.72 | **13.34** |
| DNN-RES | 21.87 | 17.12 | 14.25 | **12.56** |
| LSTM | 20.05 | 16.59 | 14.08 | **12.22** |
| Attention based RES-TDNN | **15.45** | **13.81** | **13.68** | **9.46** |

Results on IIITH-ILSC database using different neural networks

# Performance of Indian LID system using DNNs

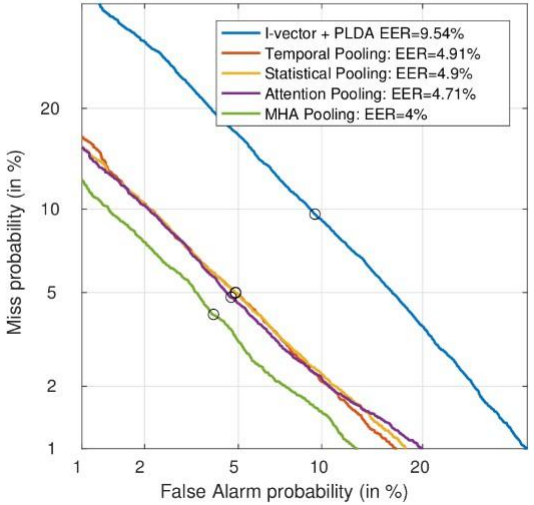


# Performance of Speaker verification using Attention Networks




An example of the Self Multi-Head Attention Pooling with 3 heads

| Approach | DCF | EER |
|---|---|---|
| I-vector + PLDA | 0.0078 | 9.54 |
| CNN + Temporal Pooling | 0.0047 | 4.91 |
| CNN + Statistical Pooling | 0.0046 | 4.9 |
| CNN + Att. Pooling | 0.005 | 4.71 |
| CNN + MHA Pooling | **0.0045** | **4.0** |




Evaluation results of the text-independent verification on VocCelb database.

# Performance of Speaker identification using Attention Networks

The results for speaker identification on VoxCeleb

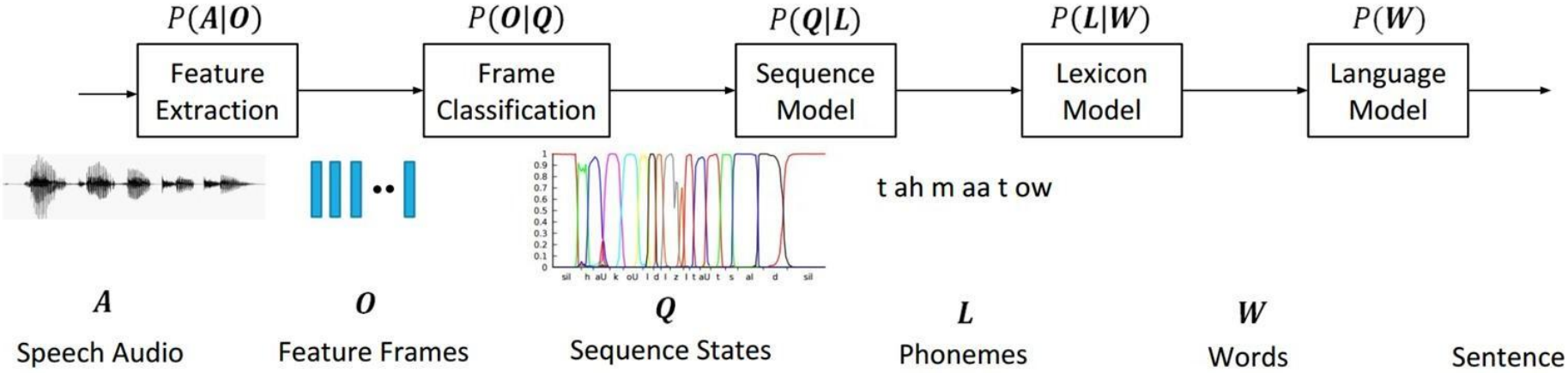
| Accuracy | Top-1 (%) | Top-5(%) |
|---|---|---|
| I-Vectors + SVM [5] | 49.0 | 56.6 |
| I-Vectors + PLDA + SVM [5] | 60.8 | 75.6 |
| I-Vectors + LogReg [6] | 65.8 | 81.4 |
| VGG-like CNN+ TAP [5] | 80.5 | 92.1 |
| ResNet-34 + TAP [6] | 88.5 | 94.9 |
| ResNet-34 + SAP [6] | 89.2 | 94.1 |
| ResNet-34 + LDE [6] | 89.9 | 95.7 |
| VGG-like CNN+Self-Attention (ours) | 88.2 | 93.8 |
| ResNet-18+Self-Attention (ours) | **90.8** | **96.5** |

The results of the three most frequently used acoustic features for speaker identification on VoxCeleb. Here, Spectr. corresponds to the spectrograms feature.

| Model | Feature Type | Top-1 (%) | Top-5(%) |
|---|---|---|---|
| VGG-like CNN [5] | Spectr. | 80.5 | 92.1 |
| VGG-like CNN+Self-Attention (ours) | Spectr. | 85.3 | 92.9 |
| ResNet-18+Self-Attention (ours) | Spectr. | 87.2 | 93.3 |
| VGG-like CNN+TAP [5] | MFCCs | 82.4 | 92.8 |
| VGG-like CNN+Self-Attention (ours) | MFCCs | 87.4 | 93.5 |
| ResNet-18+Self-Attention (ours) | MFCCs | 88.5 | 94.8 |
| ResNet-34 [6] | FBank | 89.9 | 95.7 |
| VGG-like CNN+Self-Attention (ours) | FBank | 88.2 | 93.8 |
| ResNet-18+Self-Attention (ours) | FBank | **90.8** | **96.5** |

# Automatic speech recognition (recap)

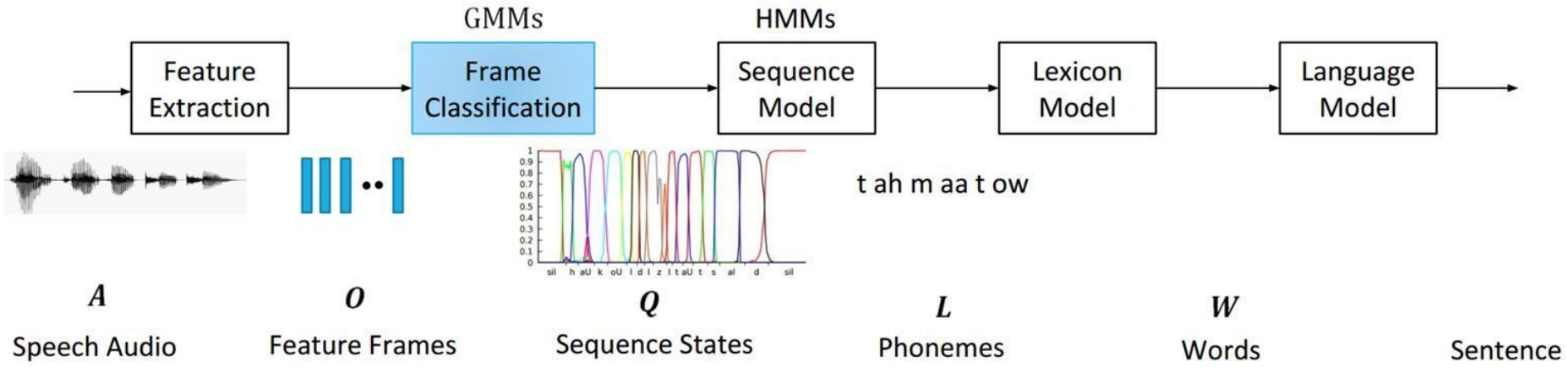
$$\widehat{W} = \underset{W}{\operatorname{argmax}} P(W|O) = \underset{W}{\operatorname{argmax}} P(A|O)P(O|Q)P(Q|L)P(L|W)P(W)$$



# Automatic speech recognition (recap)

GMMs: Gaussian Mixture Models

HMMs: Hidden Markov Models



# Lexical model

Lexical modelling forms the bridge between the acoustic and language models

Prior knowledge of language

Mapping between words and the acoustic units (phoneme is most common)

Deterministic

| Word | Pronunciation |
|---|---|
| TOMATO | t ah m aa t ow |
| | t ah m ey t ow |
| COVERAGE | k ah v er ah jh |
| | k ah v r ah jh |

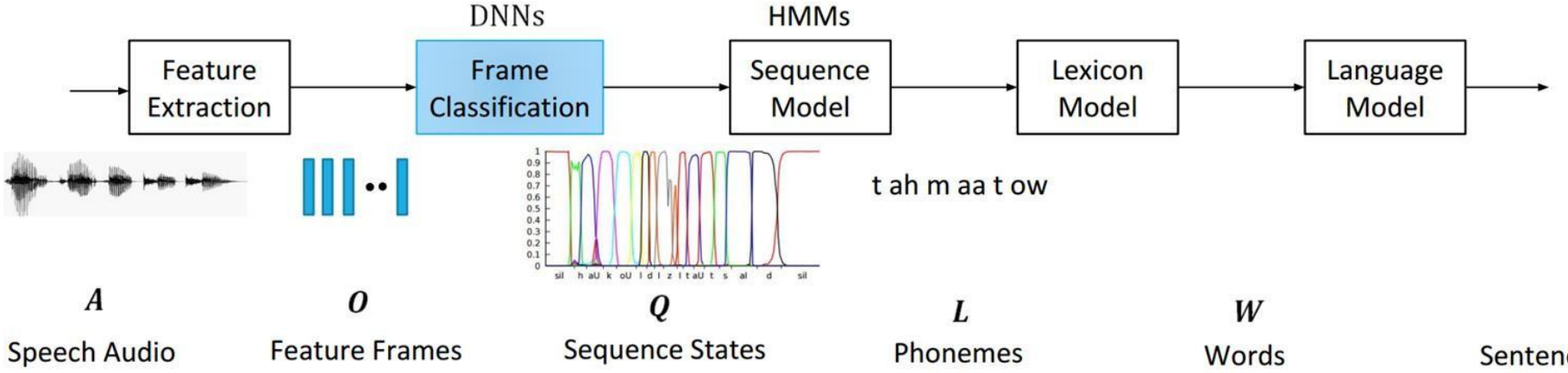
Probabilistic

| Word | Pronunciation | Probability |
|---|---|---|
| TOMATO | t ah m aa t ow | 0.45 |
| | t ah m ey t ow | 0.55 |
| COVERAGE | k ah v er ah jh | 0.65 |
| | k ah v r ah jh | 0.35 |

# DNN-HMM in Speech Recognition

DNN: Deep Neural Networks

HMMs: Hidden Markov Models



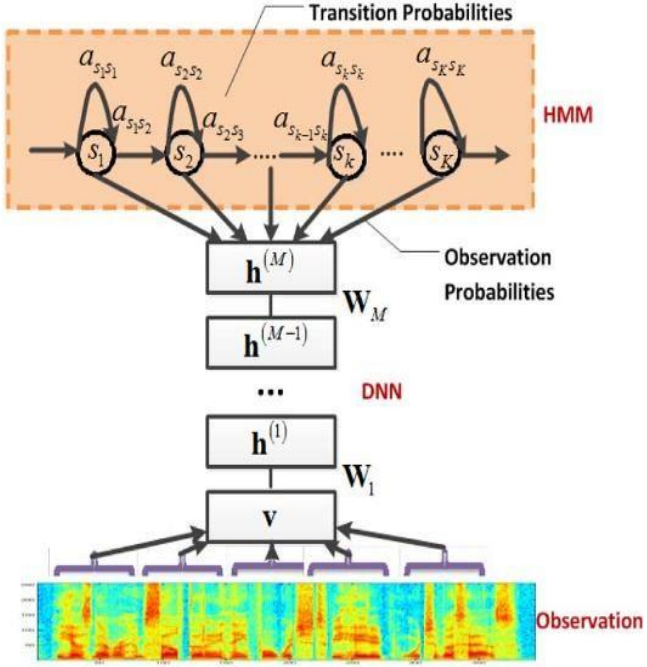
# DNN-HMM in Speech Recognition

1990s: Large vocabulary continuous dictation

2000s: Discriminative training
(minimize word/phone error rate)

2010s: Deep learning significantly

reduce error rate




George E. Dahl, et al. Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition. IEEE Trans. Audio, Speech & Language Processing, 2012.

# DNN-HMM vs. GMM-HMM



**Deep models are more powerful**

- GMM assumes data is generated from single component of mixture model
- GMM with diagonal variance matrix ignores correlation between dimensions

**Deep models take data more efficiently**

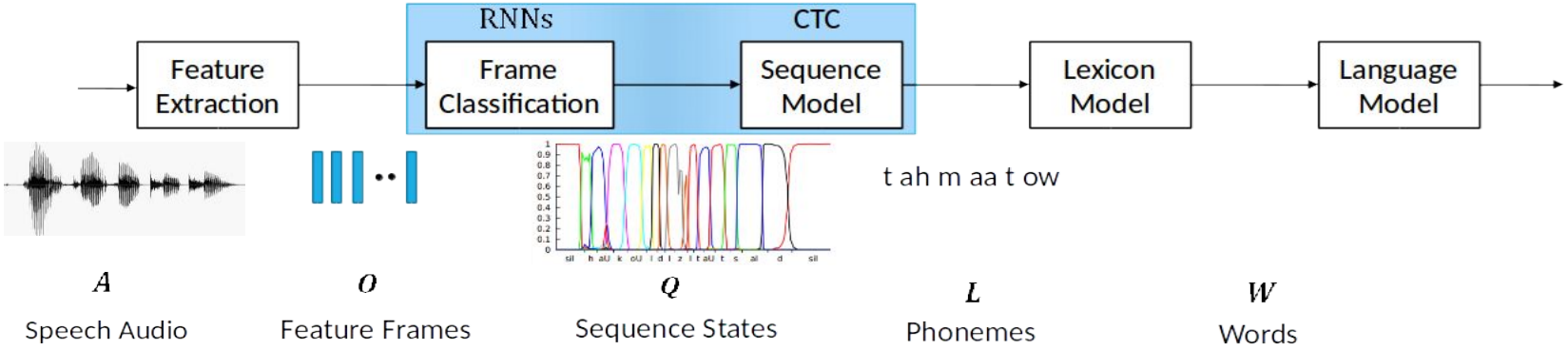
- GMM consists with many components and each learns from a small fraction of data

**Deep models can be further improved by recent advances in deep learning**

# RNN-CTC in Speech Recognition

RNN: Recurrent Neural Networks

CTC: Connectionist Temporal Classification



# Connectionist Temporal Classification (CTC)

- No need of aligned data

  **Reason**:It can assign probability for any label, given an input

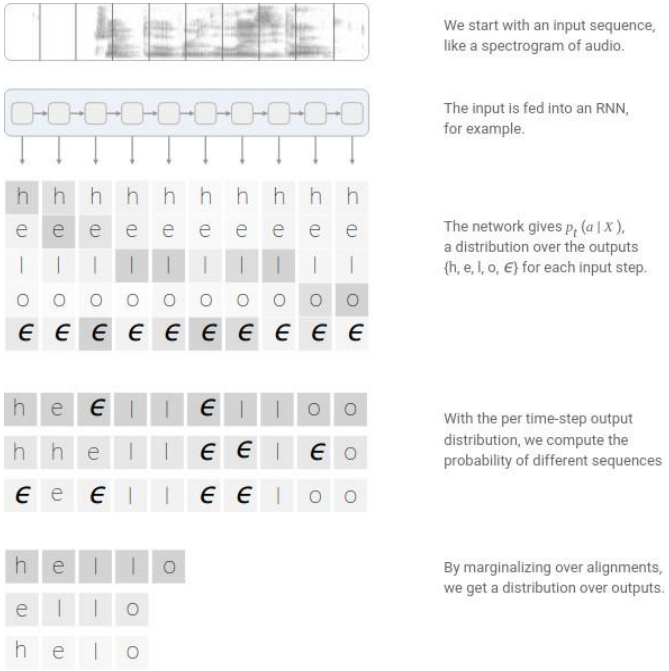
- It works by summing over the probability of all possible alignments between the input and the label

- **X** = $\{x_1, x_2 ... x_T\}$ represents input sequences
- **Y** = $\{y_1, y_2 ... y_T\}$ Transcripts

- Need an **accurate mapping** from X to Y
- Challenges using simpler supervised learning algorithms:
  - Both X and Y can vary in length.

# Loss function

- The CTC alignments give us a natural way to go from **probabilities at each time-step** to the **probability of an output sequence**
- conditional probability $\mathbf{P(Y/X) = A \in A_{X,Y} \prod_{t=1}^{T} p_t(a_t|X)}$
- $\mathbf{p_t(a_t|X)}$ computes the probability for a single alignment step-by-step.



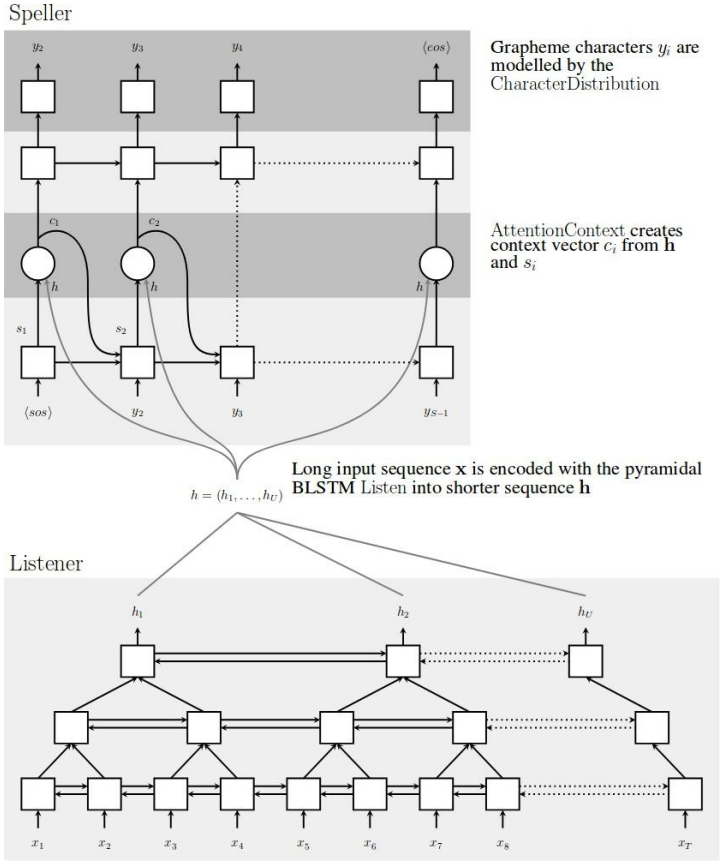
# LSTM-CTC MODELS: switch board corpus



Word error rate compared with HMM based models

| | Context | Error Rate (%) |
|---|---|---|
| LSTM-HMM | Uni | 8.9 |
| | Bi | 9.1 |
| LSTM-CTC | Uni | 9.4 |
| | Bi | 8.5 |

# Encoder Decoder models for speech recognition




Listen, Attend and Spell (LAS) model:

- Listener is a pyramidal BLSTM encoding our input sequence x into high level features h.
- Epeller is an attention-based decoder generating the y characters from h.

# Deep learning in ASR on Chime and switch board

| WER (SWB) | WER (full=SWB+CH) | Paper | Notes |
|---|---|---|---|
| 5.5% | 10.3% | English Conversational Telephone Speech Recognition by Humans and Machines | ResNet + BiLSTMs acoustic model, with 40d FMLLR + i-Vector inputs, trained on SWB+Fisher+CH, n-gram + model-M + LSTM + Strided (à trous) convs-based LM trained on Switchboard+Fisher+Gigaword+Broadcast |
| 6.3% | 11.9% | The Microsoft 2016 Conversational Speech Recognition System | VGG/Resnet/LACE/BiLSTM acoustic model trained on SWB+Fisher+CH, N-gram + RNNLM language model trained on Switchboard+Fisher+Gigaword+Broadcast |
| 6.6% | 12.2% | The IBM 2016 English Conversational Telephone Speech Recognition System | RNN + VGG + LSTM acoustic model trained on SWB+Fisher+CH, N-gram + "model M" + NNLM language model |
| 8.5% | 13% | Purely sequence-trained neural networks for ASR based on lattice-free MMI | HMM-BLSTM trained with MMI + data augmentation (speed) + iVectors + 3 regularizations + Fisher |
| 9.2% | 13.3% | Purely sequence-trained neural networks for ASR based on lattice-free MMI | HMM-TDNN trained with MMI + data augmentation (speed) + iVectors + 3 regularizations + Fisher (10% / 15.1% respectively trained on SWBD only) |
| 12.6% | 16% | Deep Speech: Scaling up end-to-end speech recognition | CNN + Bi-RNN + CTC (speech to letters), 25.9% WER if trained *only* on SWB |

HMM based system still performs better than End-to-End system on large scale dataset

# Thank you