CS 224S / LINGUIST 285
Spoken Language Processing

Andrew Maas
Stanford University
Spring 2022

Lecture 9: Acoustic Modeling, Feature Extraction, HMM-DNN models

Original slides by Dan Jurafsky

Outline for Today

- HMM-GMM acoustic models
  - Mixtures of Gaussians
  - Subphone states
- HMM-DNN models
- Feature extraction & data augmentation

Noisy Channel Model

- Probabilistic implication: Pick the highest prob S:

$$\hat{W} = \arg\max_{W \in L} P(W \mid O)$$

- We can use Bayes rule to rewrite this:

$$\hat{W} = \arg\max_{W \in L} \frac{P(O \mid W)P(W)}{P(O)}$$

- Since denominator is the same for each candidate sentence W, we can ignore it for the argmax:

$$\hat{W} = \arg\max_{W \in L} P(O \mid W)P(W)$$

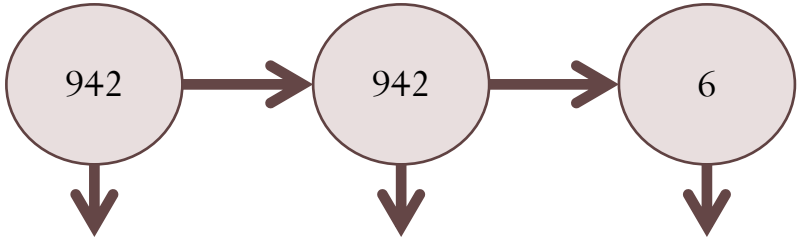
Generative HMM-GMM ASR model

Transcription: Samson

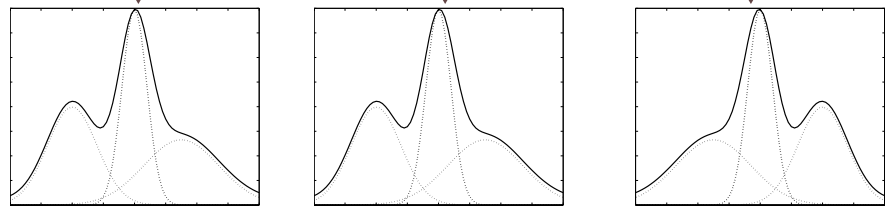
Pronunciation: S – AE – M – S –AH – N

Sub-phones : 942 – 6 – 37 – 8006 – 4422 …

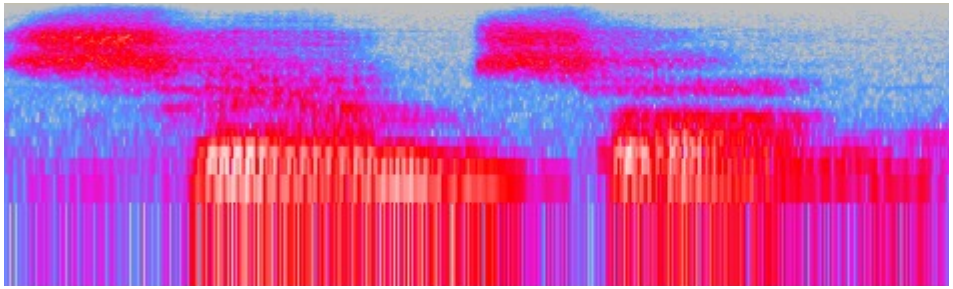
Hidden Markov Model (HMM):



Acoustic Model:



Audio Input:



GMM models:
P(x|s)
x: input features
s: HMM state

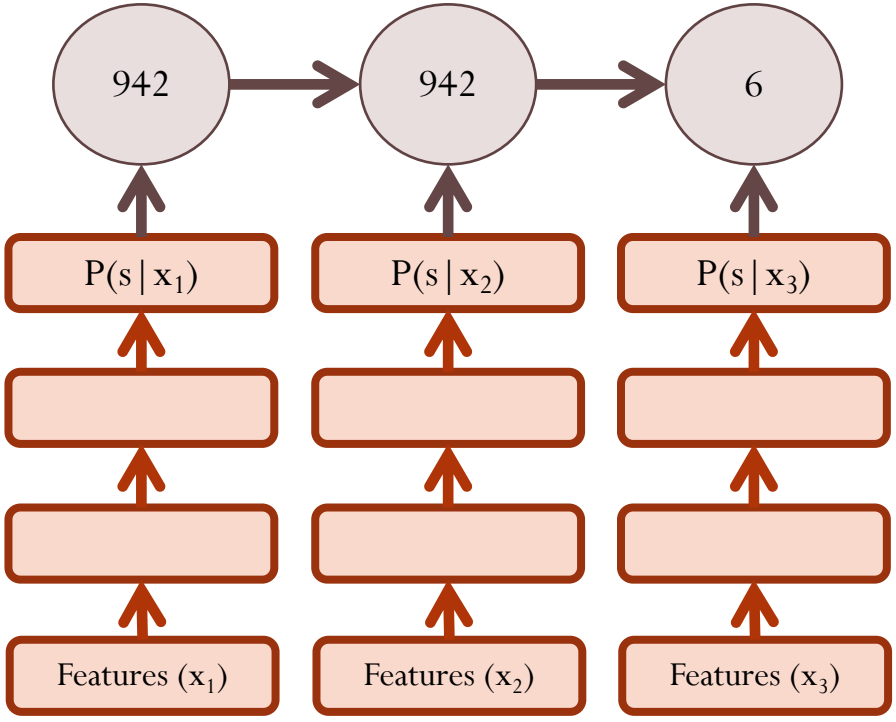
DNN Hybrid Acoustic Models

Transcription: Samson

Pronunciation: S – AE – M – S –AH – N

Sub-phones : 942 – 6 – 37 – 8006 – 4422 …

Hidden Markov Model (HMM):



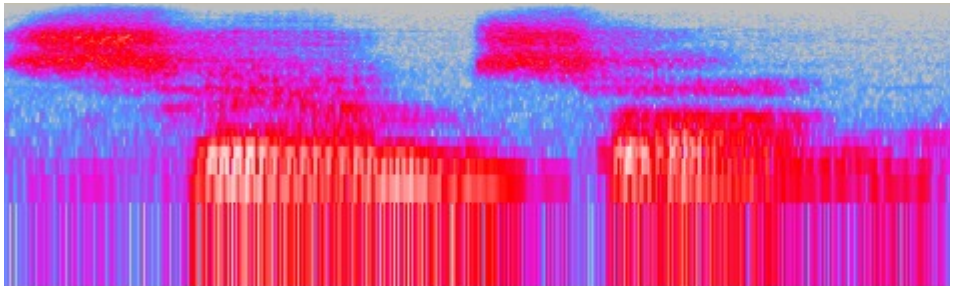
Acoustic Model:

Use a DNN to approximate: P(s|x)

Apply Bayes' Rule: P(x|s) = P(s|x) * P(x) / P(s)

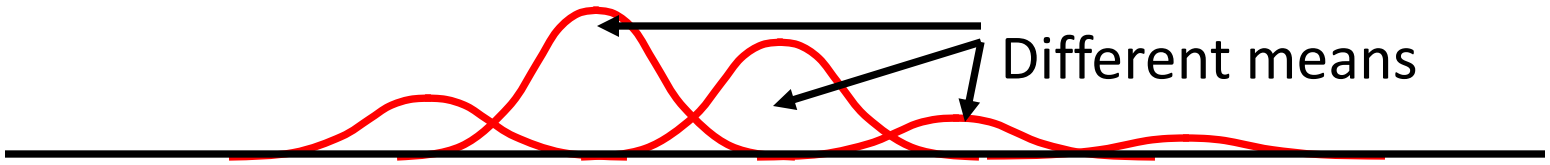
DNN * Constant / State prior

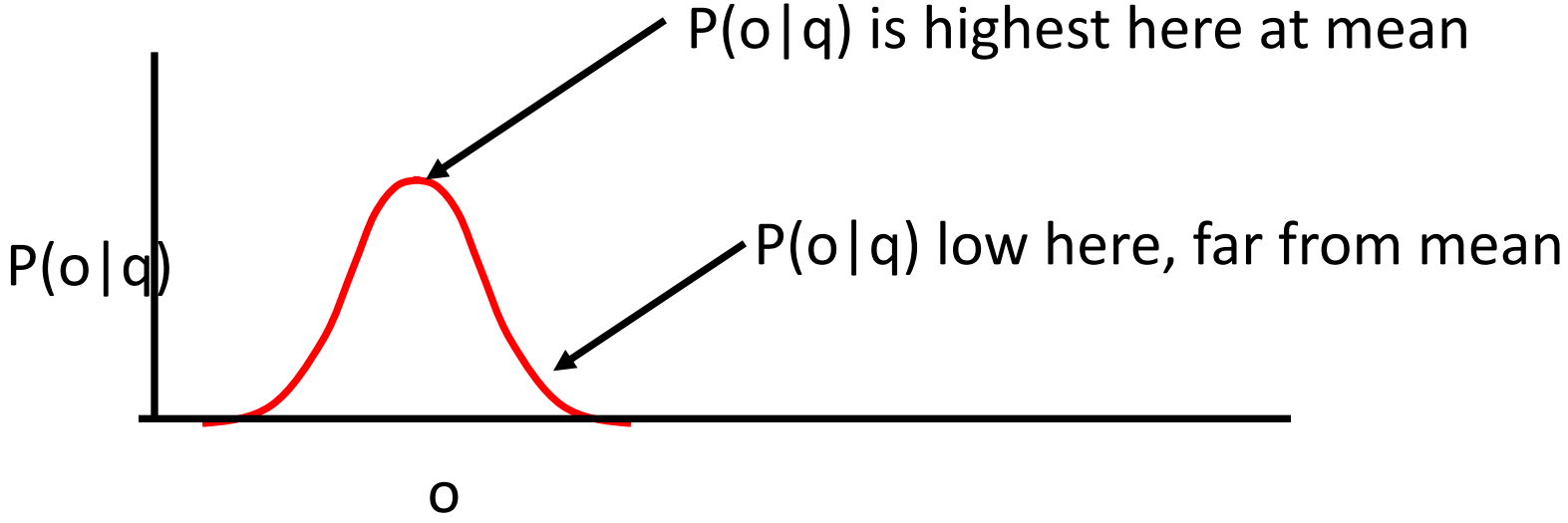
Audio Input:



Gaussians for Acoustic Modeling

- P(o|q): A Gaussian parameterized by mean and variance:



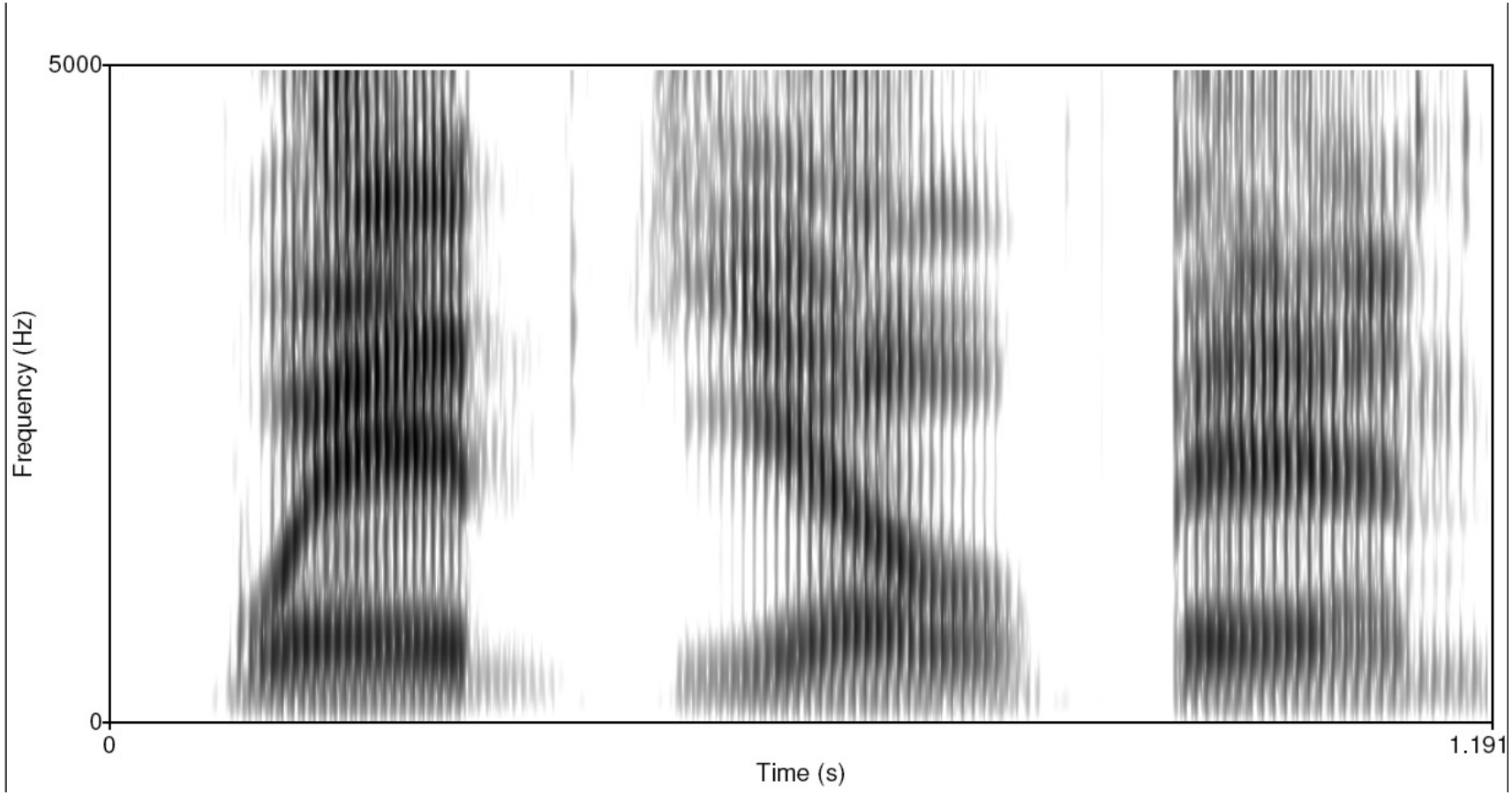


GMMs

- Summary: each state has a likelihood function parameterized by:
  - M Mixture weights
  - M Mean Vectors of dimensionality D
  - Either
    - M Covariance Matrices of DxD
  - Or more likely
    - M Diagonal Covariance Matrices of DxD
    - which is equivalent to
    - M Variance Vectors of dimensionality D

Phonetic context: different “eh”s

w eh d y eh l b eh n



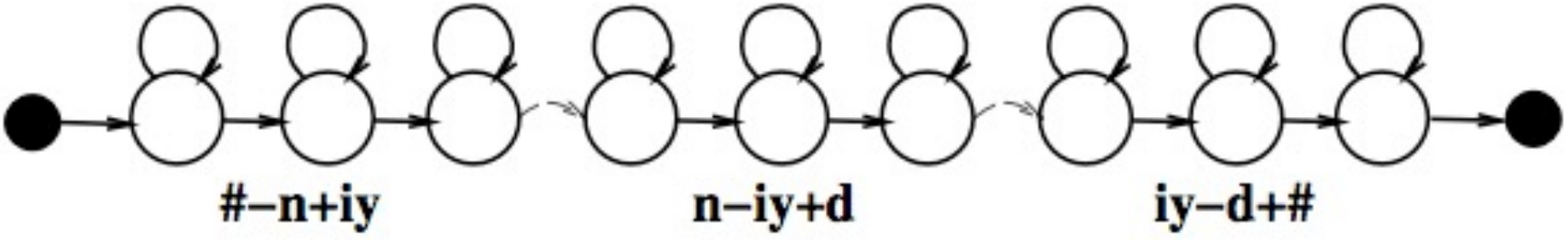
Modeling phonetic context

- The strongest factor affecting phonetic variability is the neighboring phone
- How to model that in HMMs?
- Idea: have phone models which are specific to context.
- Instead of Context-Independent (CI) phones
- We'll have Context-Dependent (CD) phones

Context dependent (CD) phones: triphones

- Triphones
- Each triphone captures facts about preceding and following phone
- Monophone:
  - p, t, k
- Triphone:
  - iy-p+aa
  - a-b+c means “phone b, preceding by phone a, followed by phone c”

“Need” with triphone models



Word-Boundary Modeling

- Word-Internal Context-Dependent Models

  ‘OUR LIST’:

  SIL AA+R AA-R L+IH L-IH+S IH-S+T S-T

- Cross-Word Context-Dependent Models

  ‘OUR LIST’:

  SIL-AA+R AA-R+L R-L+IH L-IH+S IH-S+T S-T+SIL

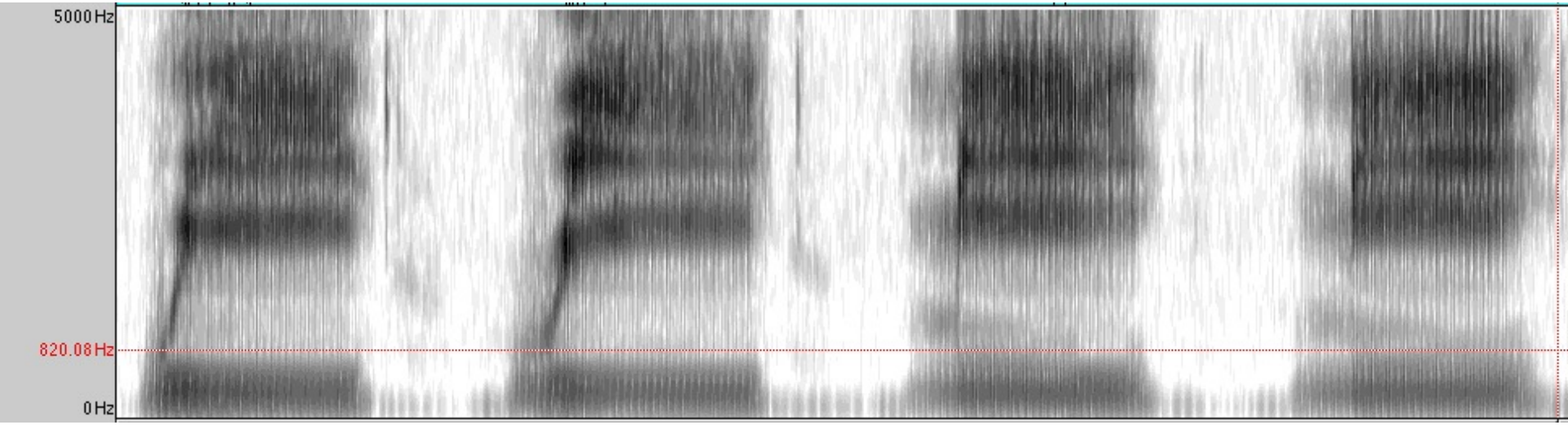
- Dealing with cross-words makes decoding harder!

Implications of Cross-Word Triphones

- Possible triphones: 50x50x50=125,000
- How many triphone types actually occur?
- 20K word WSJ Task, numbers from Young et al
- Cross-word models: need 55,000 triphones
- But in training data only 18,500 triphones occur!
- Need to generalize models

Modeling phonetic context: some contexts look similar

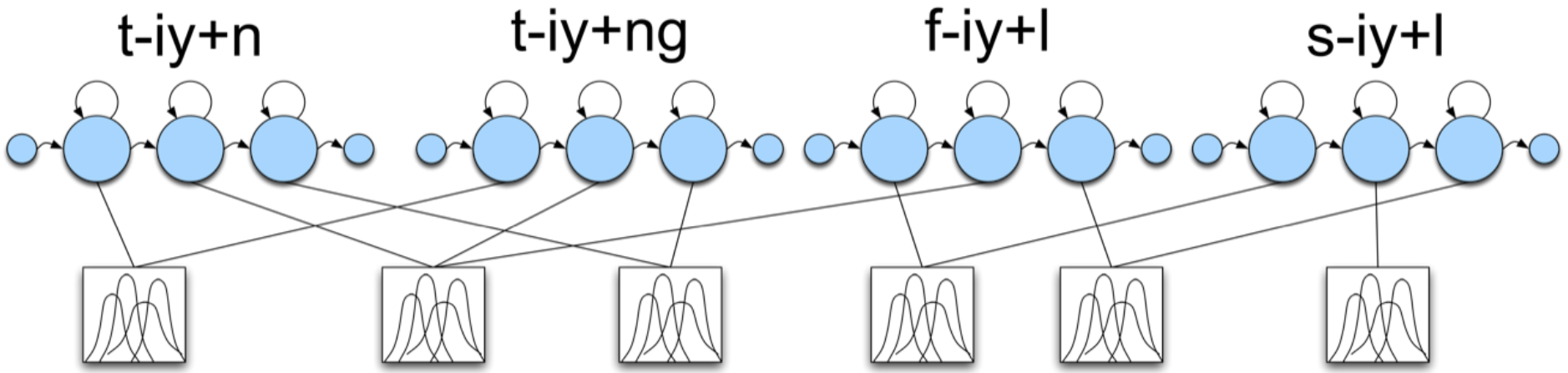
w iy r iy m iy n iy



Solution: State Tying

- Young, Odell, Woodland 1994
- Decision-Tree based clustering of triphone states
- States which are clustered together will share their Gaussians
- We call this “state tying”, since these states are “tied together” to the same Gaussian.

Young et al state tying



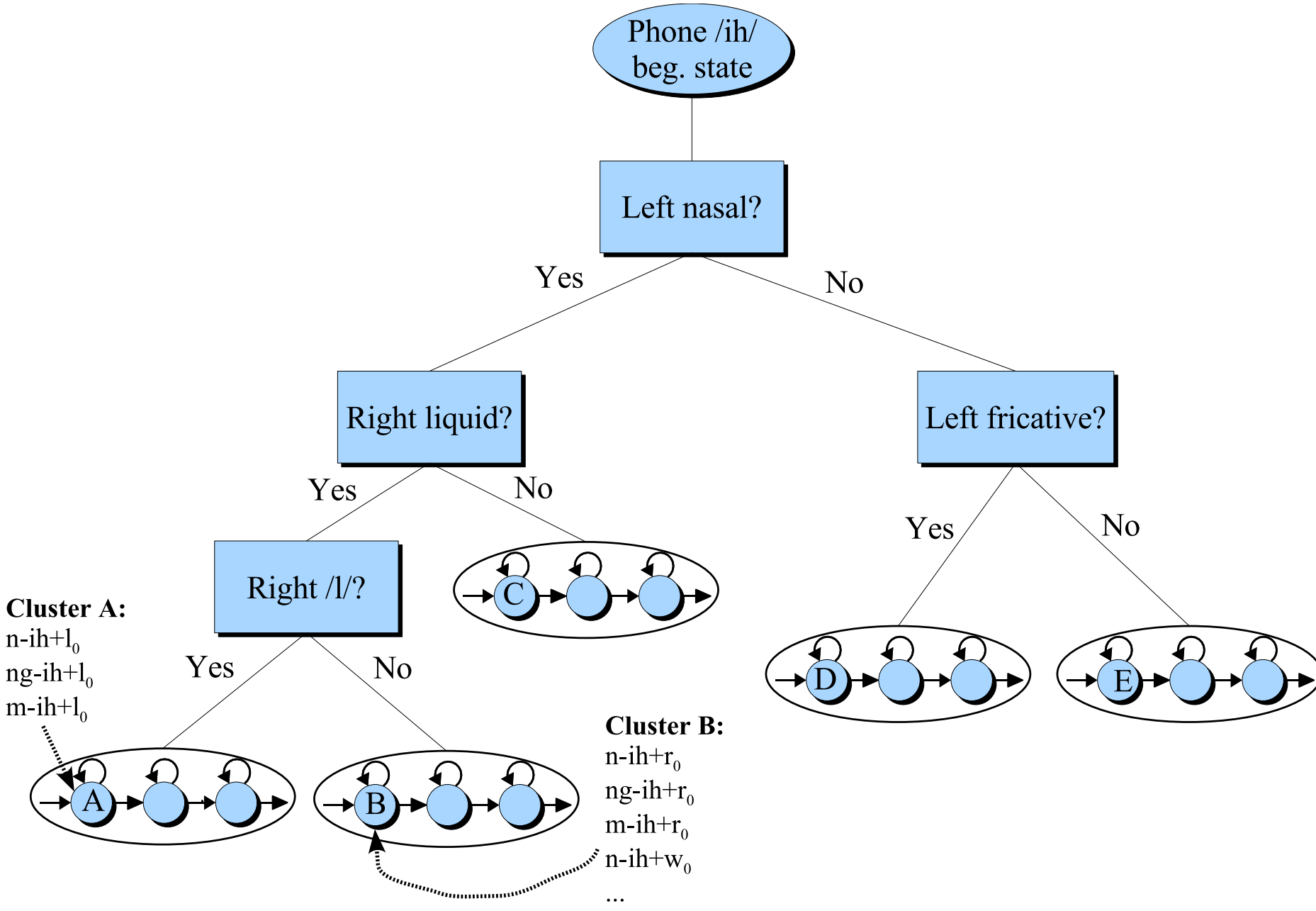
State tying/clustering

- How do we decide which triphones to cluster together?
- Use phonetic features (or ‘broad phonetic classes’)
  - Stop
  - Nasal
  - Fricative
  - Sibilant
  - Vowel
  - Lateral

Decision tree for clustering triphones for tying

Feature	Phones
Stop	b d g k p t
Nasal	m n ng
Fricative	ch dh f jh s sh th v z zh
Liquid	l r w y
Vowel	aa ae ah ao aw ax axr ay eh er ey ih ix iy ow oy uh uw
Front Vowel	ae eh ih ix iy
Central Vowel	aa ah ao axr er
Back Vowel	ax ow uh uw
High Vowel	ih ix iy uh uw
Rounded	ao ow oy uh uw w
Reduced	ax axr ix
Unvoiced	ch f hh k p s sh t th
Coronal	ch d dh jh l n r s sh t th z zh

Triphone decision tree clustering



Summary: Acoustic Modeling for LVCSR

- Increasingly sophisticated models
- For each state:
  - Gaussians
  - Multivariate Gaussians
  - Mixtures of Multivariate Gaussians
- Where a state is progressively:
  - CI Phone
  - CI Subphone (3ish per phone)
  - CD phone (=triphones)
  - State-tying of CD phone
- Neural network acoustic models *after* the above

Summary: ASR Architecture

- Five easy pieces: ASR Noisy Channel architecture
  - Feature Extraction:
    - 39 "MFCC" features
  - Acoustic Model:
    - Gaussians for computing p(o|q)
  - Lexicon/Pronunciation Model
    - HMM: what phones can follow each other
  - Language Model
    - N-grams for computing p(w_i|w_(i-1))
  - Decoder
    - Viterbi algorithm: dynamic programming for combining all these to get word sequence from speech!

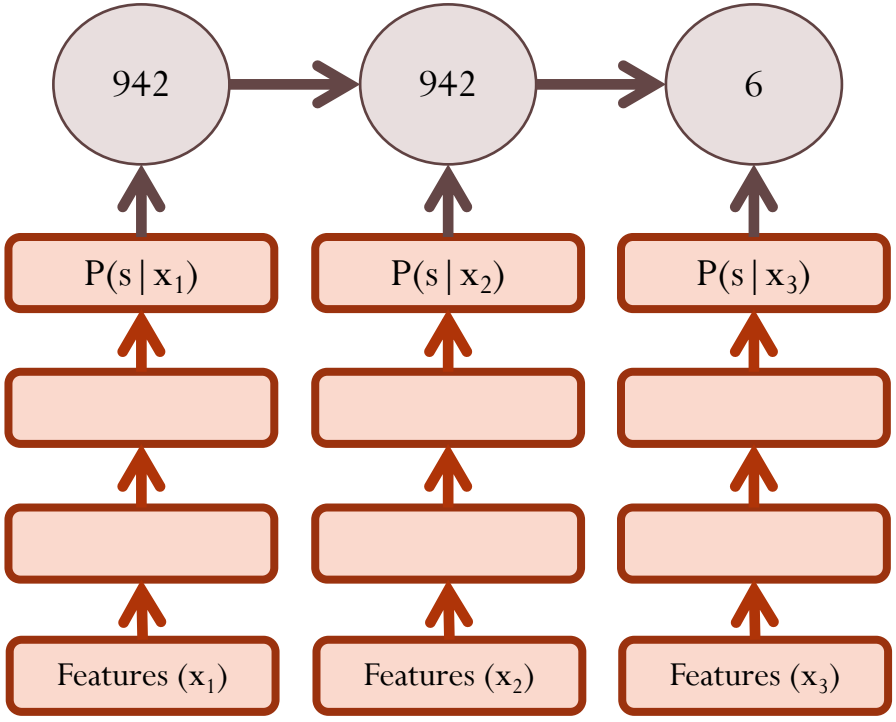
DNN Hybrid Acoustic Models

Transcription: Samson

Pronunciation: S – AE – M – S –AH – N

Sub-phones : 942 – 6 – 37 – 8006 – 4422 …

Hidden Markov Model (HMM):



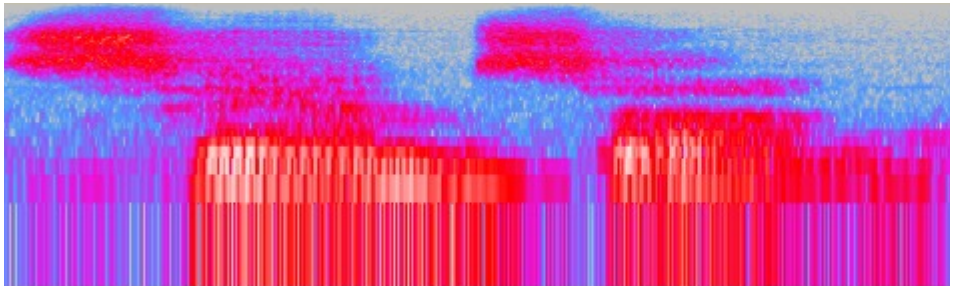
Acoustic Model:

Use a DNN to approximate: P(s|x)

Apply Bayes' Rule: P(x|s) = P(s|x) * P(x) / P(s)

DNN * Constant / State prior

Audio Input:



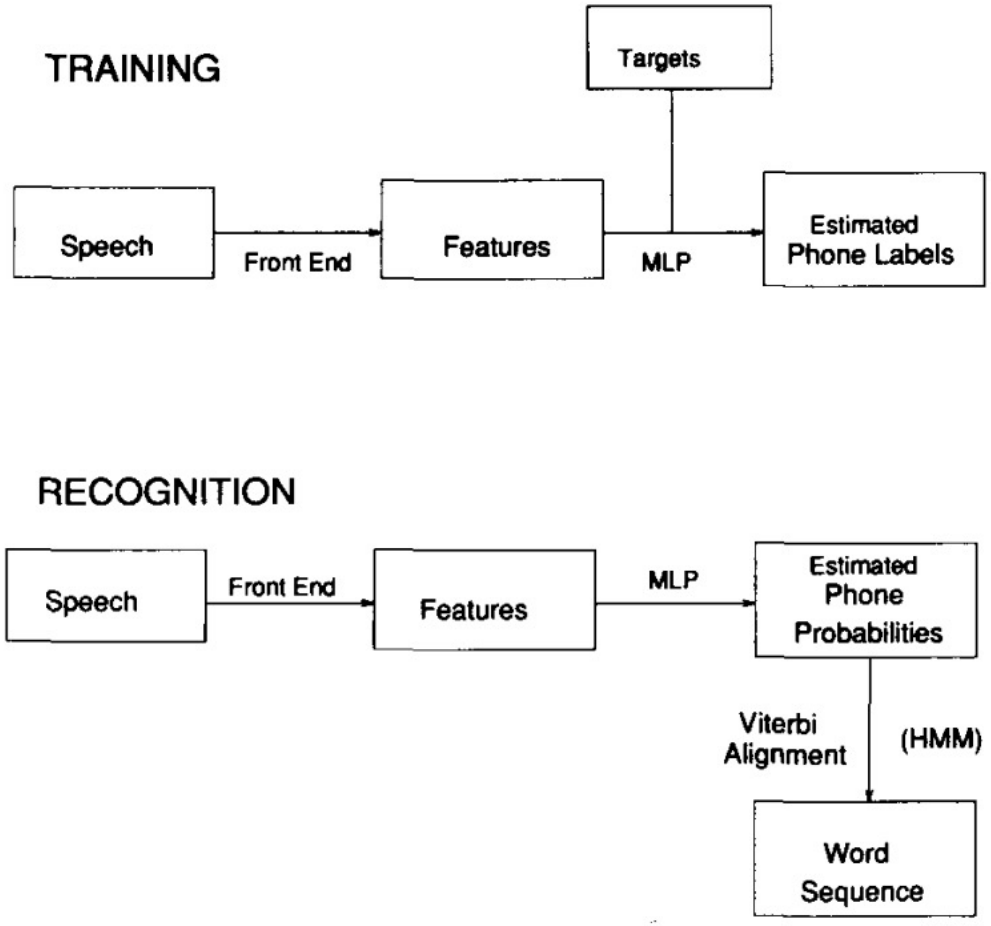
Objective Function for Learning

- Supervised learning, minimize our classification errors
- Standard choice: Cross entropy loss function
  - Straightforward extension of logistic loss for binary

$$Loss(x, y; W, b) = -\sum_{k=1}^{K} (y = k) \log f(x)_k$$

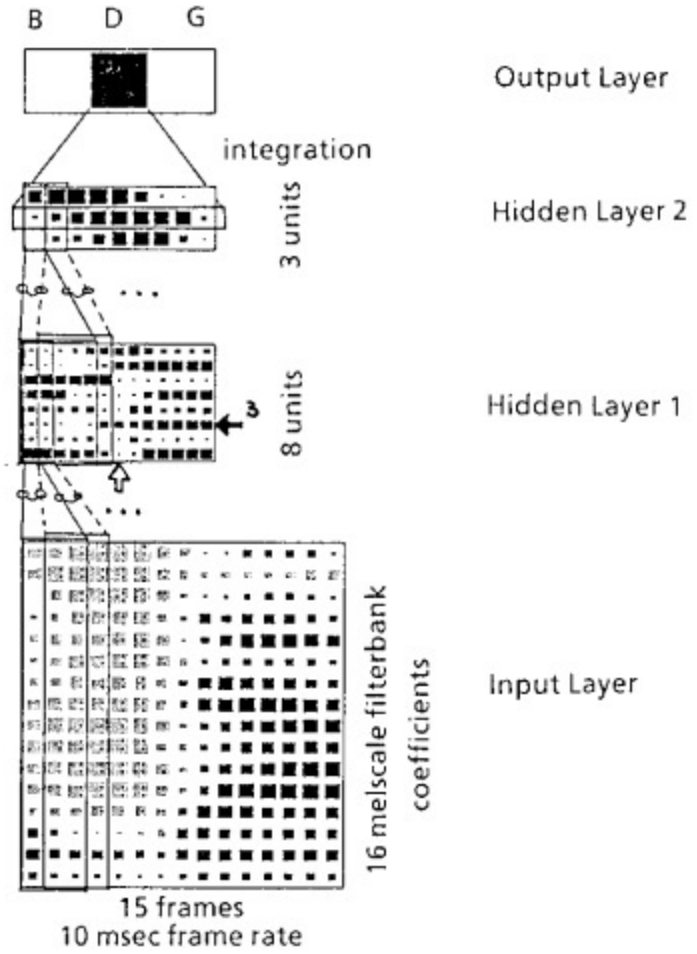
- This is a *frame-wise* loss. We use a label for each frame from a forced alignment
- Other loss functions possible. Can get deeper integration with the HMM or word error rate

Not Really a New Idea

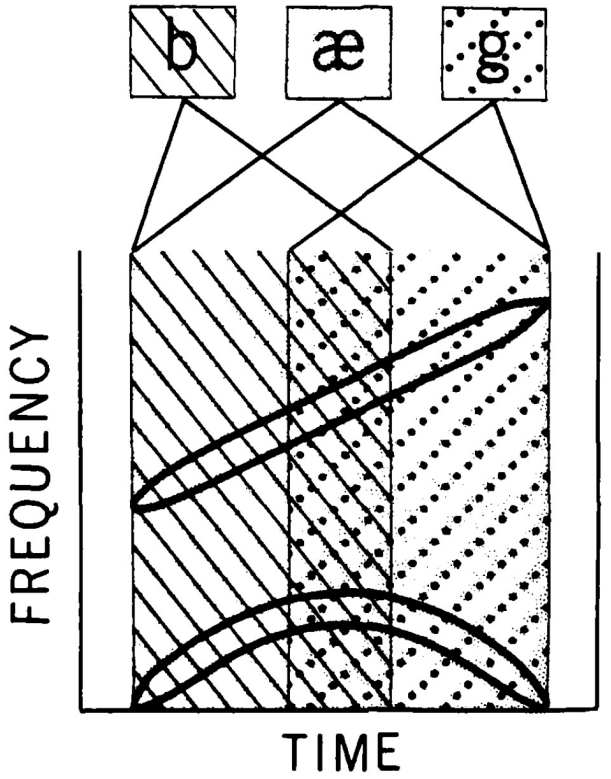


Renals, Morgan, Bourland, Cohen, & Franco. 1994.

Early neural network approaches



(Waibel, Hanazawa, Hinton, Shikano, & Lang. 1988)



(McClelland, & Elman. 1985)

Hybrid MLPs on Resource Management

TABLE I
RESULTS USING THE THREE TEST SETS WITH THE PERPLEXITY 60 WORDPAIR GRAMMAR. (CI-MLP is the context-independent MLP-HMM hybrid system, CD-HMM is the full context-dependent Decipher system, and the MIX system is a simple interpolation between the CD-HMM and the CI-MLP.)

	% error		
Test Set	CI-MLP	CD-HMM	MIX
Feb 91	5.8	3.8	3.2
Sep 92a	10.9	10.1	7.7
Sep 92b	9.5	7.0	5.7

TABLE II
RESULTS USING THE THREE TEST SETS USING NO GRAMMAR (PERLPEXITY 991)

	% error		
Test Set	CI-MLP	CD-HMM	MIX
Feb 91	24.7	19.3	15.9
Sep 92a	31.5	29.2	25.4
Sep 92b	30.9	26.6	21.5

Renals, Morgan, Bourland, Cohen, & Franco. 1994.

Hybrid Systems now Dominate ASR

[TABLE 3] A COMPARISON OF THE PERCENTAGE WERs USING DNN-HMMs AND GMM-HMMs ON FIVE DIFFERENT LARGE VOCABULARY TASKS.

TASK	HOURS OF TRAINING DATA	DNN-HMM	GMM-HMM WITH SAME DATA	GMM-HMM WITH MORE DATA
SWITCHBOARD (TEST SET 1)	309	18.5	27.4	18.6 (2,000 H)
SWITCHBOARD (TEST SET 2)	309	16.1	23.6	17.1 (2,000 H)
ENGLISH BROADCAST NEWS	50	17.5	18.8	
BING VOICE SEARCH (SENTENCE ERROR RATES)	24	30.4	36.2	
GOOGLE VOICE INPUT	5,870	12.3		16.0 (>> 5,870 H)
YOUTUBE	1,400	47.6	52.3	

Hinton et al. 2012.

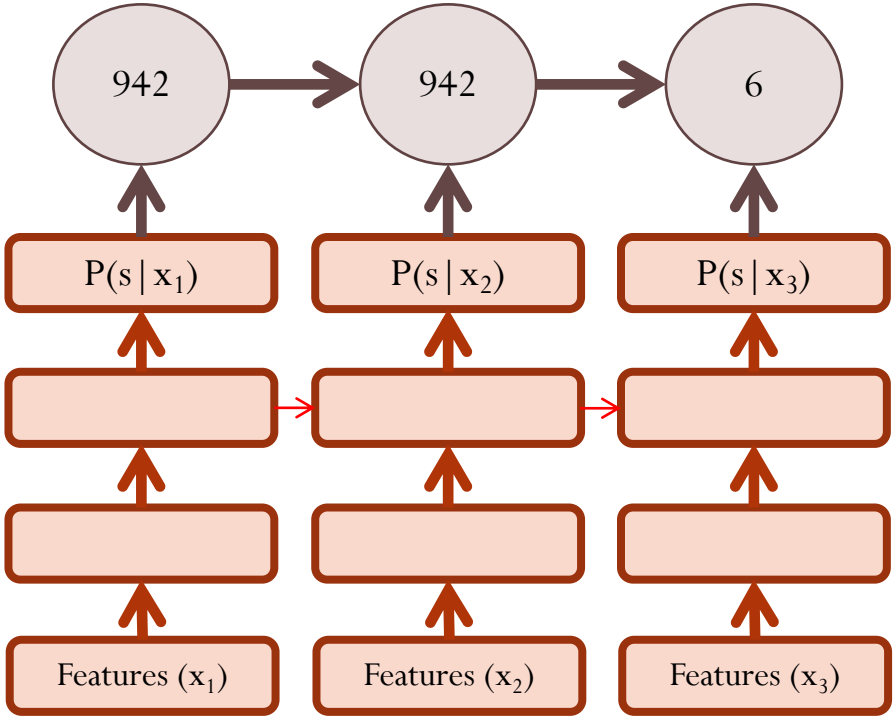
What's Different in Modern DNNs?

- Context-dependent HMM states
- Deeper nets improve on single hidden layer nets
- Hidden unit nonlinearity
- Many more model parameters (scaling up)
- Specific depth (e.g. 3 vs 7 hidden layers)
- Fast computers = run many experiments
- Architecture choices (easiest is replacing sigmoid)
- Pre-training *does not matter**. Initially we thought this was the new trick that made things work

Recurrent DNN Hybrid Acoustic Models

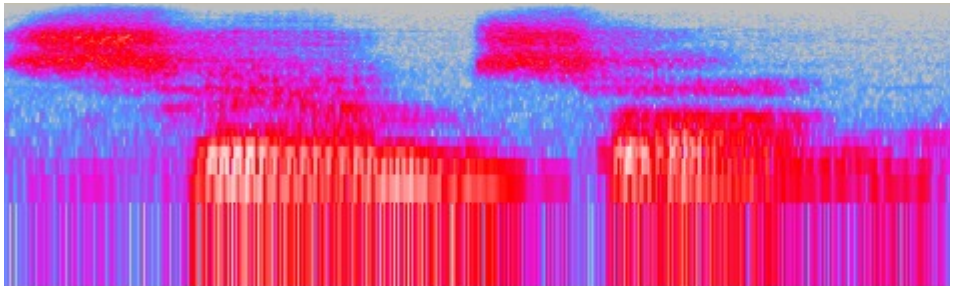
Transcription: Samson
Pronunciation: S – AE – M – S –AH – N
Sub-phones : 942 – 6 – 37 – 8006 – 4422 …

Hidden Markov Model (HMM):

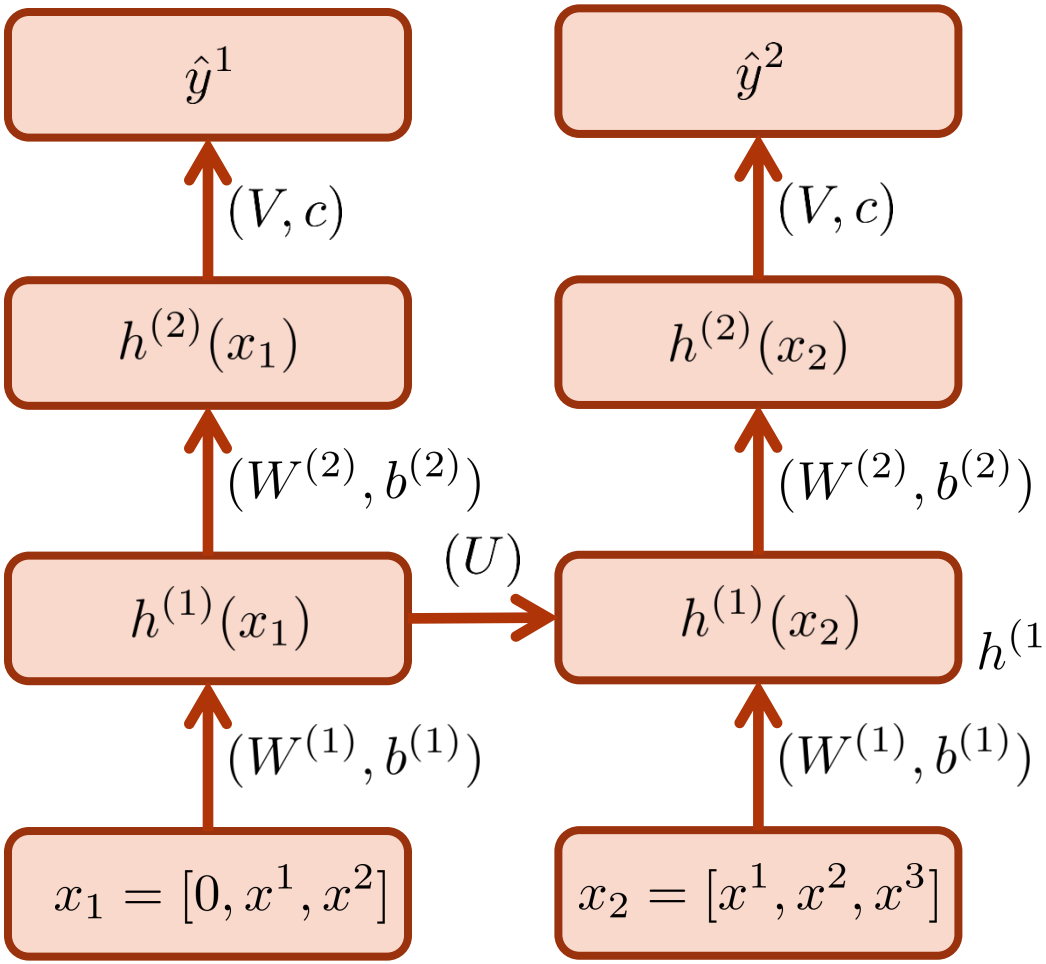


Acoustic Model:

Audio Input:



Deep Recurrent Network



Output Layer

$$\hat{y}^2 = Vh^{(1)}(x_2) + c$$

Hidden Layer

$$h^{(2)}(x_2) = \sigma(W^{(2)}h^{(1)}(x_2) + b^{(2)})$$

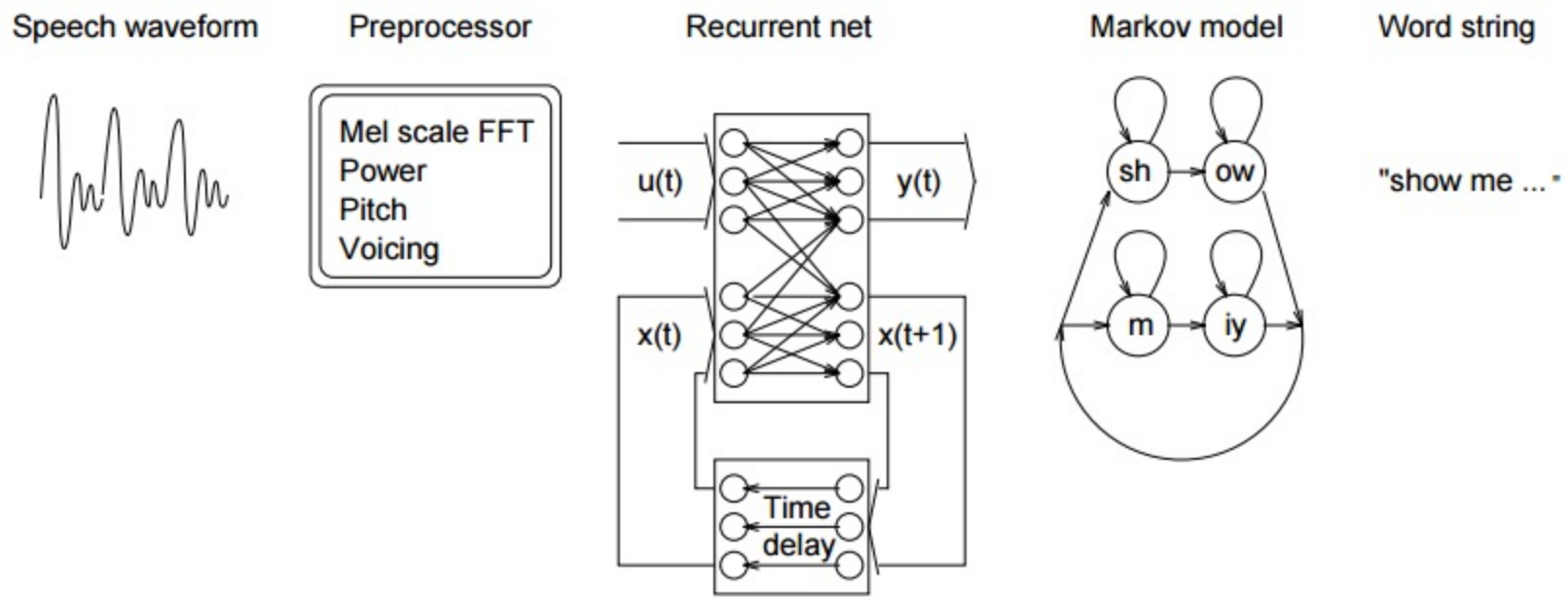
Hidden Layer

$$h^{(1)}(x_2) = \sigma(W^{(1)}x_2 + b^{(1)} + Uh^{(1)}(x_1))$$

Input

$\mathbf{X}$

RNNs for acoustic modeling in 1996



(Robinson, Hochberg, & Renals. 1996)

Adding More Parameters 20 Years Ago

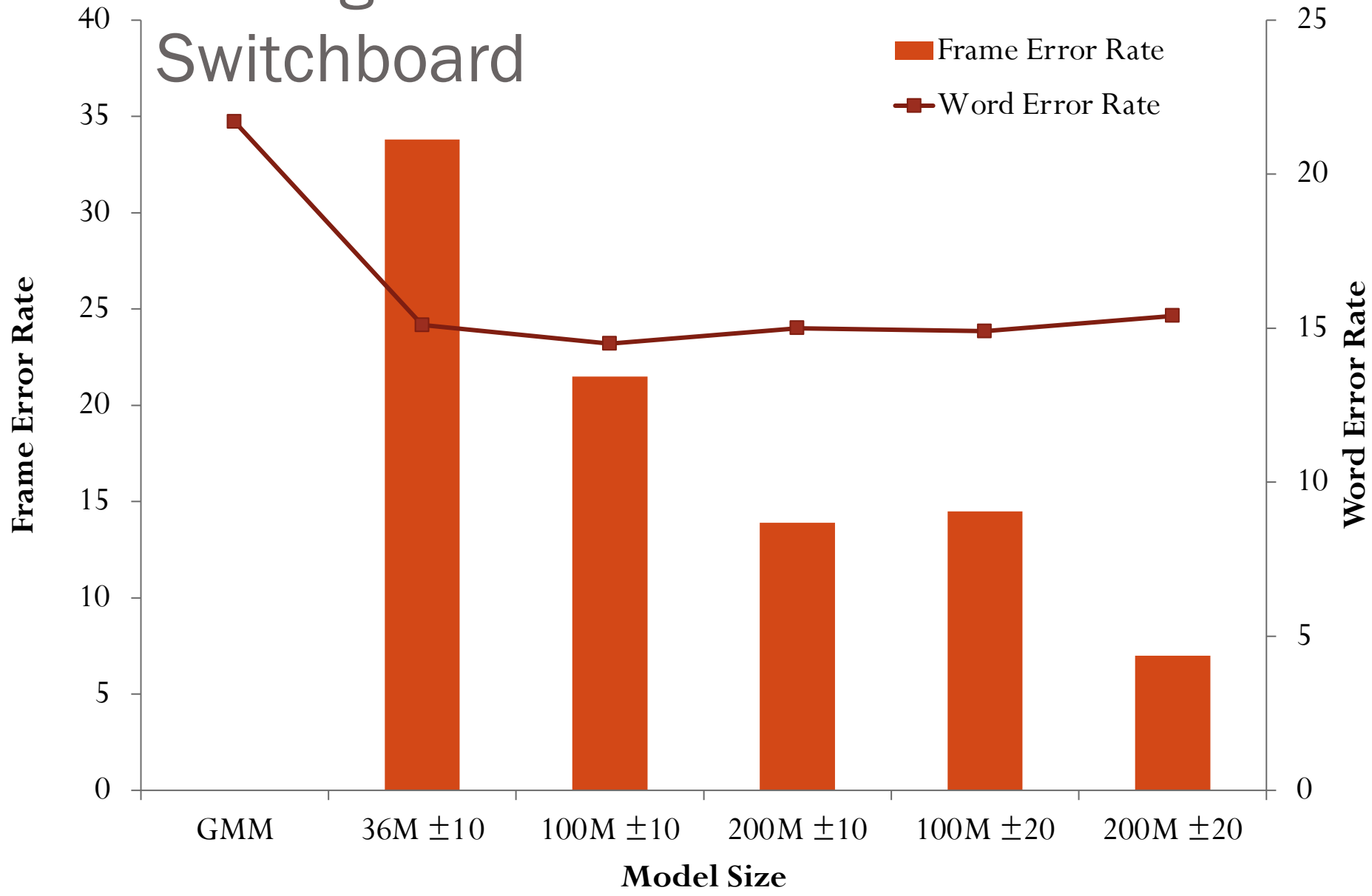
Size matters: An empirical study of neural network training for LVCSR. Ellis & Morgan. ICASSP. 1999.

Hybrid NN. 1 hidden layer. 54 HMM states.

74hr broadcast news task

"…improvements are almost always obtained by increasing either or both of the amount of training data or the number of network parameters … We are now planning to train an 8000 hidden unit net on 150 hours of data … this training will require over three weeks of computation."

Scaling Total Parameters on Switchboard

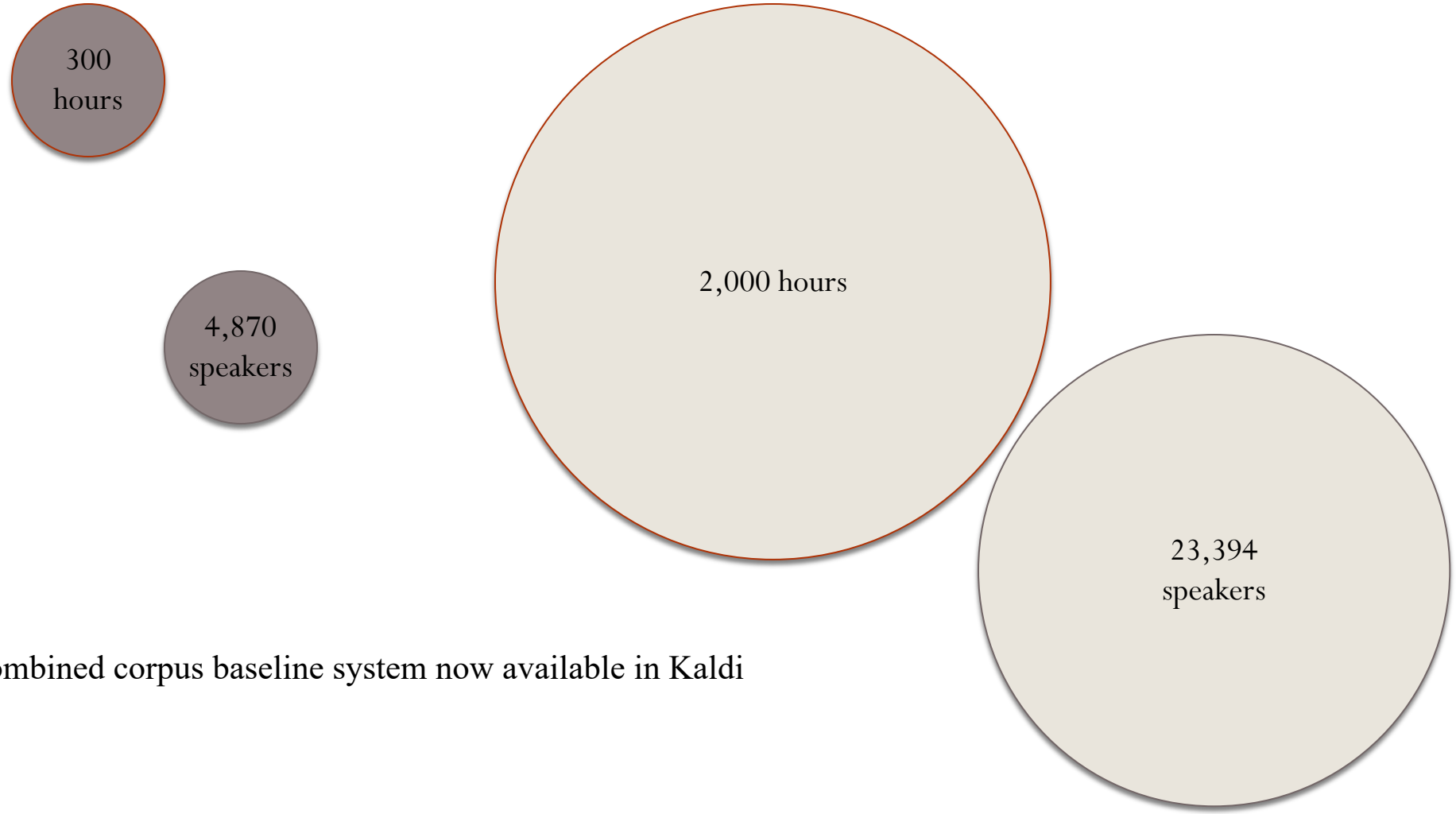


(Maas, Qi, Xie, Hannun, Lengerich, Jurafsky, & Ng. 2017)

Combining Speech Corpora

Switchboard

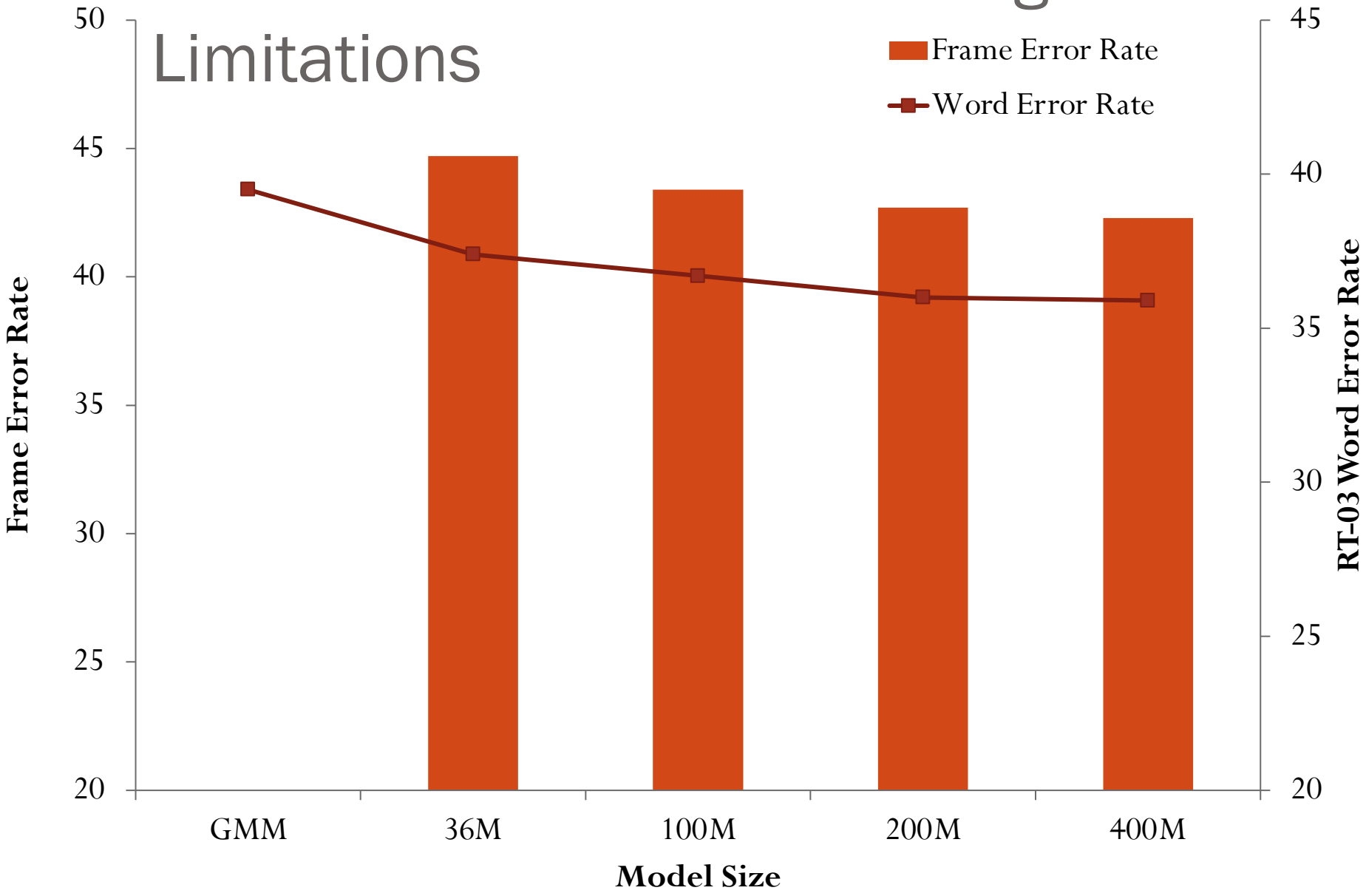
Fisher



Combined corpus baseline system now available in Kaldi

(Maas, Qi, Xie, Hannun, Lengerich, Jurafsky, & Ng. In

Framework + Isolated Training Limitations



(Maas, Qi, Xie, Hannun, Lengerich, Jurafsky, & Ng. 2017)

HMM-Free Recognition

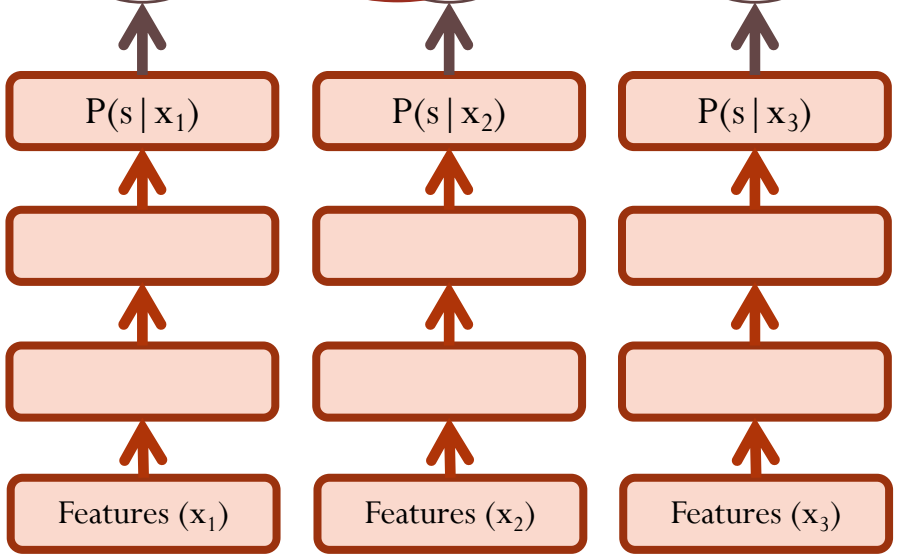
Transcription: Samson

Pronunciation: S – AE – M … N

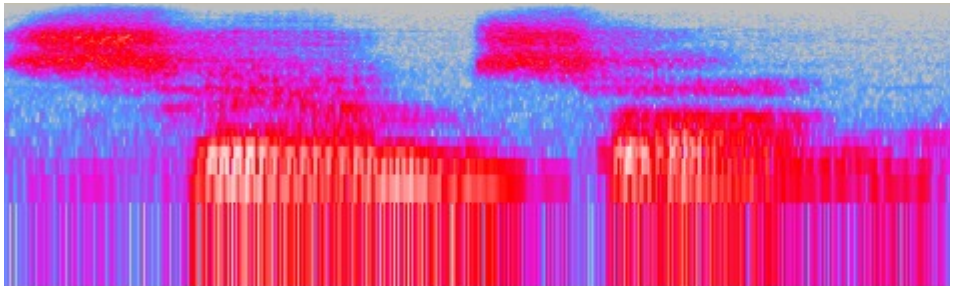
Sub-phones : 942 – 6 … 8006 … 22 …

Hidden Markov Model (HMM):

Acoustic Model:



Audio Input:



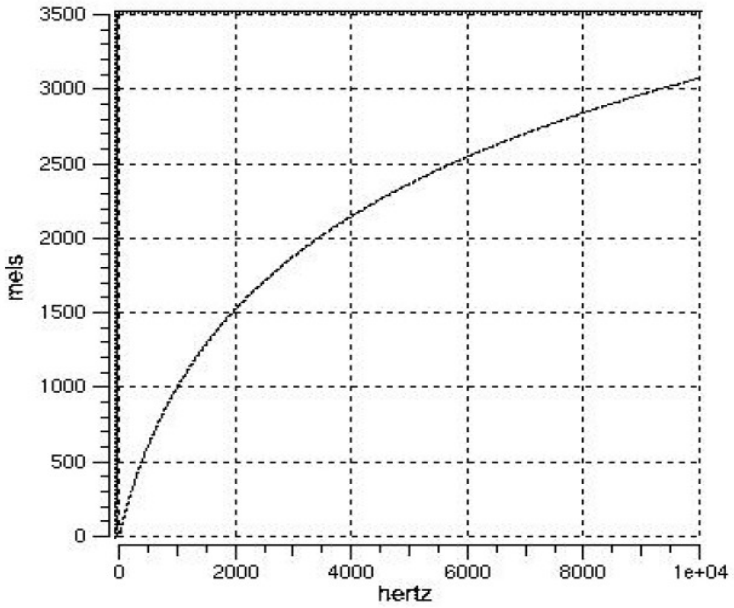
(Graves & Jaitly. 2014)

Outline for Today

- HMM-GMM acoustic models
  - Mixtures of Gaussians
  - Subphone states
- HMM-DNN models
- **Feature extraction & data augmentation**

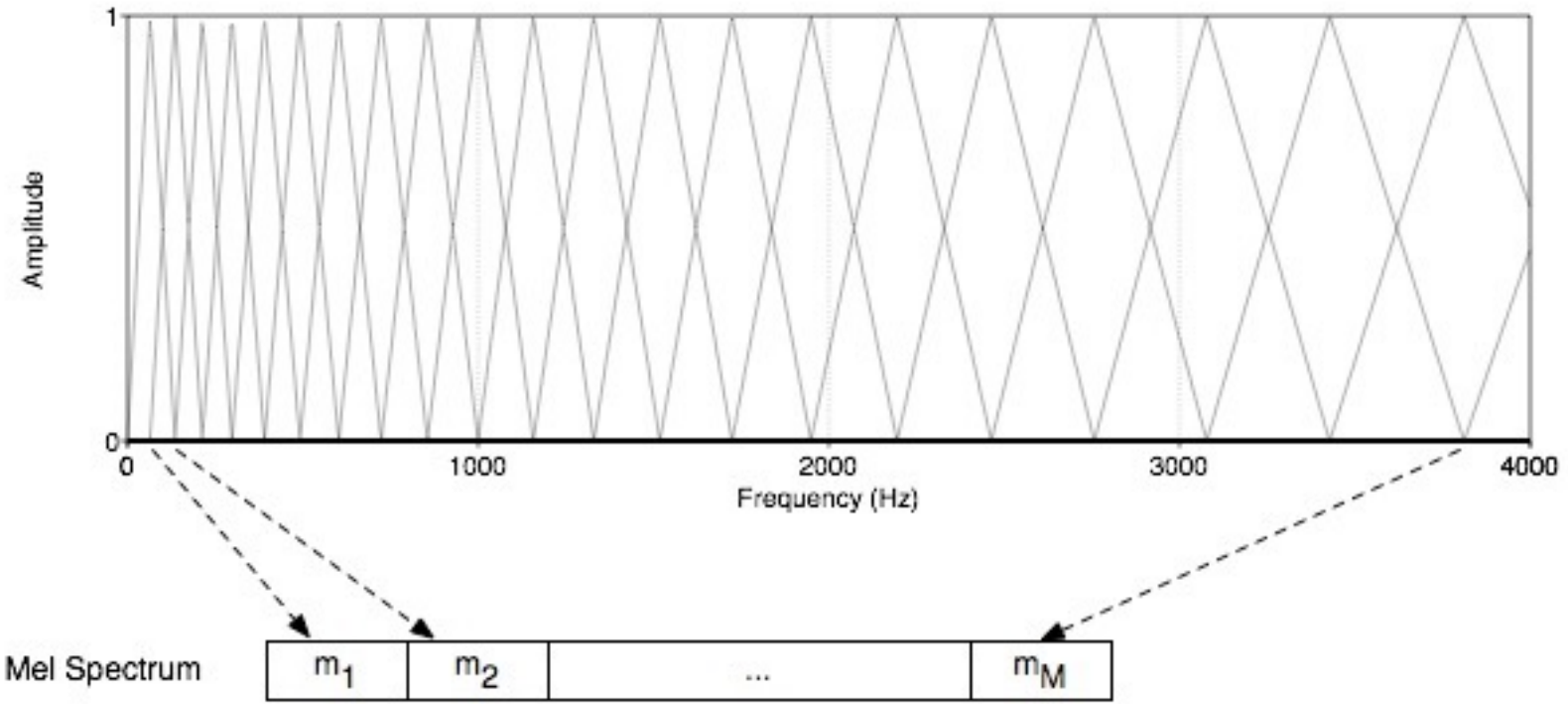
Mel-scale

- Human hearing is not equally sensitive to all frequency bands
- Less sensitive at higher frequencies, roughly > 1000 Hz
- I.e. human perception of frequency is non-linear:



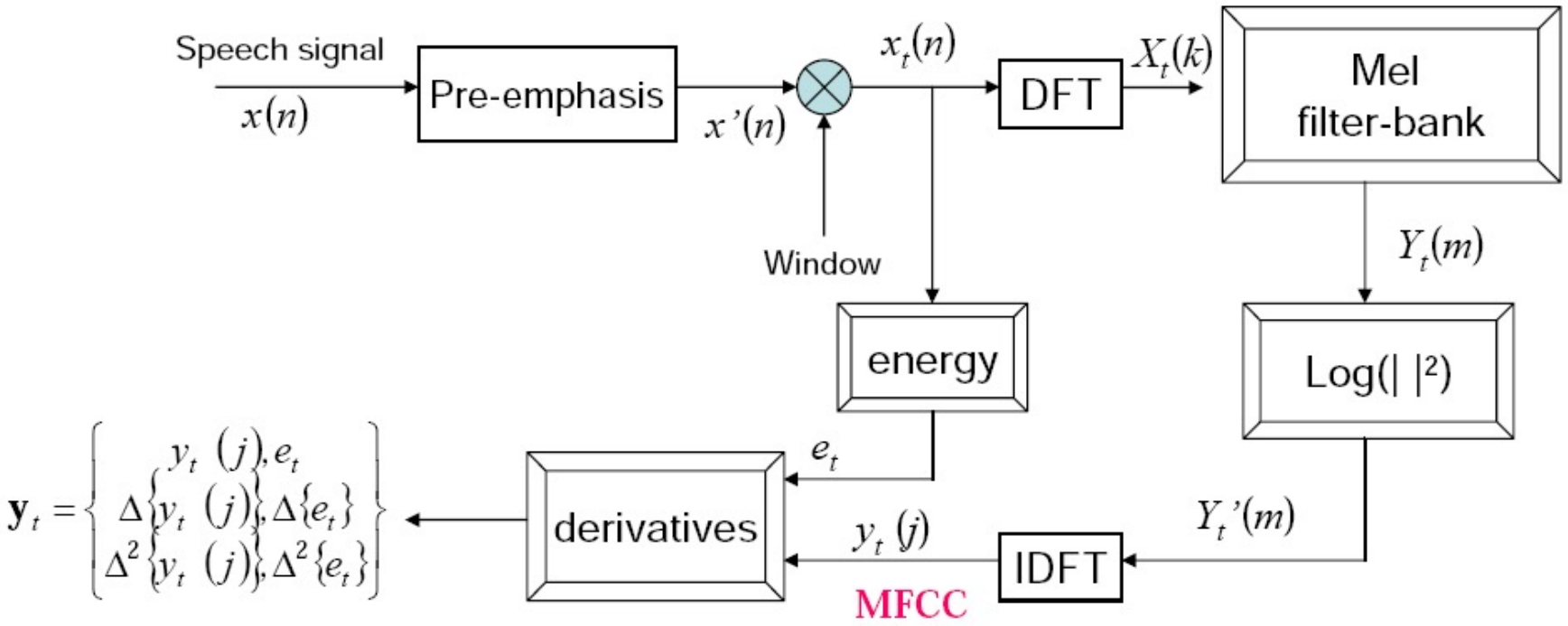
Mel Filter Bank Processing

- Mel Filter bank
  - Roughly uniformly spaced before 1 kHz
  - logarithmic scale after 1 kHz



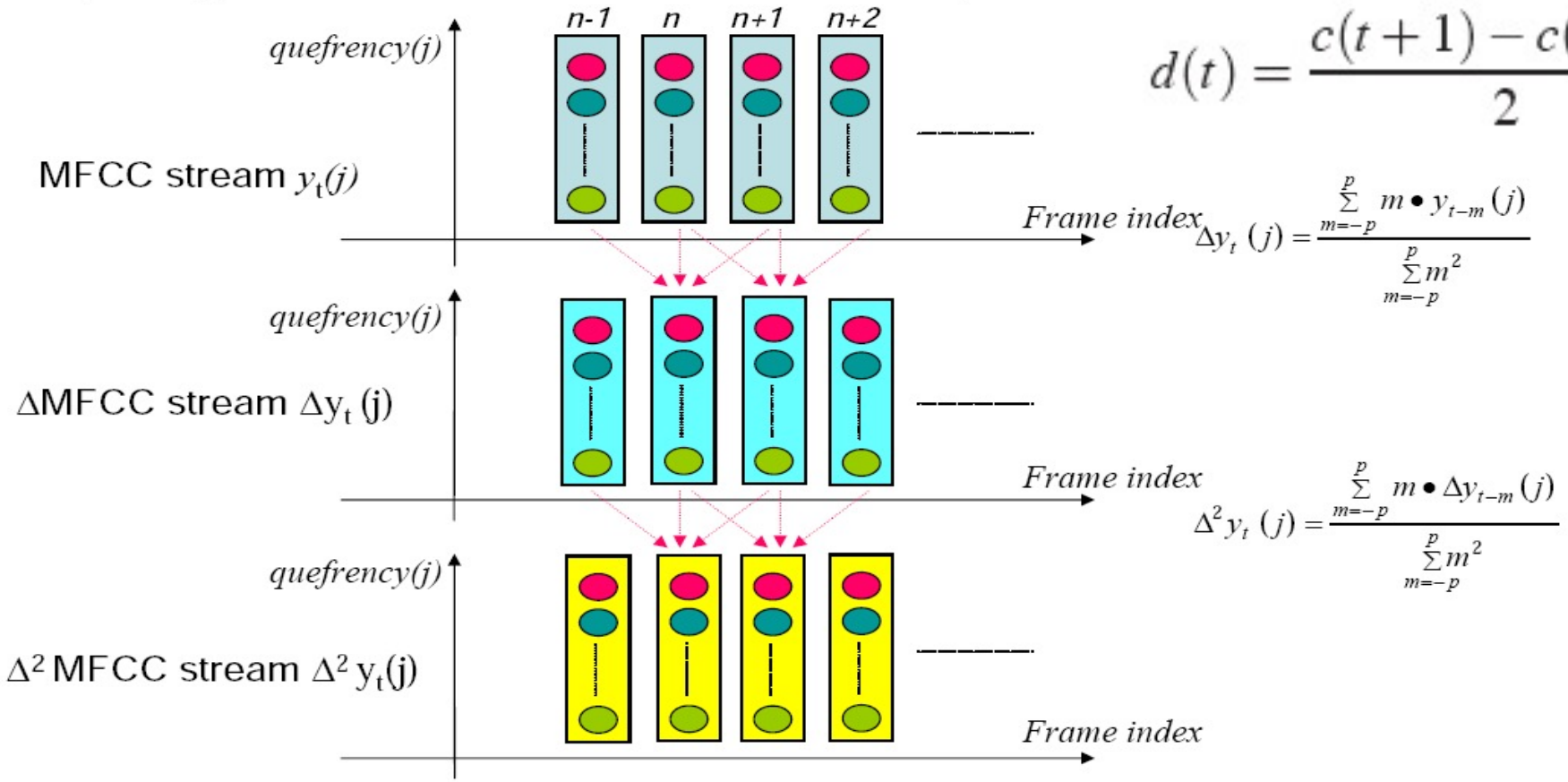
MFCC

- Mel-Frequency Cepstral Coefficient (MFCC)
  - Most widely used spectral representation in ASR



Delta and double-delta

- Derivative: in order to obtain temporal information



Typical MFCC features

- Window size: 25ms
- Window shift: 10ms
- Pre-emphasis coefficient: 0.97
- MFCC:
  - 12 MFCC (mel frequency cepstral coefficients)
  - 1 energy feature
  - 12 delta MFCC features
  - 12 double-delta MFCC features
  - 1 delta energy feature
  - 1 double-delta energy feature
- Total 39-dimensional features

Training Augmentation: SpecAugment

1. Time warping (image warp)
2. Frequency masking
3. Time masking

Mix using different policies

Implementations available for PyTorch

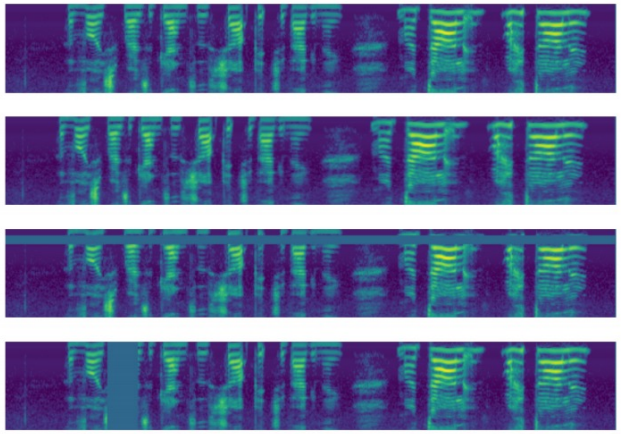


Figure 1: *Augmentations applied to the base input, given at the top. From top to bottom, the figures depict the log mel spectrogram of the base input with no augmentation, time warp, frequency masking and time masking applied.*

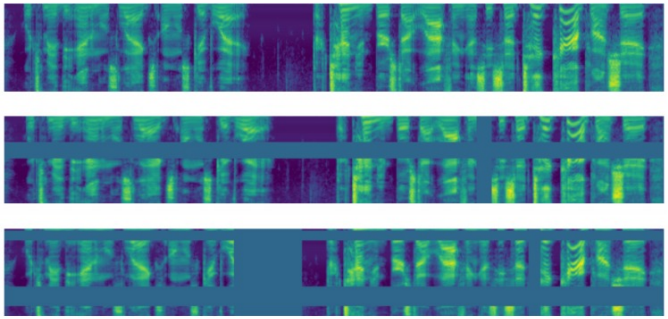


Figure 2: *Augmentation policies applied to the base input. From top to bottom, the figures depict the log mel spectrogram of the base input with policies None, LB and LD applied.*

(*Park et al*, ICASSP 2020)

Training Augmentation: SpecAugment

Table 5: *Switchboard 300h WERs (%).*

Method	No LM		With LM	
	SWBD	CH	SWBD	CH
HMM				
Veselý et al., (2013) [41]			12.9	24.5
Povey et al., (2016) [30]			9.6	19.3
Hadian et al., (2018) [42]			9.3	18.9
Zeyer et al., (2018) [24]			8.3	17.3
CTC				
Zweig et al., (2017) [43]	24.7	37.1	14.0	25.3
Audhkhasi et al., (2018) [44]	20.8	30.4		
Audhkhasi et al., (2018) [45]	14.6	23.6		
LAS				
Lu et al., (2016) [46]	26.8	48.2	25.8	46.0
Toshniwal et al., (2017) [47]	23.1	40.8		
Zeyer et al., (2018) [24]	13.1	26.1	11.8	25.7
Weng et al., (2018) [48]	12.2	23.3		
Zeyer et al., (2018) [38]	11.9	23.7	11.0	23.1
Our Work				
LAS	11.2	21.6	10.9	19.4
LAS + SpecAugment (SM)	**7.2**	14.6	**6.8**	14.1
LAS + SpecAugment (SS)	7.3	**14.4**	7.1	**14.0**

(*Park et al*, ICASSP 2020)

Appendix

Why is MFCC so popular?

- Efficient to compute
- Incorporates a perceptual Mel frequency scale
- Separates the source and filter
- IDFT(DCT) de-correlates the features
  - Necessary for diagonal assumption in GMM modeling
- There are alternatives like PLP
- Choice matters less for neural network acoustic models

Acoustic Model Adaptation

- Shift the means and variances of Gaussians to better match the input feature distribution
  - Maximum Likelihood Linear Regression (MLLR)
  - Maximum A Posteriori (MAP) Adaptation
- For both speaker adaptation and environment adaptation
- Widely used!

Maximum Likelihood Linear Regression (MLLR)

- Leggetter, C.J. and P. Woodland. 1995. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. Computer Speech and Language 9:2, 171-185.
- Given:
  - a trained AM
  - a small "adaptation" dataset from a new speaker
- Learn new values for the Gaussian mean vectors
  - Not by just training on the new data (too small)
  - But by learning a linear transform which moves the means.

Maximum Likelihood Linear Regression (MLLR)

- Estimates a linear transform matrix (W) and bias vector (ω) to transform HMM model means:

$$\mu_{new} = W_r \mu_{old} + \omega_r$$

- Transform estimated to maximize the likelihood of the adaptation data

Slide from Bryan Pellom

MLLR

- New equation for output likelihood

$$b_j(o_t) = \frac{1}{(2\pi)^{n/2} |\Sigma_j|^{1/2}} \exp\left(-\frac{1}{2} (o_t - (W\mu_j + \omega)) |\Sigma_j|^{-1} (o_t - (W\mu_j + \omega))^T \right)$$

MLLR

- Q: Why is estimating a linear transform from adaptation data different than just training on the data?
- A: Even from a very small amount of data we can learn 1 single transform for all triphones! So small number of parameters.
- A2: If we have enough data, we could learn more transforms (but still less than the number of triphones). One per phone (~50) is often done.

MLLR: Learning

- Given
  - a small labeled adaptation set (a couple sentences)
  - a trained AM
- Do forward-backward alignment on adaptation set to compute state occupation probabilities $\gamma_j(t)$.
- W can now be computed by solving a system of simultaneous equations involving $\gamma_j(t)$

MLLR performance on baby task (RM)

(Leggetter and Woodland 1995)

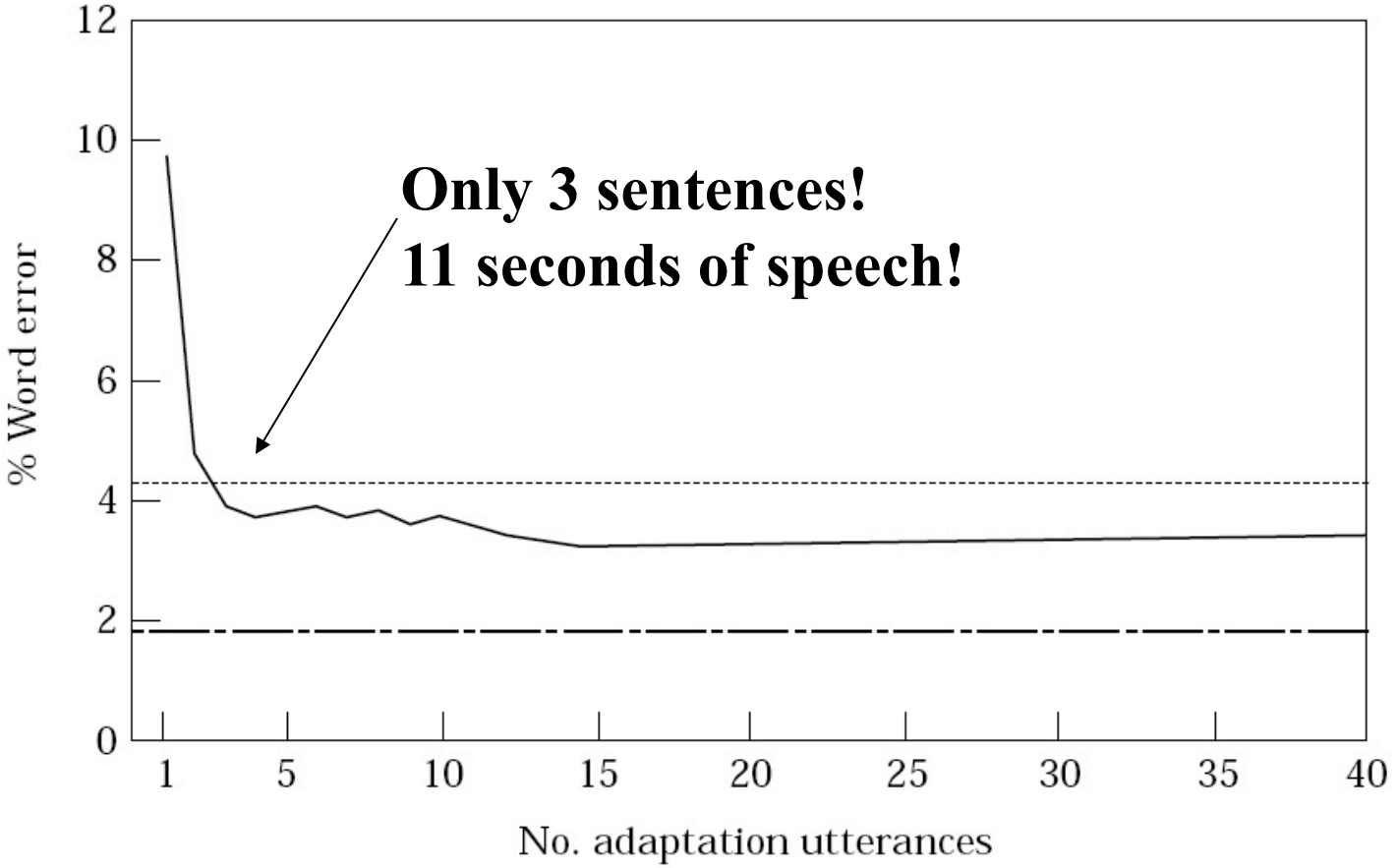


Figure 2. Full matrix maximum likelihood linear regression using global regression class. (·····), Speaker independent; (—-—-—-), speaker dependent; (——), speaker adapted.

MLLR doesn't need supervised adaptation set!

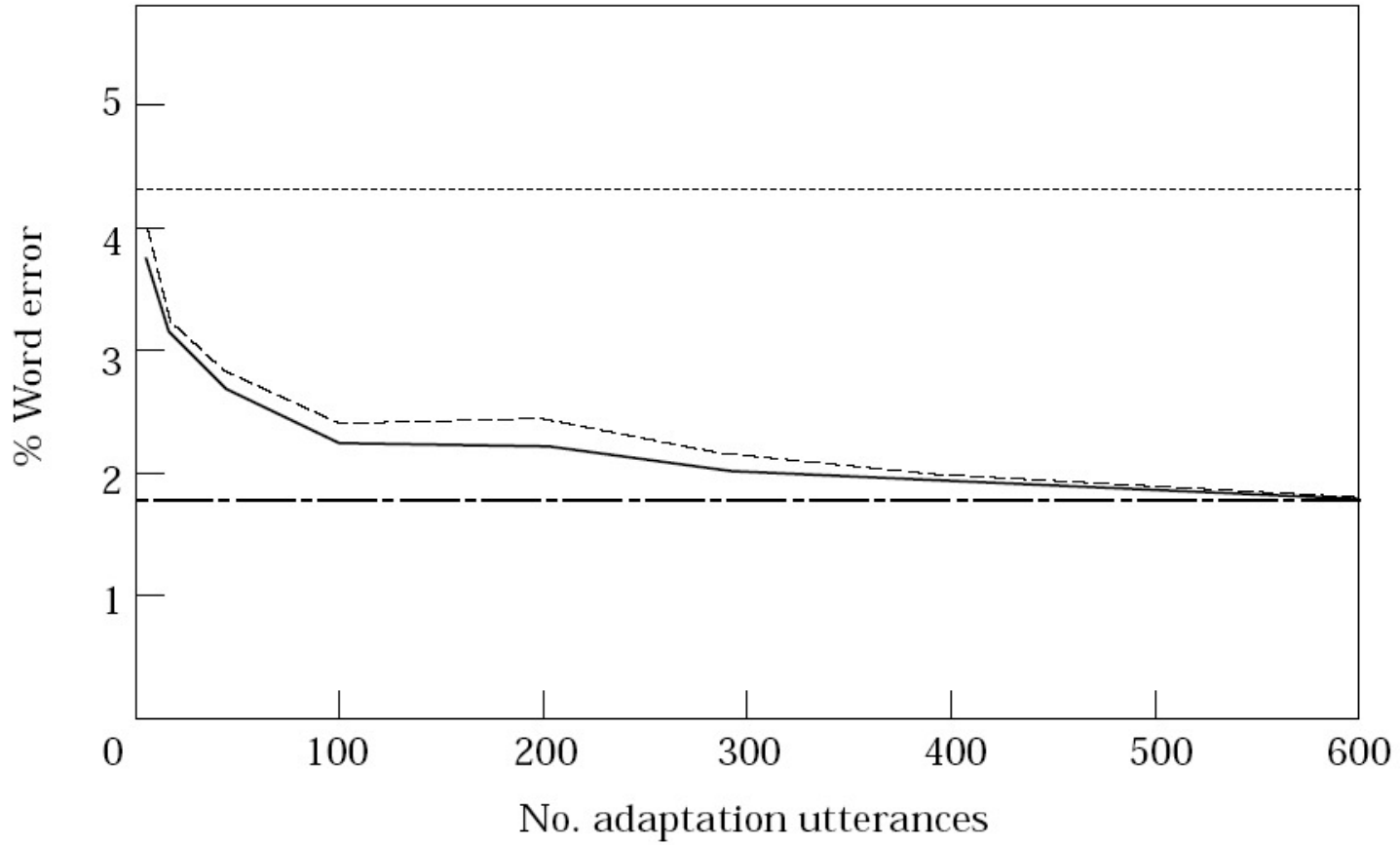


Figure 3. Supervised vs. unsupervised adaptation using maximum likelihood linear regression. (·····), Speaker independent; (—-—-—-), speaker dependent; (——), supervised adapted; (–––), unsupervised adapted.

Maximum A Posteriori Adaptation (MAP)

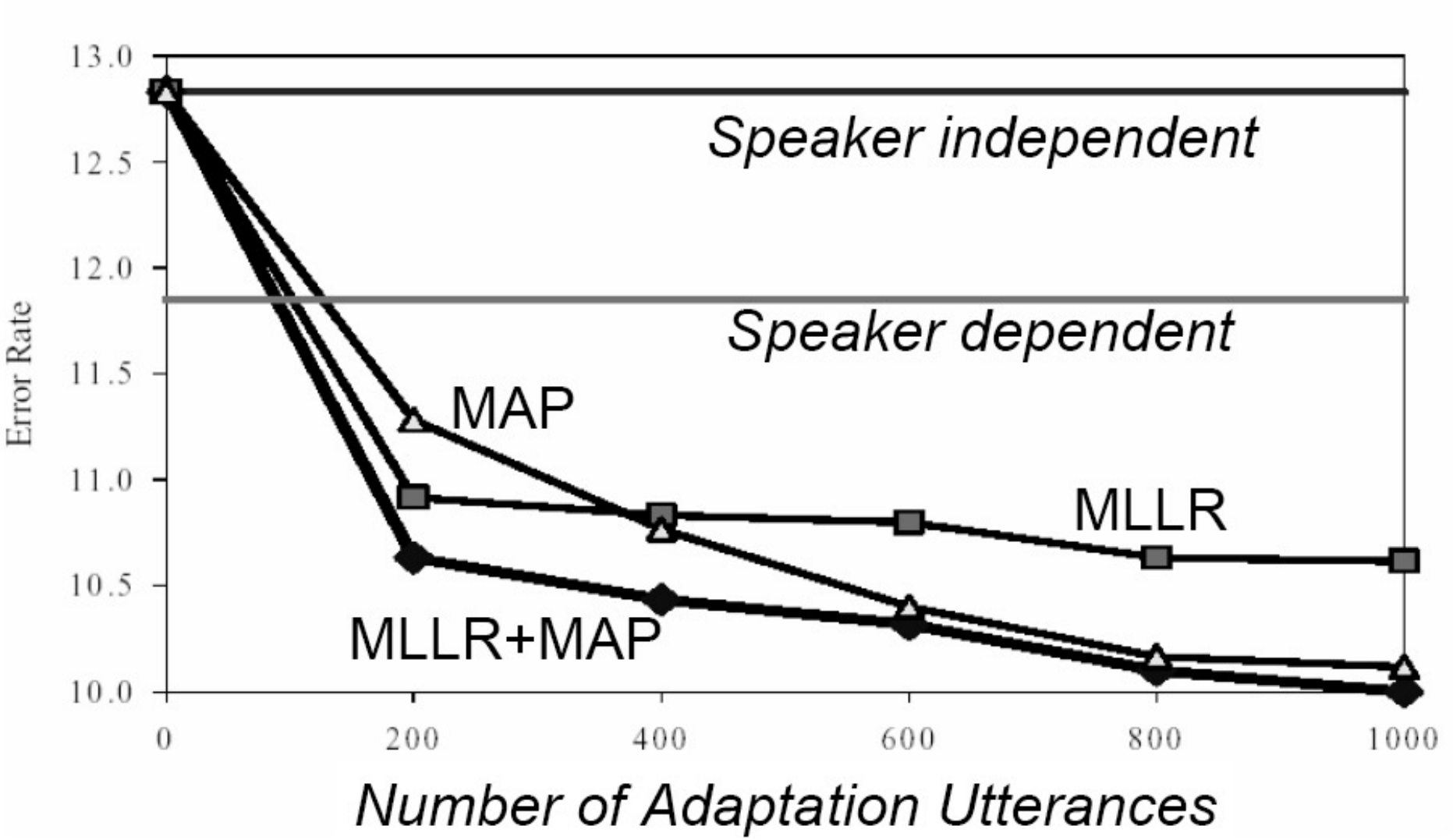
- **MAP Adaptation can only be applied Gaussians that are "seen" in the test data,**

$$\mu_{new} = \frac{\hat{N}}{\hat{N} + \alpha} \hat{m}_{obs} + \frac{\alpha}{\hat{N} + \alpha} \mu_{old}$$

$\hat{N}$	Number of frames of adaptation data
α	Weight for prior estimate of old mean
$\hat{m}_{obs}$	Mean vector of adaptation data assigned to Gauss.

Slide from Bryan Pellom

Performance of MLLR and MAP



Slide from Bryan Pellom after Huang et al

Summary

- MLLR: works on small amounts of adaptation data
- MAP: Maximum A Posterior Adaptation
  - Works well on large adaptation sets
- Acoustic adaptation techniques are quite successful at dealing with speaker variability
- If we can get 10 seconds with the speaker.

Discrete Representation of Signal

- Represent continuous signal into discrete form.

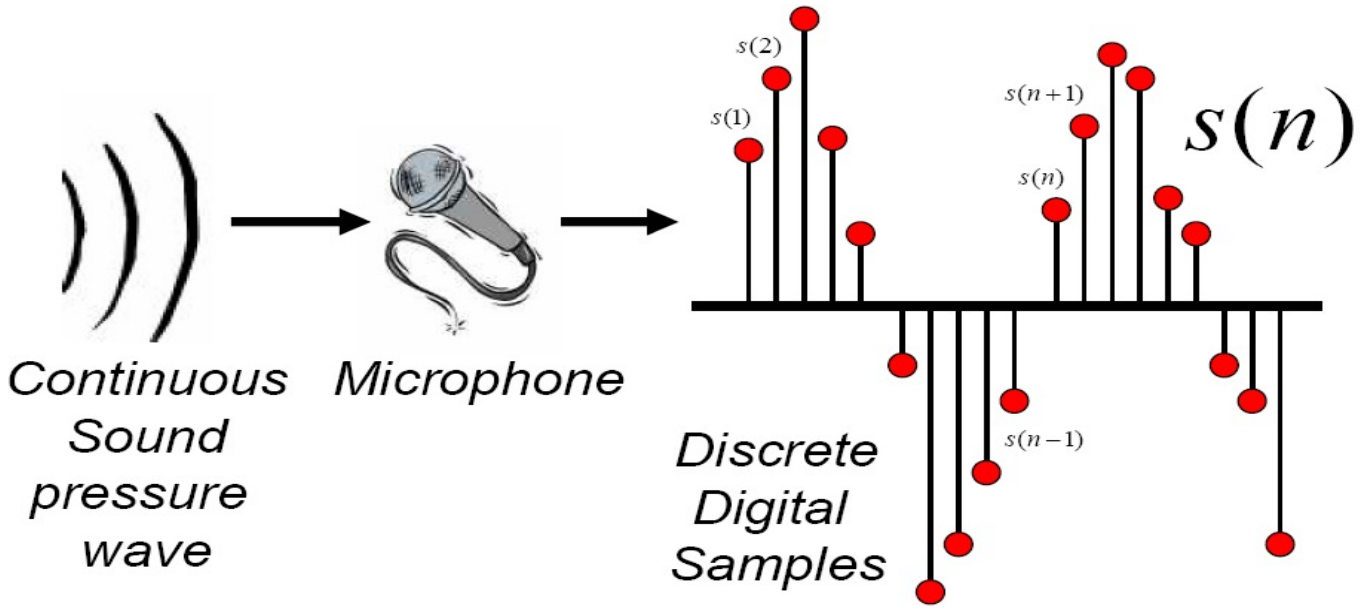
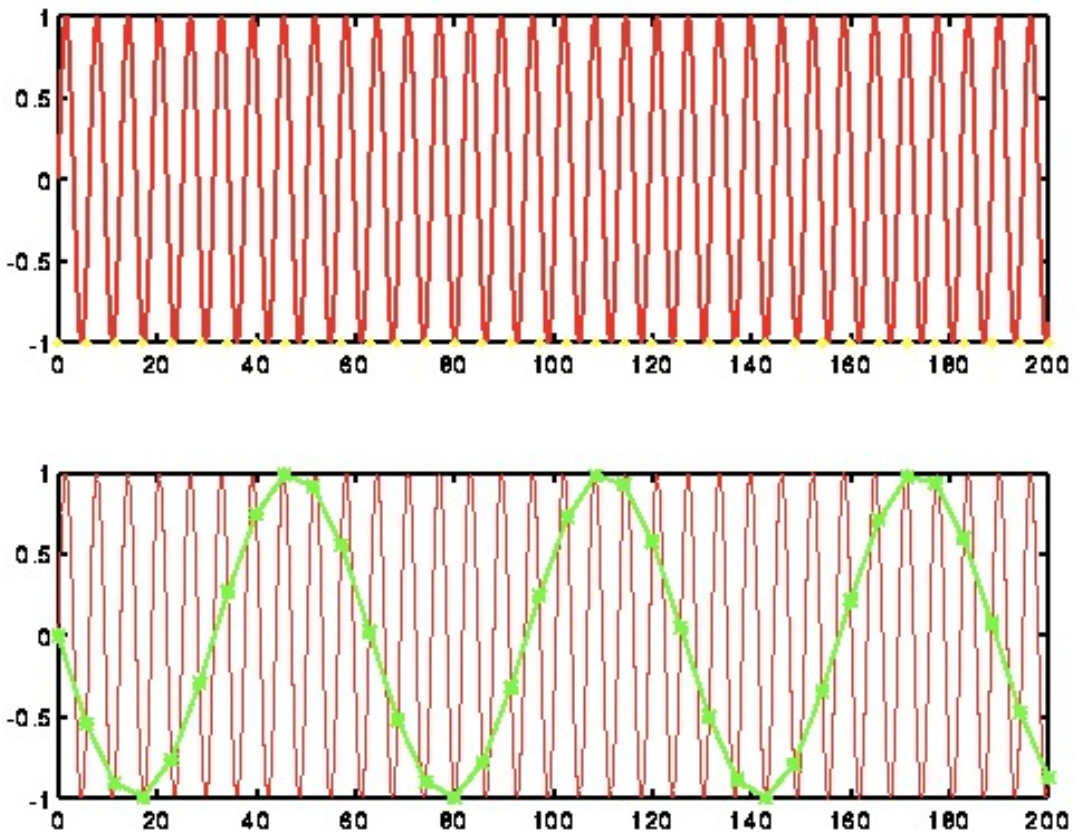


Image from Bryan Pellom

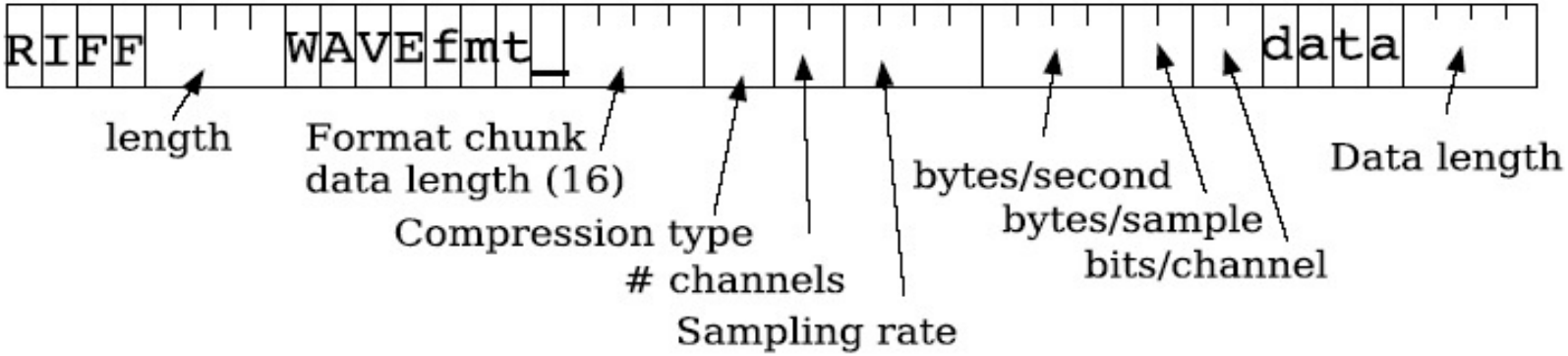
Sampling

If measure at green dots, will see a lower frequency wave and miss the correct higher frequency one!

Original signal in red:



WAV format



Many formats, trade-offs in compression, quality

Nice sound manipulation tool: Sox

http://sox.sourceforge.net/

convert speech formats

Windowing

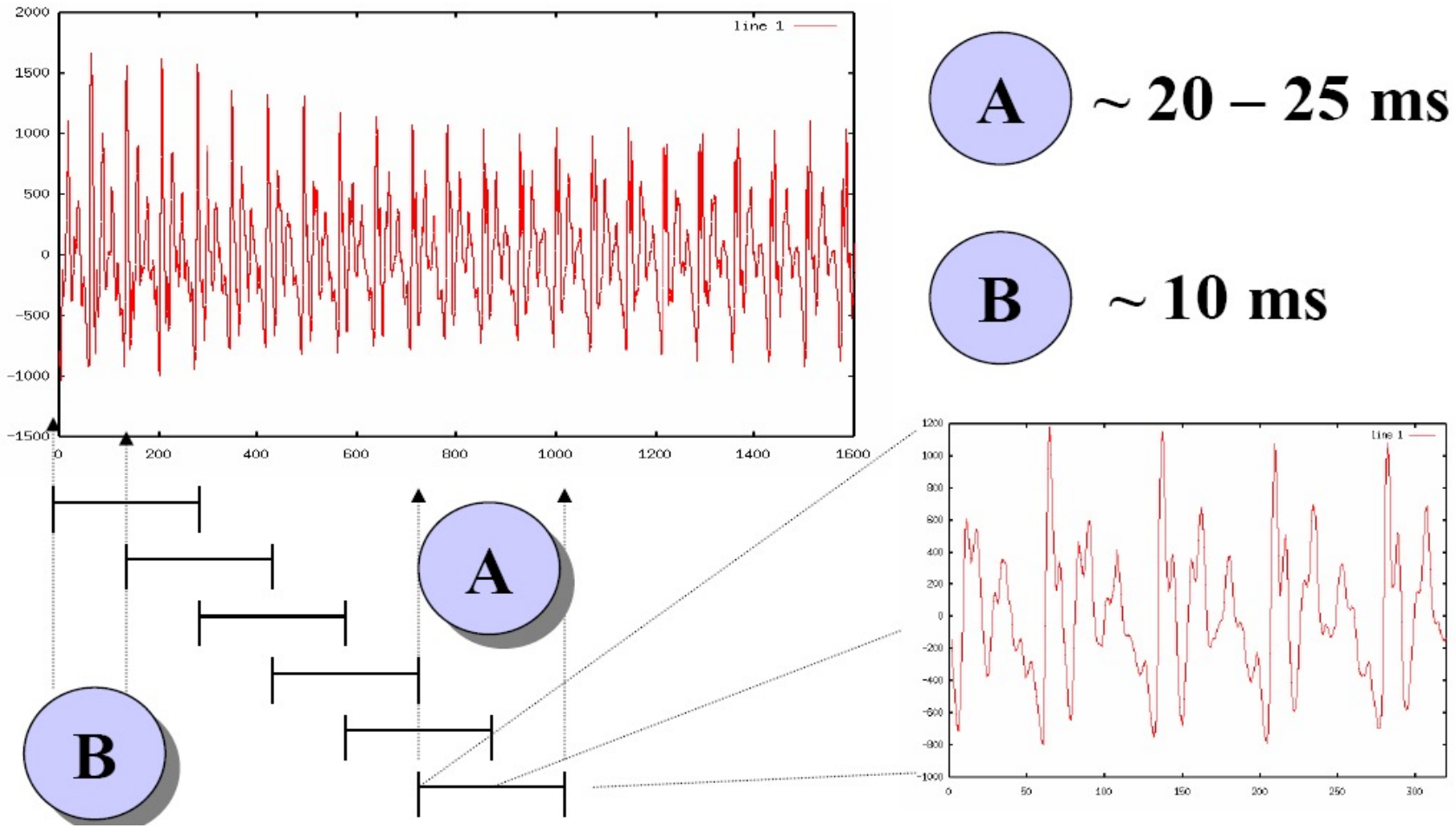
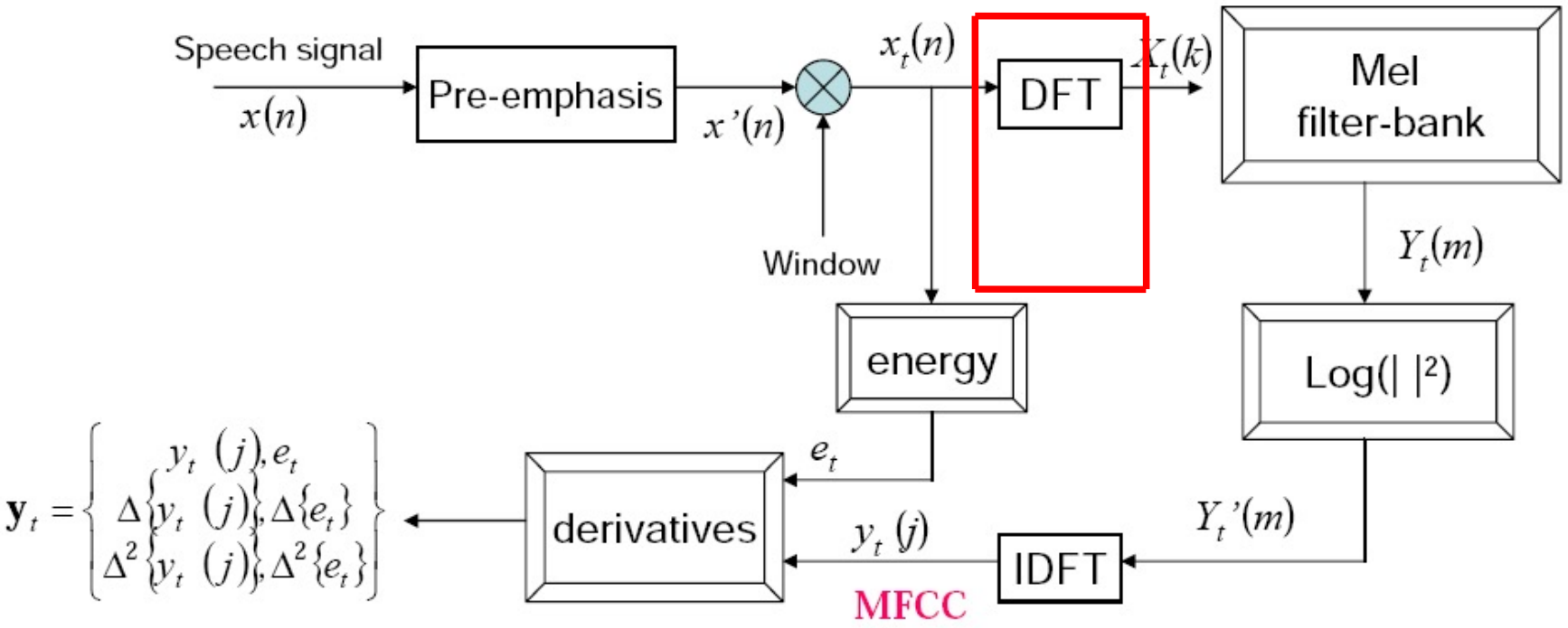


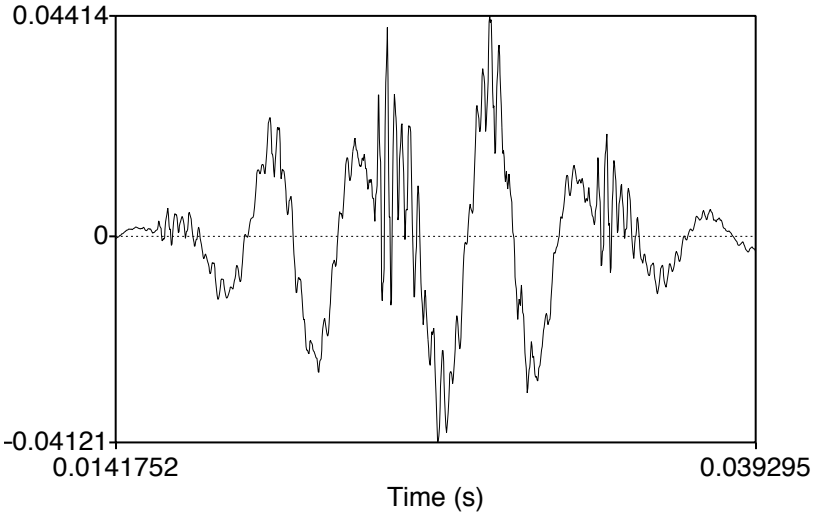
Image from Bryan Pellom

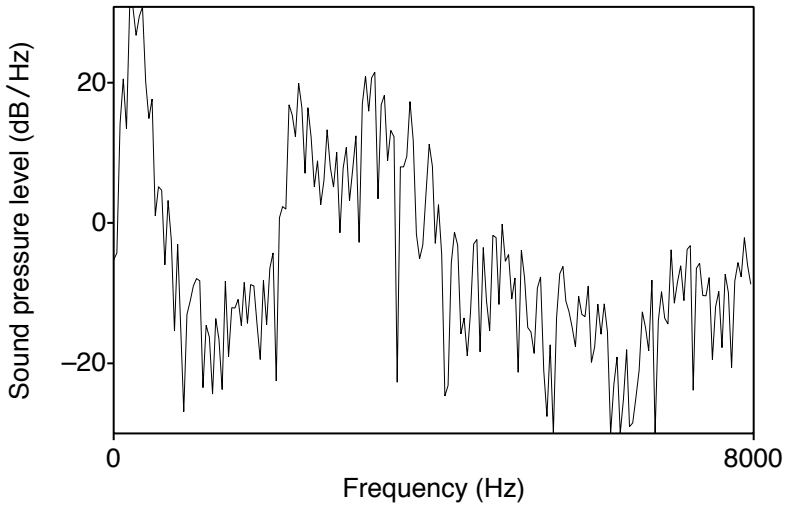
MFCC


Speech signal
$x(n)$
Pre-emphasis
$x'(n)$
$x_t(n)$
Window
DFT
$X_t(k)$
Mel filter-bank
$Y_t(m)$
Log(| |²)
$Y_t'(m)$
IDFT
$y_t(j)$
MFCC
energy
e_t
derivatives
$\mathbf{y}_t = \left\{ \begin{matrix} y_t(j), e_t \\ \Delta\{y_t(j)\}, \Delta\{e_t\} \\ \Delta^2\{y_t(j)\}, \Delta^2\{e_t\} \end{matrix} \right\}$

Discrete Fourier Transform computing a spectrum

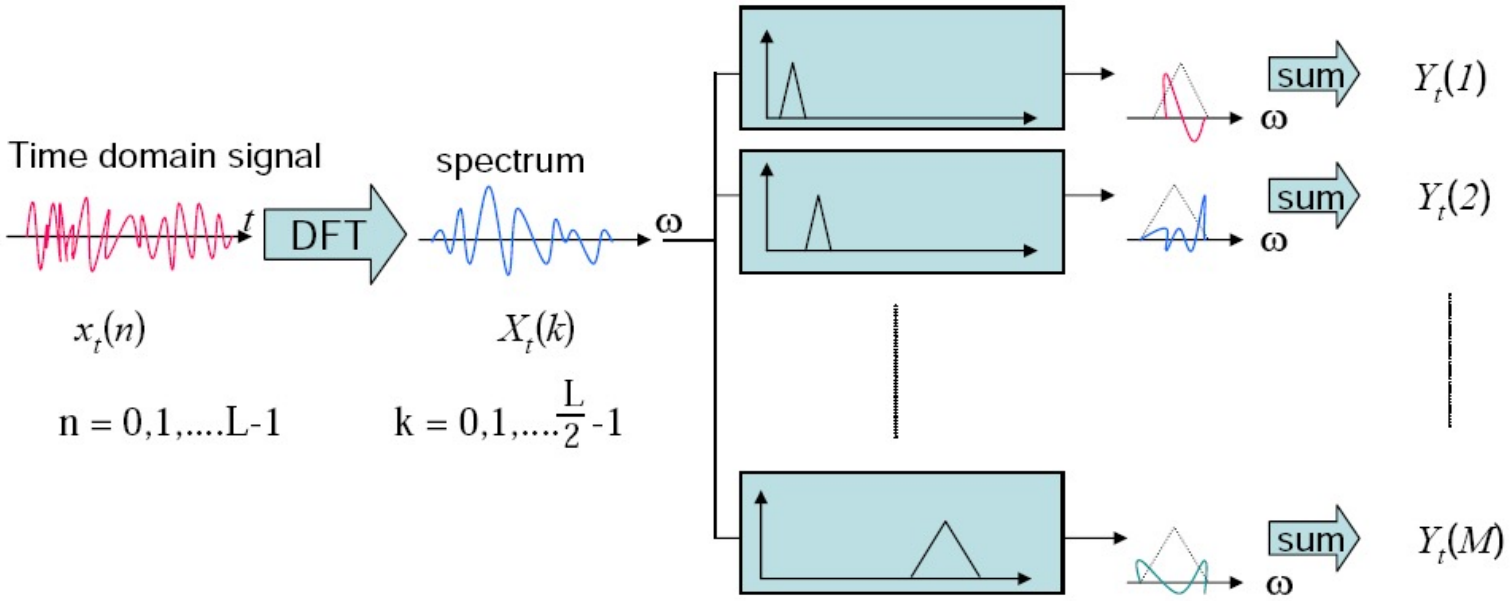
- A 25 ms Hamming-windowed signal from [iy]
  - And its spectrum as computed by DFT (plus other smoothing)



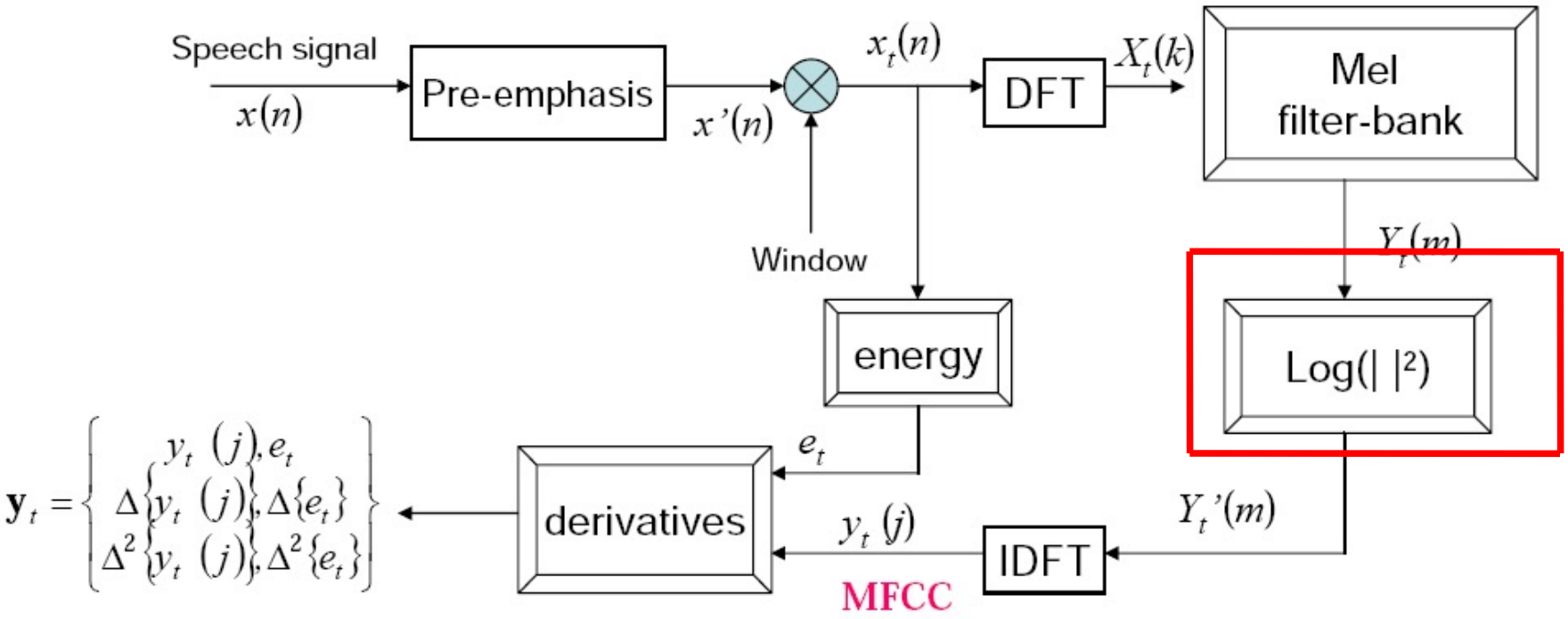


Mel-filter Bank Processing

- Apply the bank of Mel-scaled filters to the spectrum
- Each filter output is the sum of its filtered spectral components

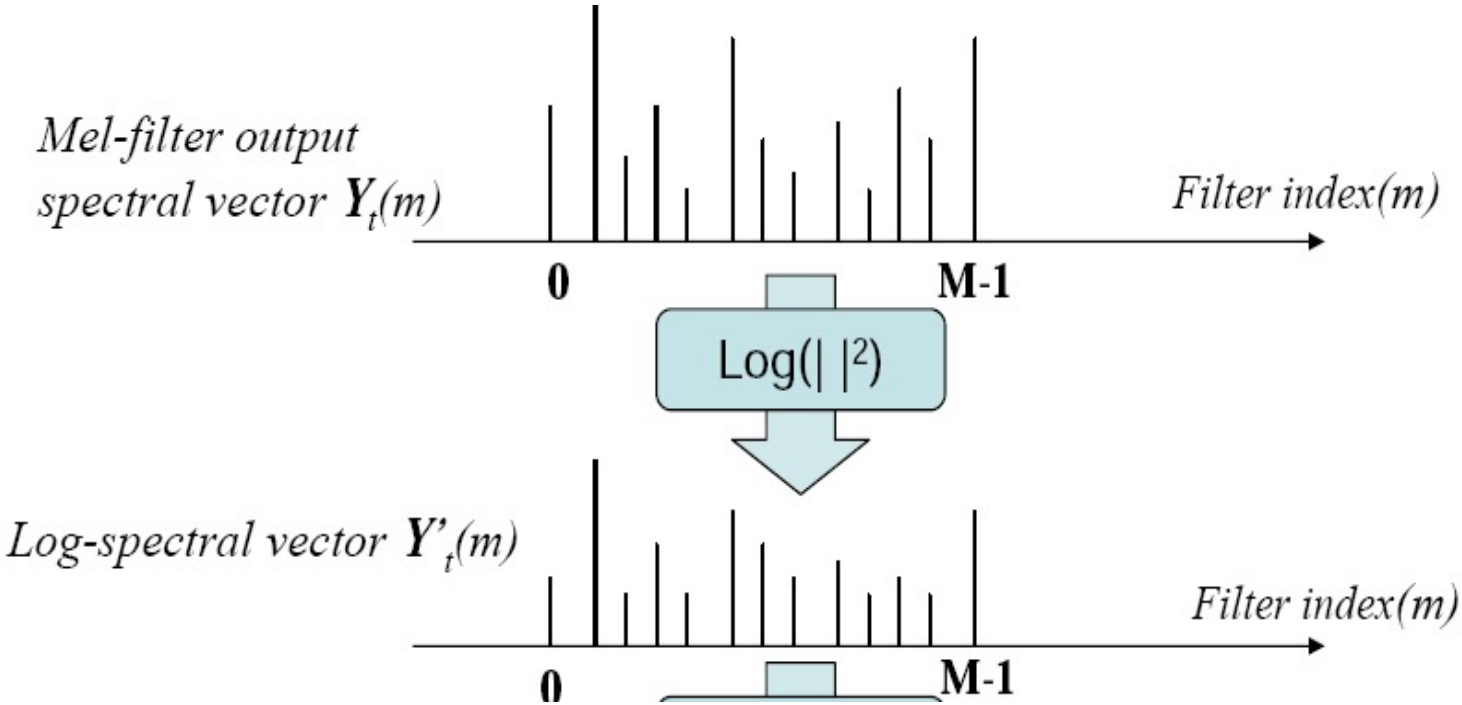


MFCC

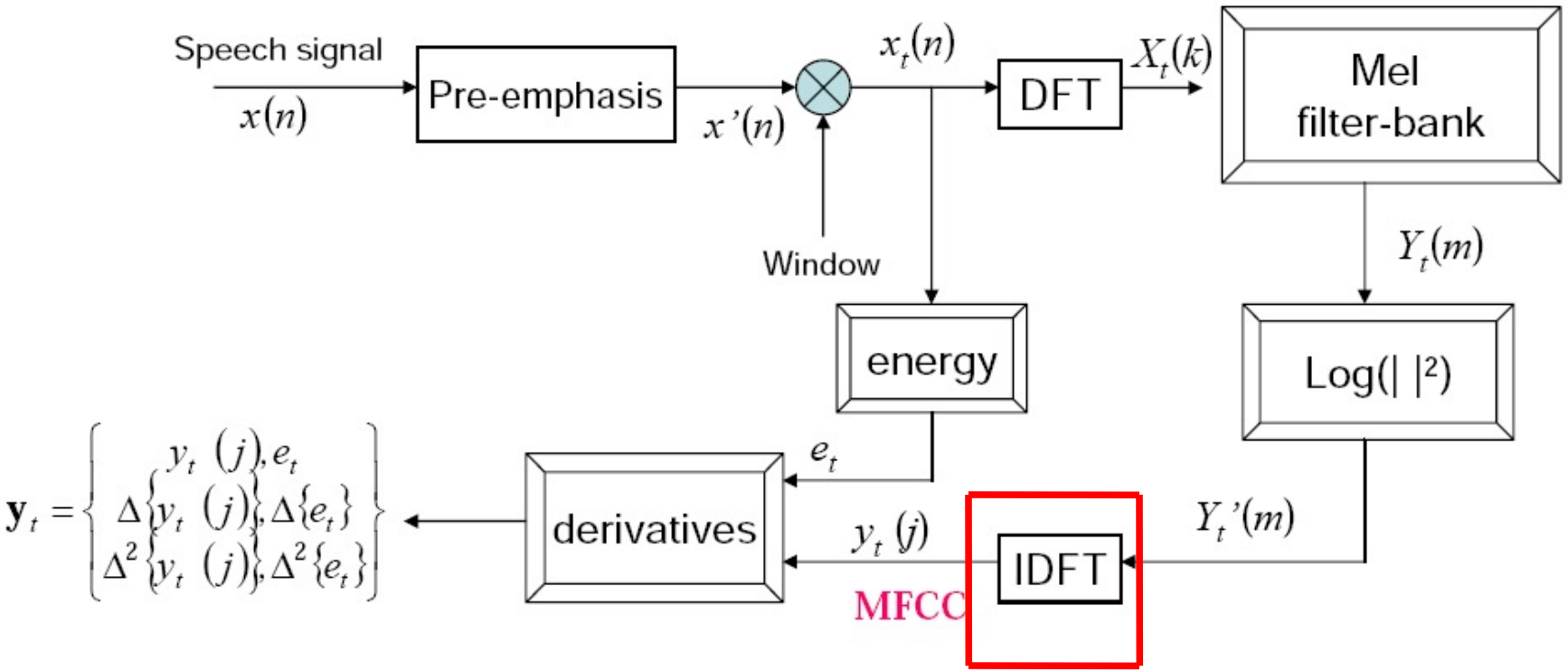

Speech signal
$x(n)$
Pre-emphasis
$x'(n)$
$x_t(n)$
Window
DFT
$X_t(k)$
Mel
filter-bank
$Y_t(m)$
Log(| |²)
$Y_t'(m)$
IDFT
$y_t(j)$
energy
e_t
derivatives
$\mathbf{y}_t = \begin{Bmatrix} y_t(j), e_t \\ \Delta\{y_t(j)\}, \Delta\{e_t\} \\ \Delta^2\{y_t(j)\}, \Delta^2\{e_t\} \end{Bmatrix}$
MFCC

Log energy computation

- Compute the logarithm of the square magnitude of the output of Mel-filter bank



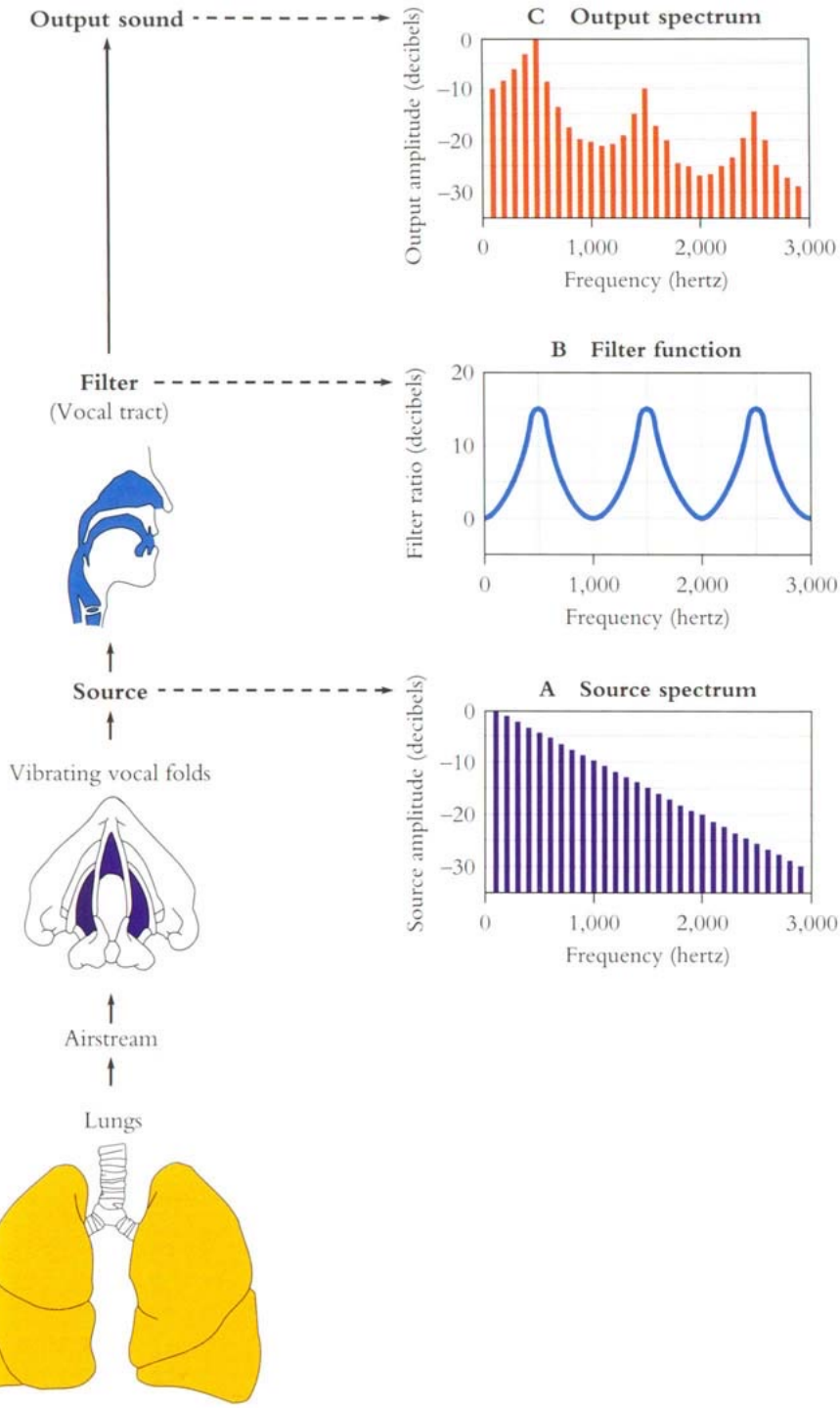
MFCC


Speech signal
$x(n)$
Pre-emphasis
$x'(n)$
$x_t(n)$
DFT
$X_t(k)$
Mel filter-bank
$Y_t(m)$
Log(| |²)
$Y_t'(m)$
IDFT
$y_t(j)$
MFCC
Window
energy
e_t
derivatives
$\mathbf{y}_t = \left\{ \begin{matrix} y_t(j), e_t \\ \Delta\{y_t(j)\}, \Delta\{e_t\} \\ \Delta^2\{y_t(j)\}, \Delta^2\{e_t\} \end{matrix} \right\}$

The Cepstrum

- One way to think about this
  - Separating the source and filter
  - Speech waveform is created by
    - A glottal source waveform
    - Passes through a vocal tract which because of its shape has a particular filtering characteristic
- Remember articulatory facts from lecture 2:
  - The vocal cord vibrations create harmonics
  - The mouth is an amplifier
  - Depending on shape of oral cavity, some harmonics are amplified more than others

George Miller figur

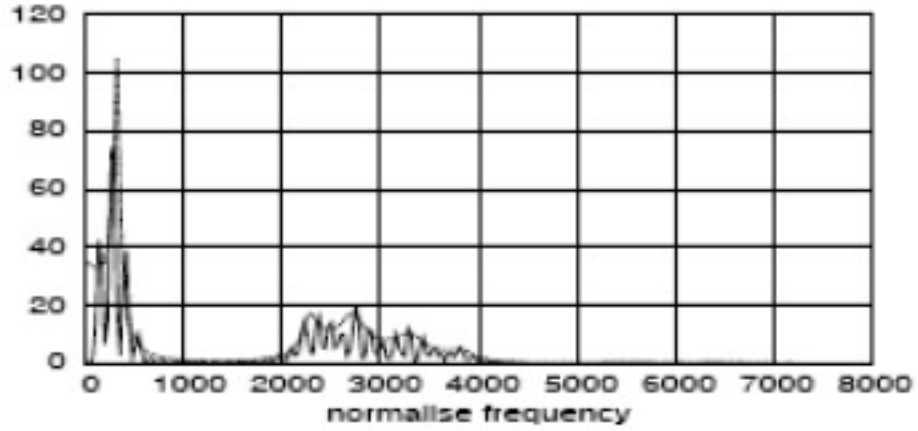


We care about the filter not the source

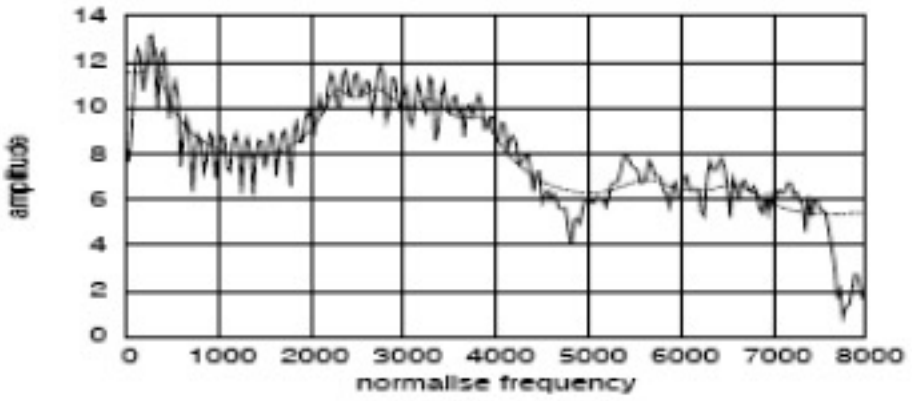
- Most characteristics of the source
  - F0
  - Details of glottal pulse
- Don’t matter for phone detection
- What we care about is the filter
  - The exact position of the articulators in the oral tract
- So we want a way to separate these
  - And use only the filter function

The Cepstrum

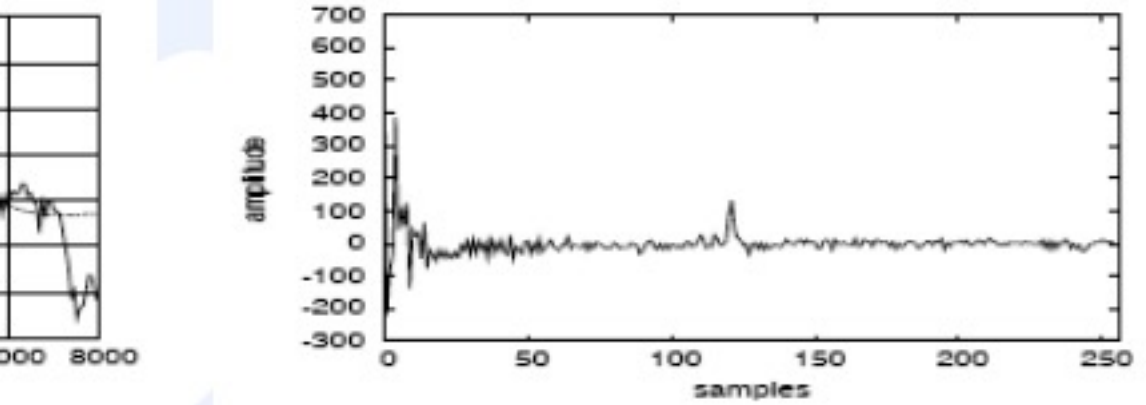
- The spectrum of the log of the spectrum



Spectrum



Log spectrum



Spectrum of log spectrum

Another advantage of the Cepstrum

- DCT produces highly uncorrelated features
- If we use only the diagonal covariance matrix for our Gaussian mixture models, we can only handle uncorrelated features.
- In general we'll just use the first 12 cepstral coefficients (we don't want the later ones which have e.g. the F0 spike)