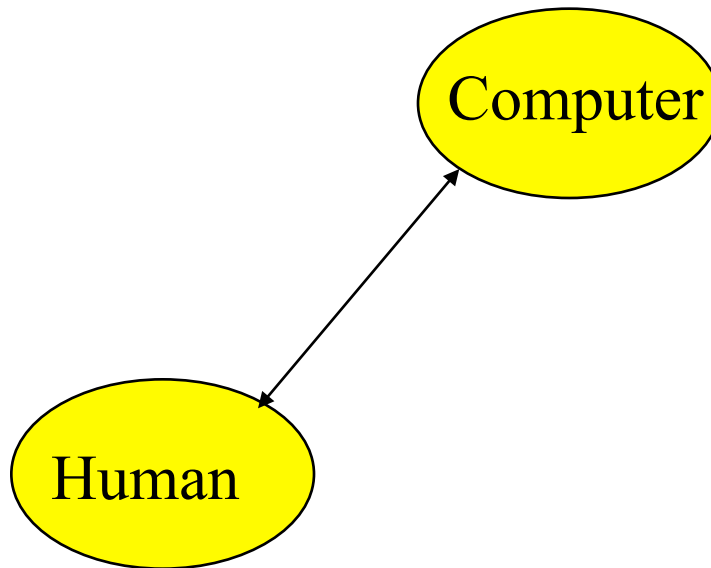
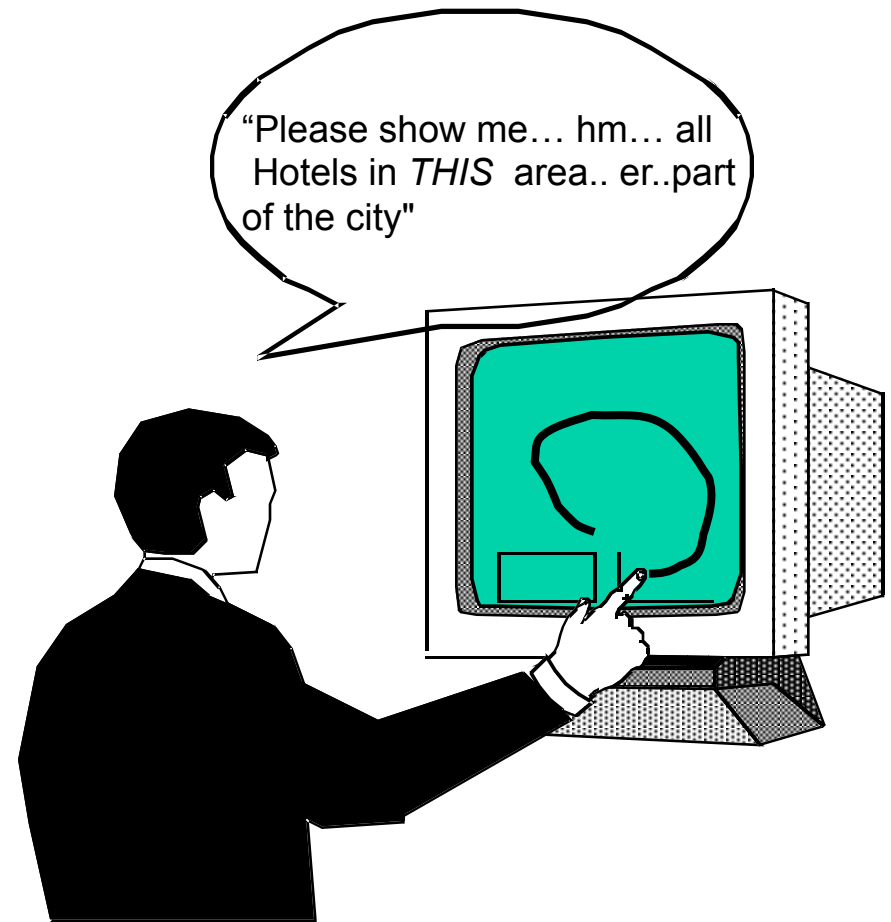


# Classical Human-Computer Interaction



# Better Human-Machine Interaction

- Speaking
- Pointing,
- Gesturing
- Hand-Writing
- Drawing
- Presence/Focus of Attention
- Combination
  - Sp+HndWrtg+Gestr.
  - Repair
- Multimodal NLP & Dialog
- Learning from Experience



# Interpreting Human Communication

*“Why did Joe get angry at Bob about the budget ?”*

Need Recognition and Understanding of Multimodal Cues

- Verbal:

- Speech

- Words
    - Speakers
    - Emotion
    - Genre

- Language
  - Summaries
  - Topic
  - Handwriting

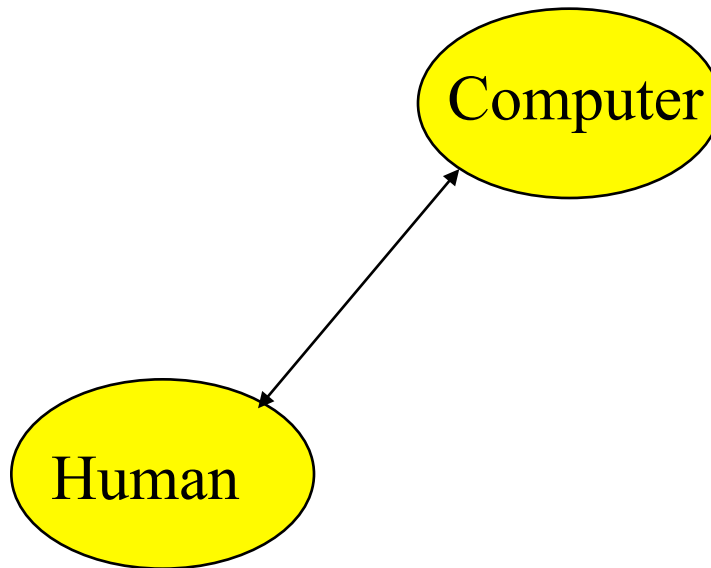
- Visual

- Identity
  - Gestures
  - Body-language
  - Track Face, Gaze, Pose
  - Facial Expressions
  - Focus of Attention



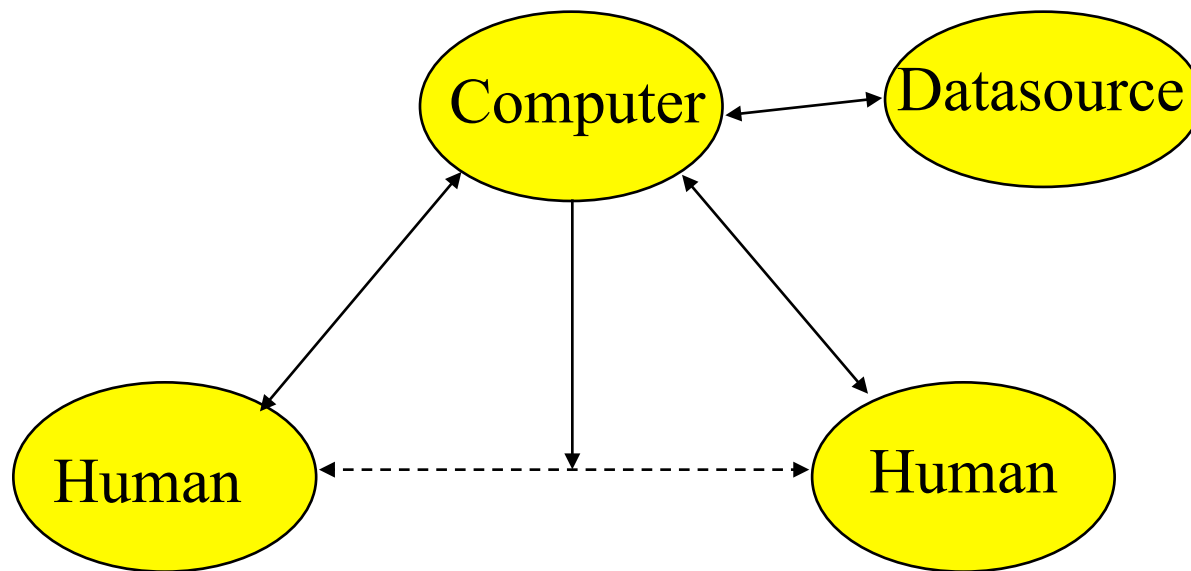
We need to understand the: **Who, What, Where, Why and How !**

# Classical Human-Computer Interaction





# Humans and Computers



# Interaction with Humanoid Robots



- Some Observations:
  - Multimodal Input
  - Robustness
  - Dialog
  - Learning from Interaction
  - Understanding the Context
  - Direct Interaction vs. Implied

# First Feasibility Demo

- 1991 – First Public Demonstration of Speech  
July 27, 1991 – UKA, CMU, ATR

## Machine speaks tongues

3 languages  
exchanged  
by computer

By Byron Spice  
Science Editor, Post-Gazette

You speak only English. The person at the other end of the telephone conversation speaks only German.



The system has a limitation — about 400 words — or stumbles when it hears sounds it doesn't recognize. Certainly no one expects the system to carry on conversations any time soon.

But Alex Waibel, a scientist at Carnegie Mellon University, says the limitations were beside the point. "You want to wait for the system," he asked, "or do you want one that handles a lot of things and can be easily modified?"

Named for the god of beginnings, the system was designed to meet the needs of Waibel and his colleagues. "conversational systems such as Janus must be able to register for meetings, order catalogs or make airline reservations."

## „Janus“ als Übersetzer

Karlsruhe (dpa) - Ein Computerprogramm, das in begrenztem Umfang gesprochene Sprache erkennen, „zerlegen“ und in das Englische und Japanische übersetzen kann, hat die Karlsruher Fakultät für Informatik vorgestellt. Es wird zur Zeit über einen Wortschatz von 400 Wörtern und in zwölf Dialogen als Partner angesprochen werden.

### Computer spricht drei Sprachen

Uni Karlsruhe ist Partner eines internationalen Forschungsprojekts

KARLSRUHE. Deutsche, amerikanische und japanische Wissenschaftler arbeiten gemeinsam an einem Übersetzungssystem, das schon in zehn Jahren die internationale Kommunikation erheblich erleichtern könnte. Dieses System „versteht“ drei Sprachen und antwortet auch in diesen Sprachen – vorerst in Deutsch, Englisch und Japanisch. Der Benutzer spricht

mit; ein französischer Partner ist bisher noch nicht gefunden worden. Das Projekt läuft seit fünf Jahren, Ausgangspunkt war ein Auftrag des japanischen Postministeriums für ein „übersetzendes Telefon“. Andere Forschungseinrichtungen in Japan arbeiten inzwischen an ähnlichen Projekten etwa zur automatischen Untertitelung fremdsprachiger Nachrichten.



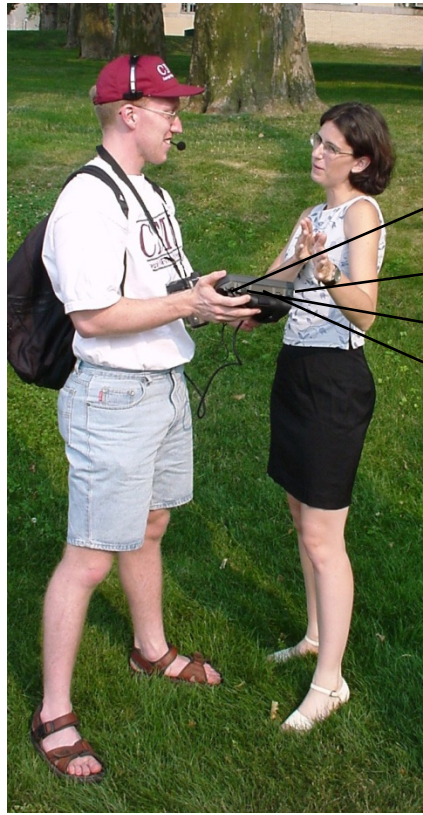
# First Speech Translation Conference '91-92

- 1992 – C-STAR Consortium for Speech Translation Advanced Research
- 1993 – Public C-STAR Demo, ATR-CMU-UKA-Siemens

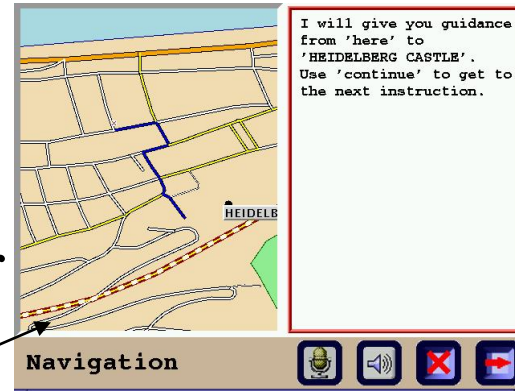


# LingWear:

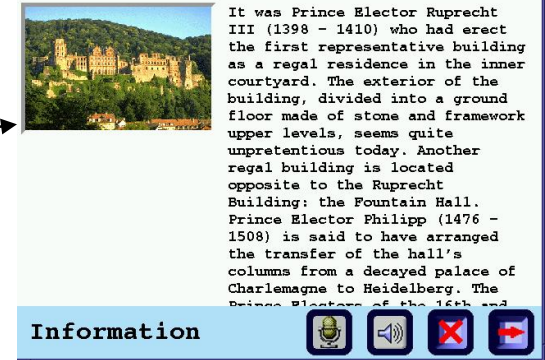
## Wearable Language Assistance for the Information Warrior



Navigation



Information Access



Document Translation



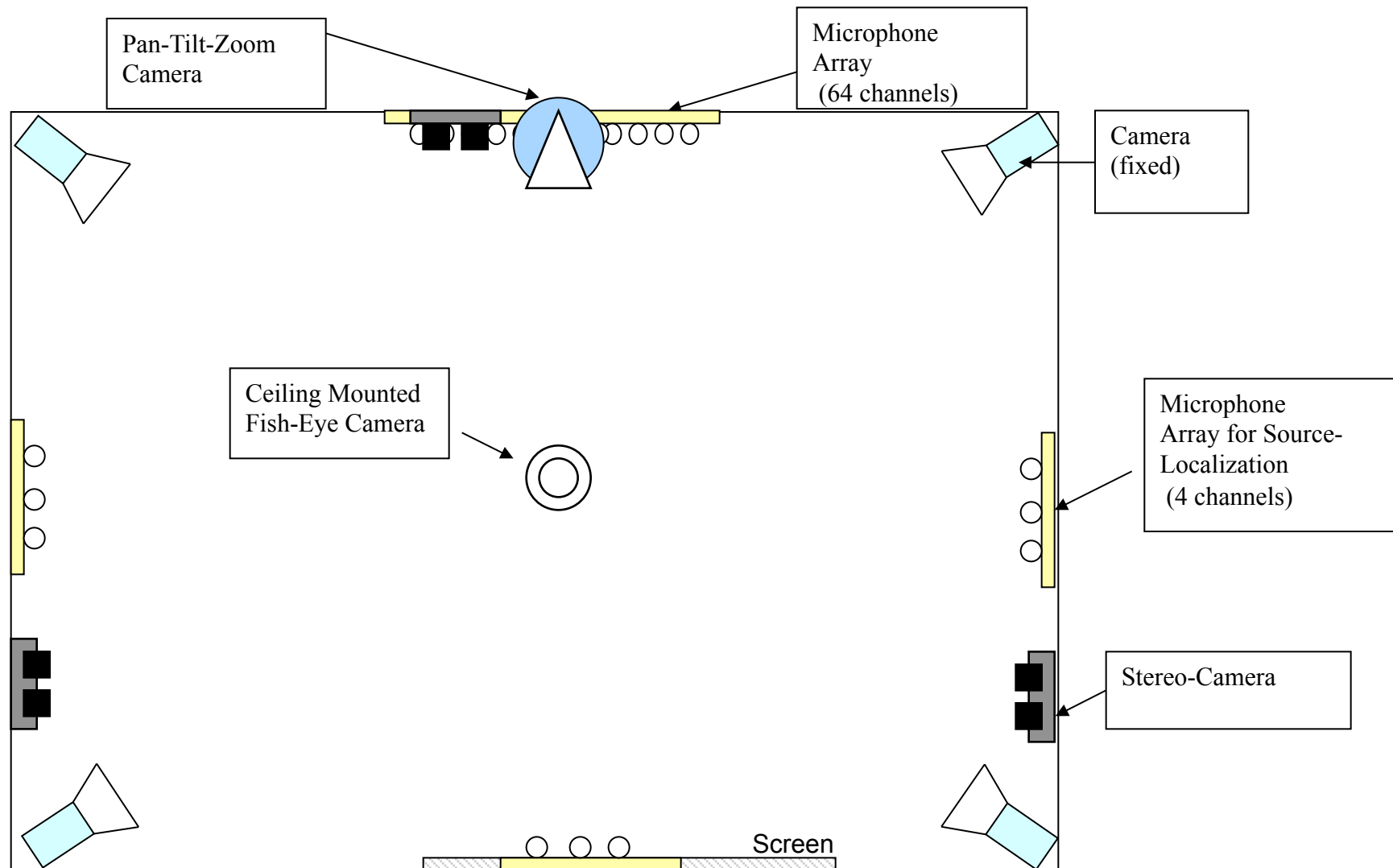
水栖、陆栖、两栖动物化石及鱼、蟹、龟、  
窝、蚌贝等；有各种草席、木席植物化石，  
达数百种。  
这是一个中外早已知名的科研现场。二  
代初，受到国内外地质考古学家的重视，  
行系统的考查研究，为世界地质科研提供  
典论著。直到七十年代末我国人类学家在  
区作了进一步的考查研究。



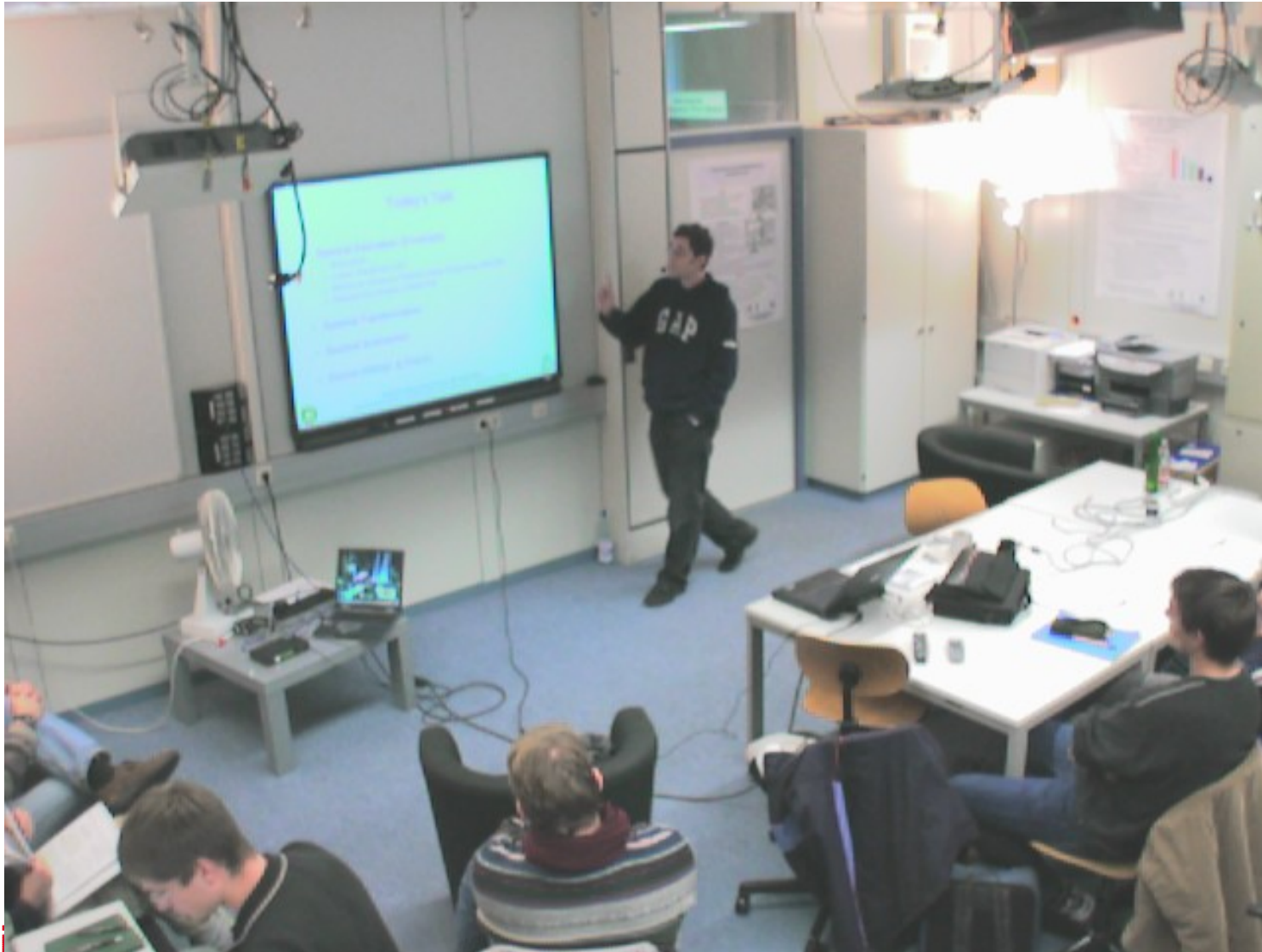
# Meetings



# Sensors in a Smart Room (Project CHIL)

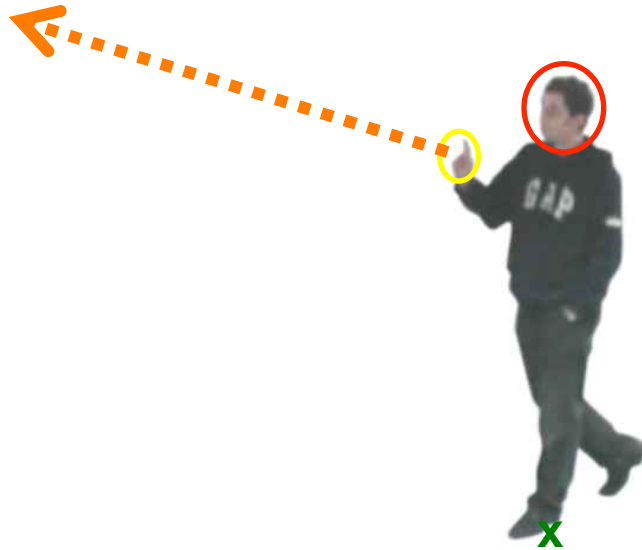


# Describing Human Activities

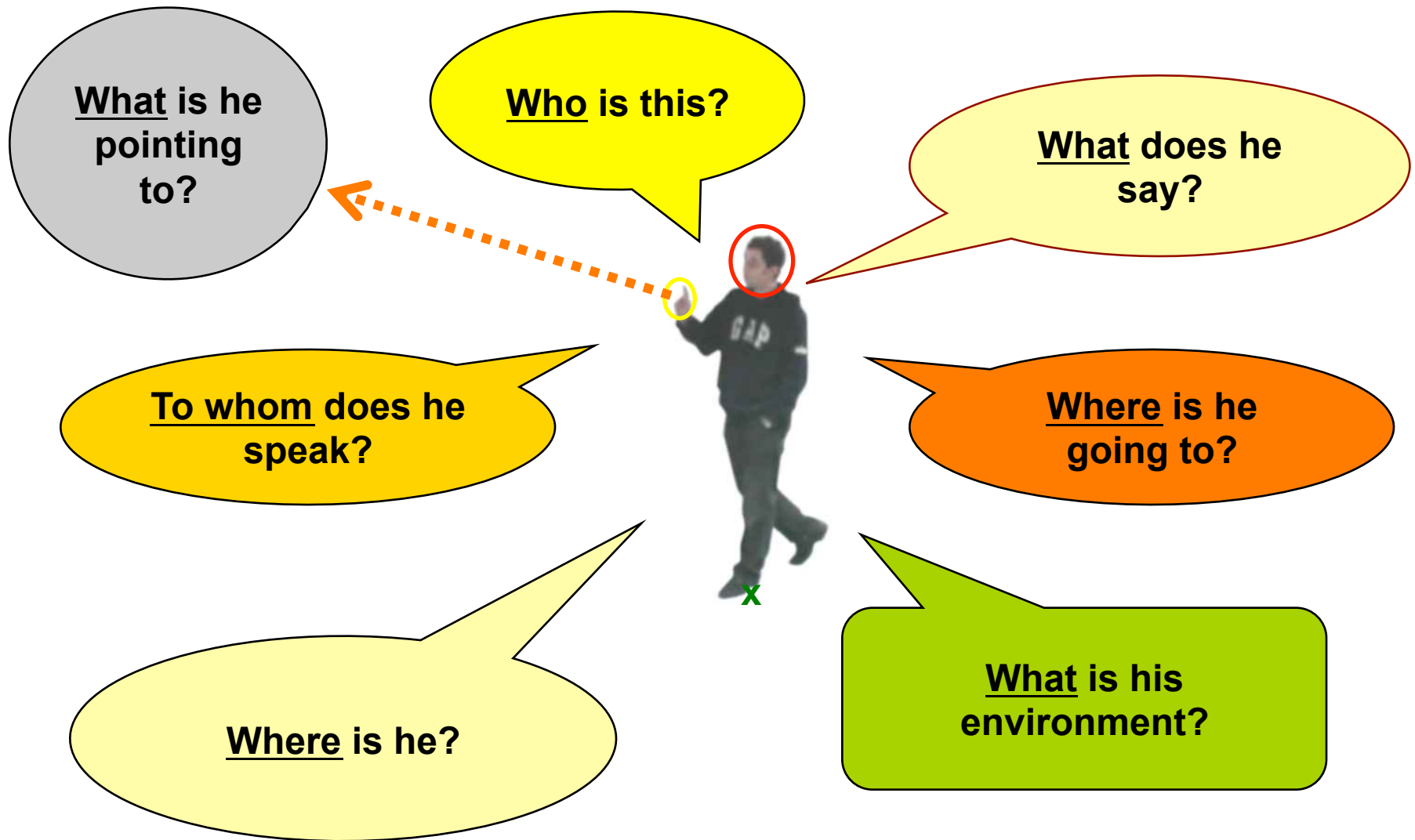




# Describing Human Activities



# Technologies/Functionalities



# Technologies & Fusion

- **Who & Where ?**

- Audio-Visual Person Tracking
- Tracking Hands and Faces
- AV Person Identification
- Head Pose / Focus of Attention
- Pointing Gestures
- Audio Activity Detection

- **What ? (Input)**

- Far-field Speech Recognition
- Far-field Audio-Visual Speech Recognition
- Acoustic Event Classification

- **What ? (Output)**

- Animated Social Agents
- Steerable targeted Sound
- Q&A Systems
- Summarization

- **Why & How ?**

- Classification of Activities
- Emotion Recognition
- Interaction & Context Modelling
- Vision-based posture recognition
- Topical Segmentation

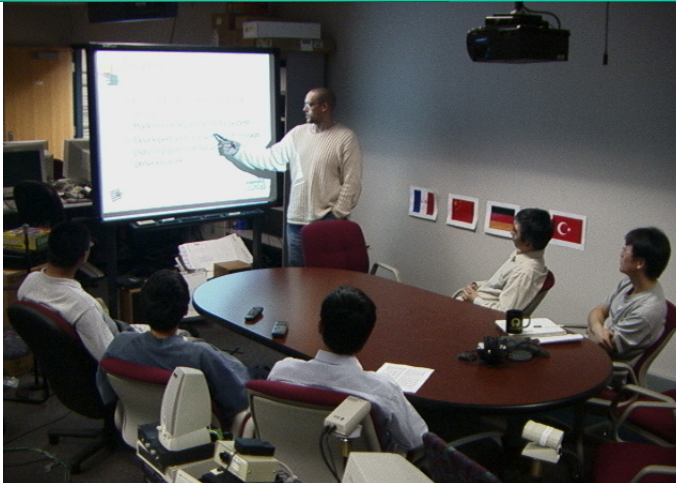
# Human-Human Communication in a Multilingual Distributed Context



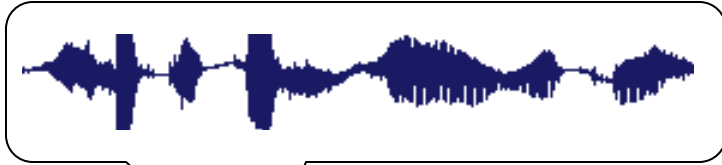
你们的评估准则是什么

## Meetings, Lectures

- Participants are Remote *and* Local
- Participants Speak Different Languages
  - Cross-Lingual Dialog *and* Translation of Monolingual Dialog
- Invisible Computer Provides Transparent Services
  - Translation
  - Summarization



# Speech



**Transcript: Onune baksana be adam!**

**Turkish**

**Bus  
Station  
Angry**

**Negotiation**

**Umut**

**Chemicals**

**Istanbul**

**Language ID**

**Acoustic Scene**

**Emotion ID**

**Discourse Analysis**

**Speaker ID**

**Topic ID**

**Entity Tracking**

# Speech

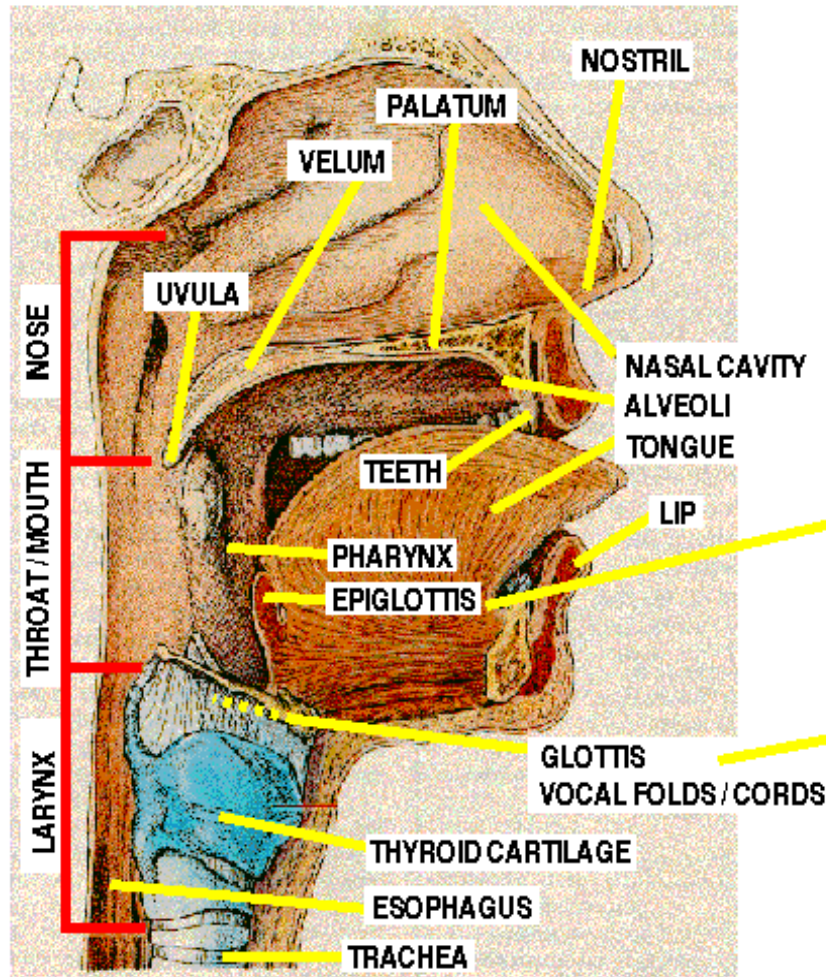
- Acoustic Phonetics, Speech Production and Perception
- Linguistics and Psychology
- Speech Recognition
  - Isolated and Continuous Word
  - Large Vocabulary Continuous Speech (Read Speech)
  - Conversational Speech
- Speaker Recognition
- Speaker Verification
- Language Identification
- Emotion Recognition
- Speech Synthesis
- Topic Identification
- Spoken Language Understanding
- Dialog Processing
- Machine Translation

# A Few Related Sciences

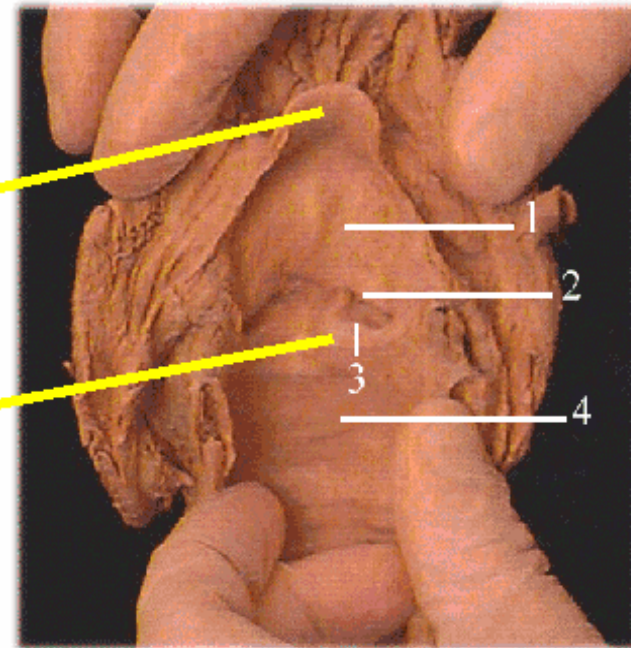
- Statistics
- Biology
- Linguistics
- Psychology
- Physiology / Anatomy
- Mathematics, Physics
- Electrical Engineering, Communication Theory
- Information Theory, Coding Theory
- Signal Processing
- Pattern Recognition
- Artificial Intelligence
- Language Processing



# Anatomy of Speech Production

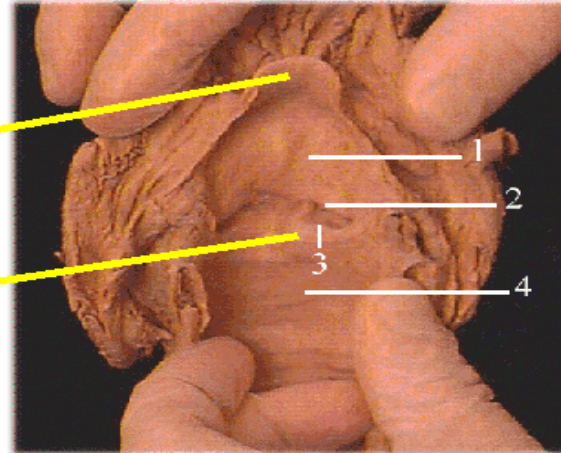
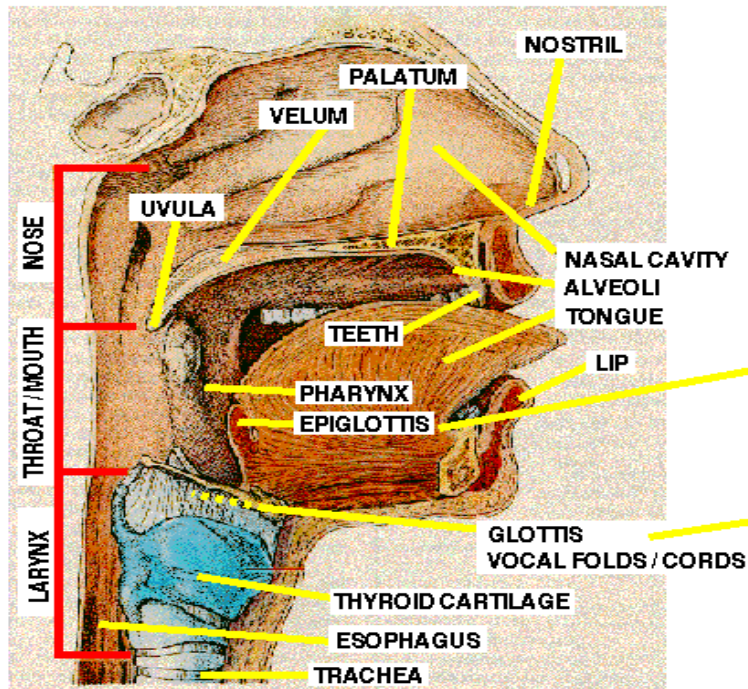


University  
Erlangen  
Department of  
Phoniatrics and  
Pedaudiology  
Waldstr.1  
D-91054 Erlangen

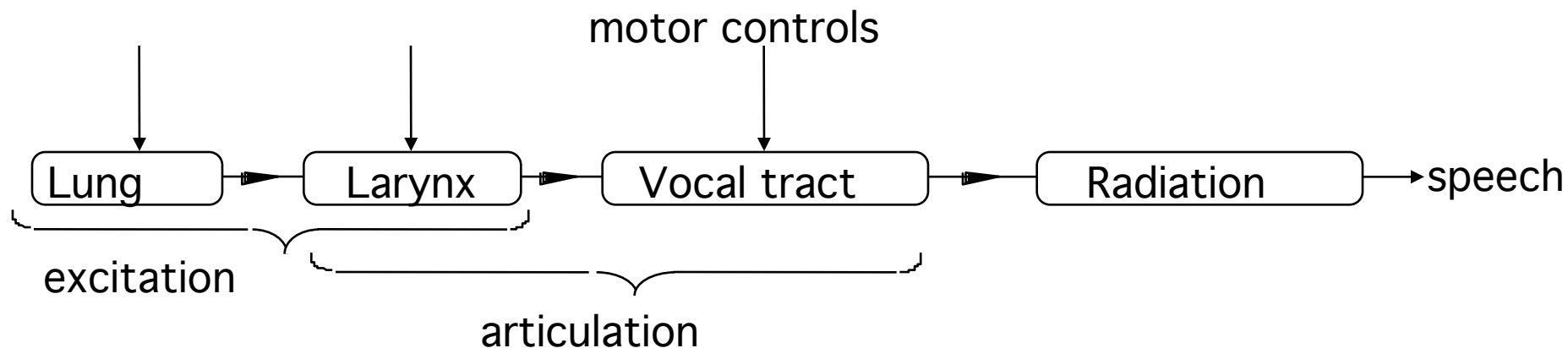




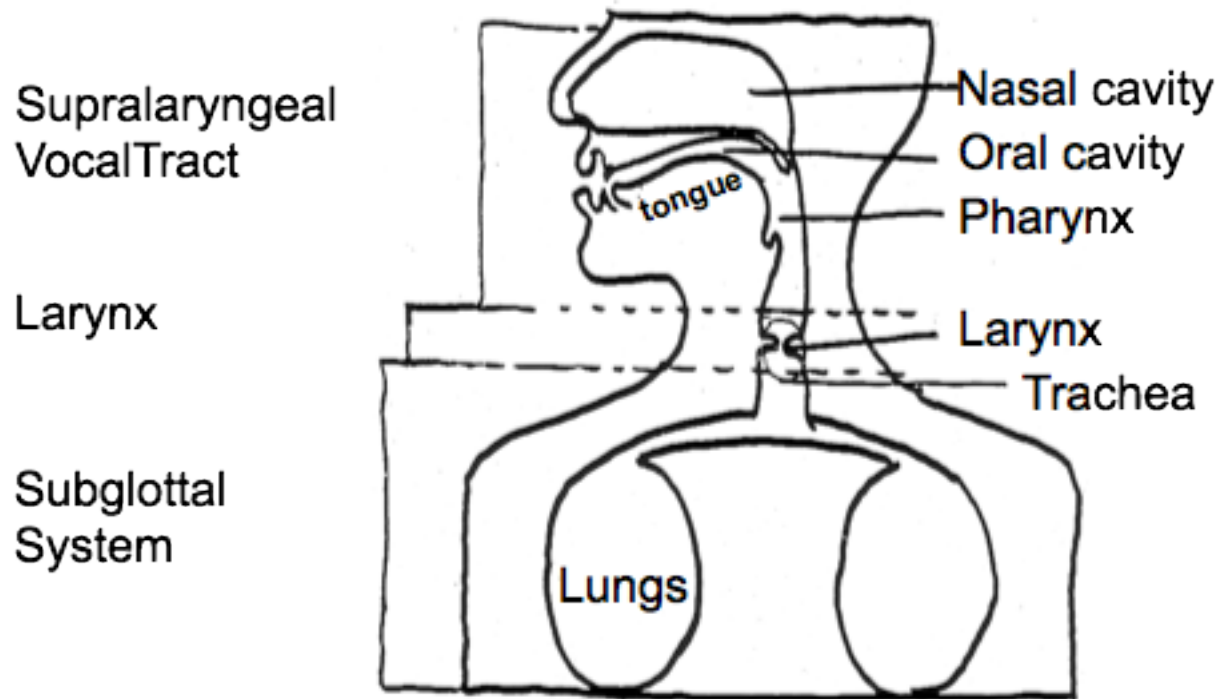
# Speech Production



The three physiologic components of human speech production

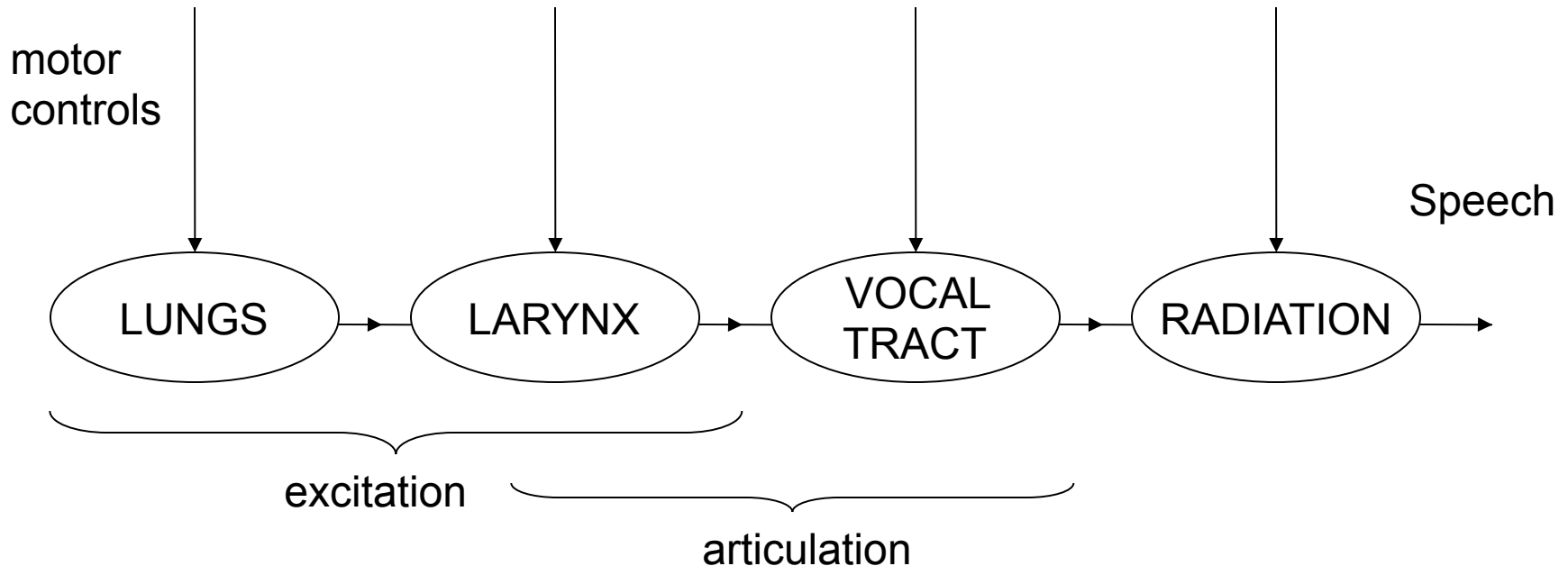


# Speech Production



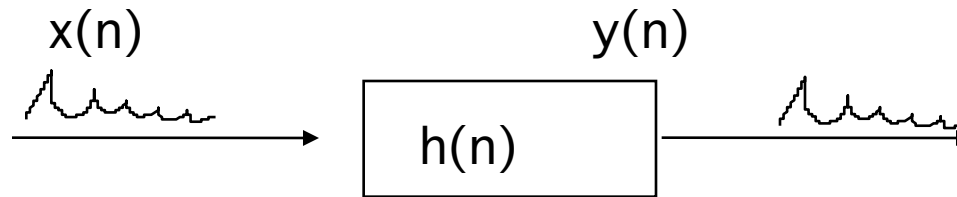
- The three physiologic components of human speech production -

# Speech Production (cont.)



- Functional Block Diagram of Speech Production -

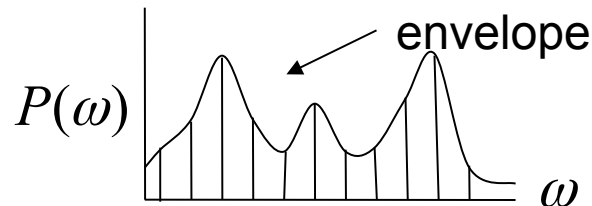
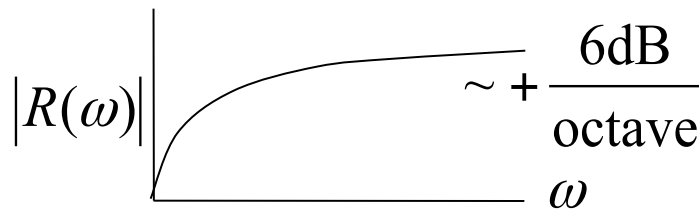
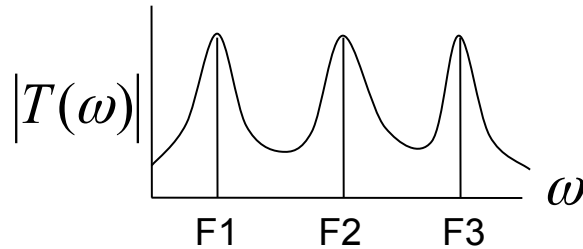
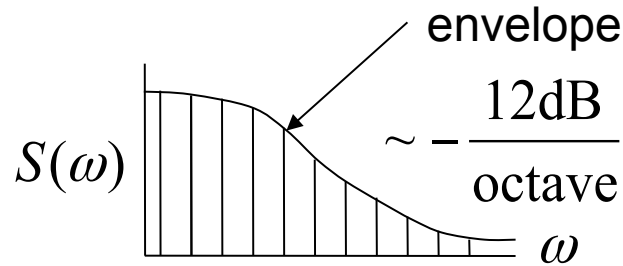
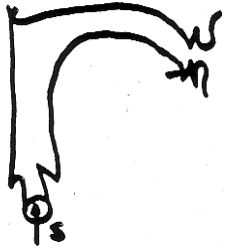
# Convolution



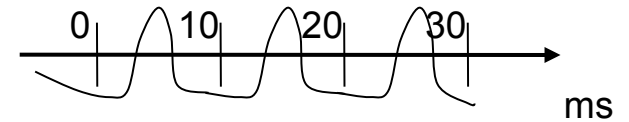
$$y(n) = x(n) * h(n) = \sum_{k=-\infty}^{\infty} x(k)h(n - k)$$

$$Y(\omega) = F(x(n) * h(n)) = X(\omega) \cdot H(\omega)$$

# Transfer Functions of the Different Components of Speech Production



Periodic excitation (Vowel)



# Different Vocal Tract Shapes

**BEAT**



**BIT**



**BAIT**



**BET**



**BAT**



**BART**



**BALL**



**BOY**



**BUTCH**



**BOOT**



**BUT**



**BIRD**

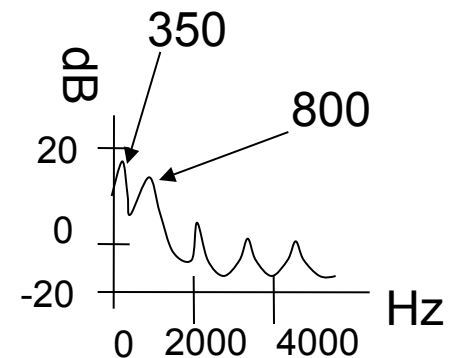
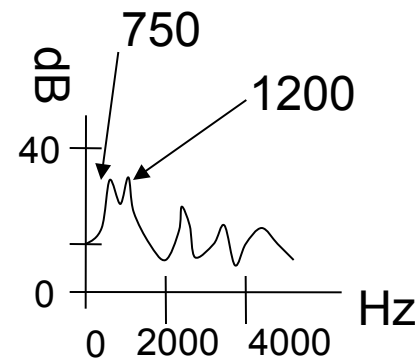
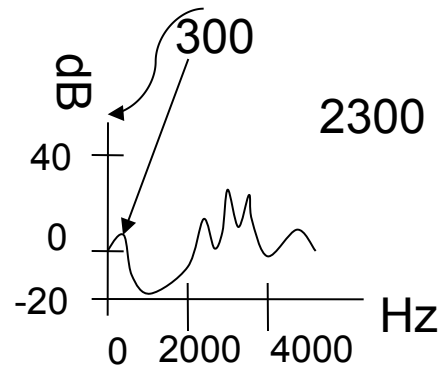


# Vocal Tract Transfer Functions for Different Vowels

Vocal Tract Shapes

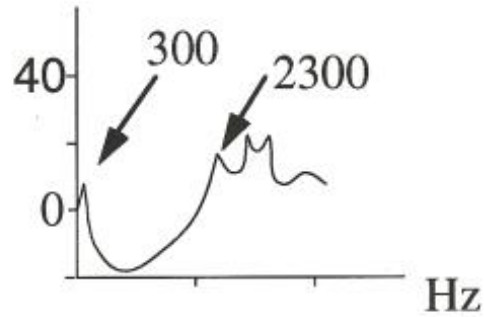


Resulting Transfer Functions (Spectra)

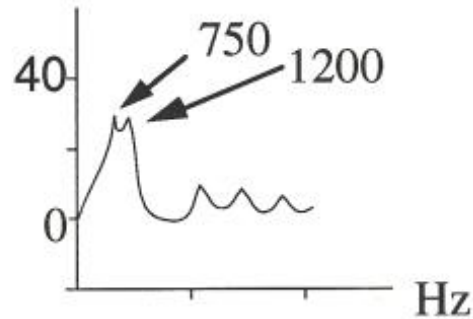


# Vokaltrakt & Transferfunktionen

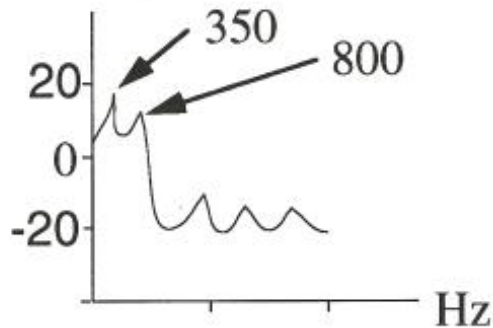
/i/



/a/



/u/

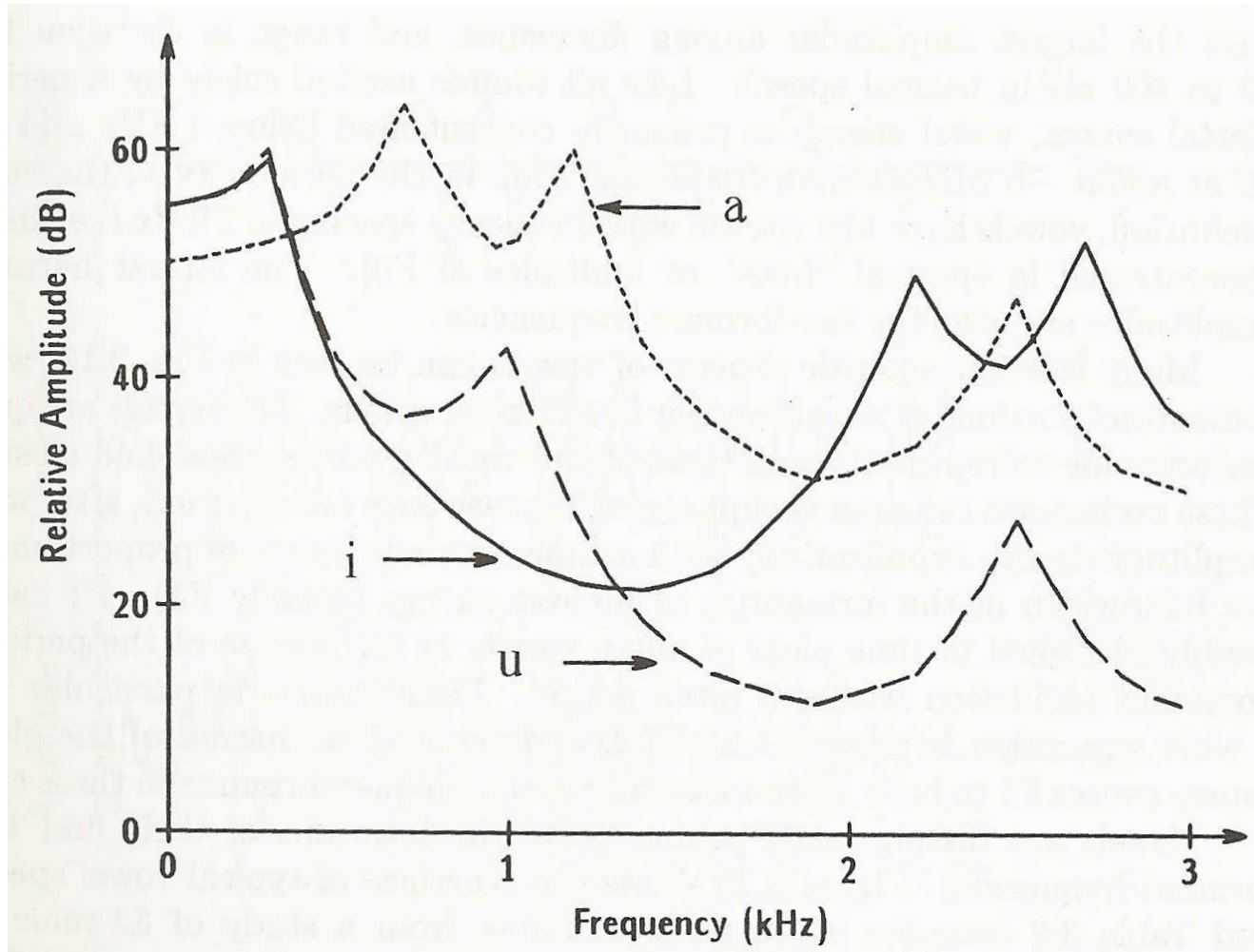


Vokaltraktformen

Resultierende  
Transferfunktionen



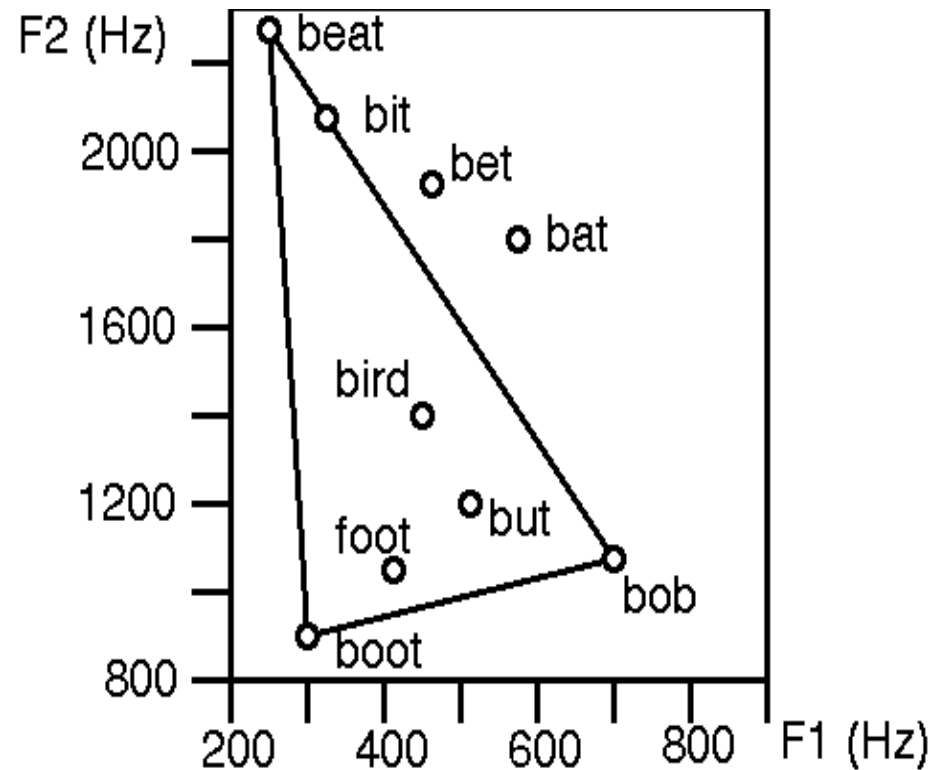
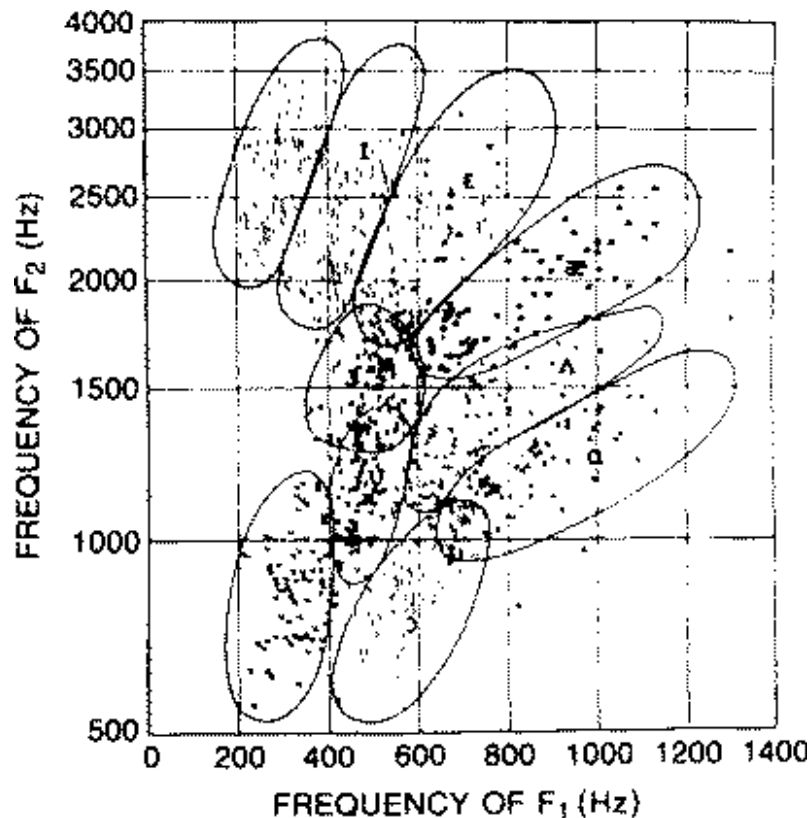
# Die Vokale /a/, /i/ und /u/



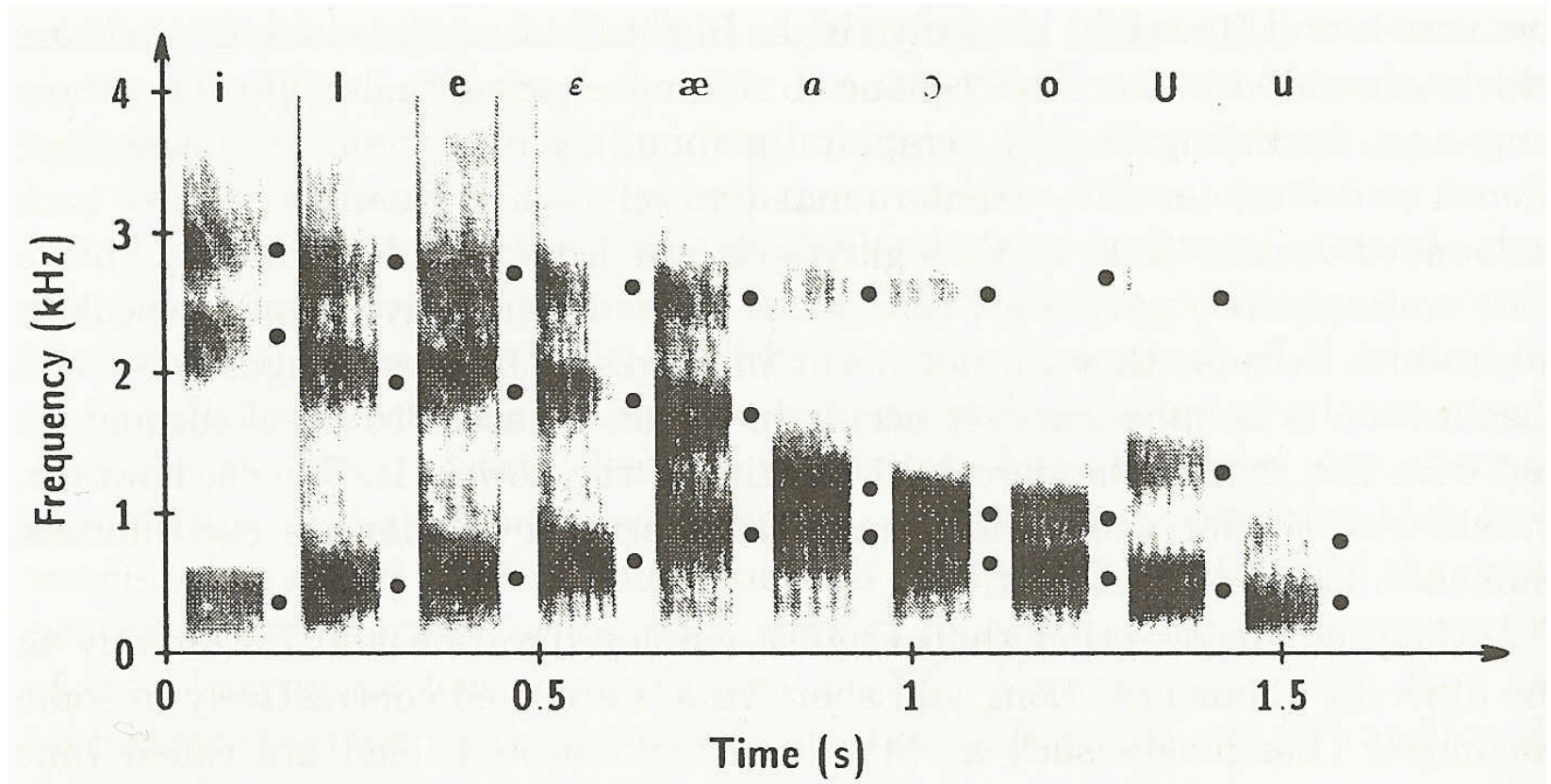
# Formants

The resonance frequencies of the vocal tract transfer function are called formants. In practice, only the first few formants are of interest.

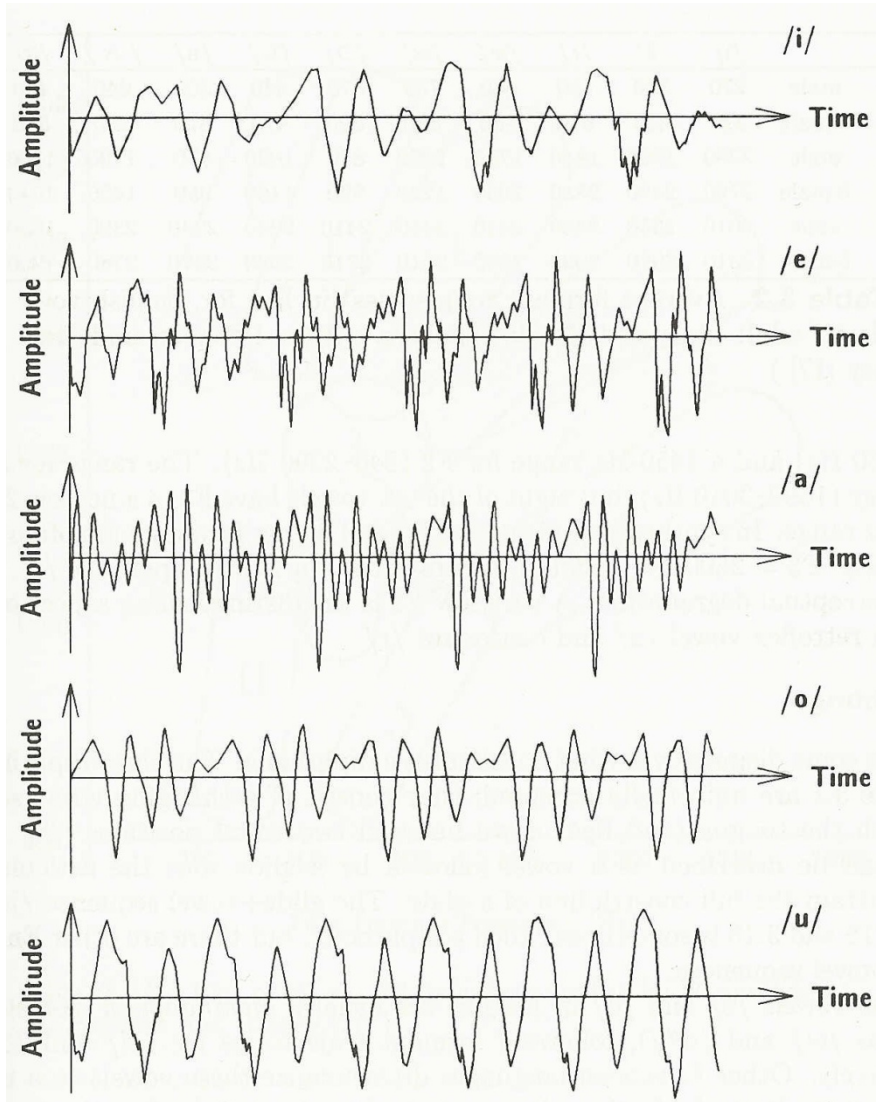
## The Vowel-Triangle



# Spektrogramme



# Vokale im Zeitbereich

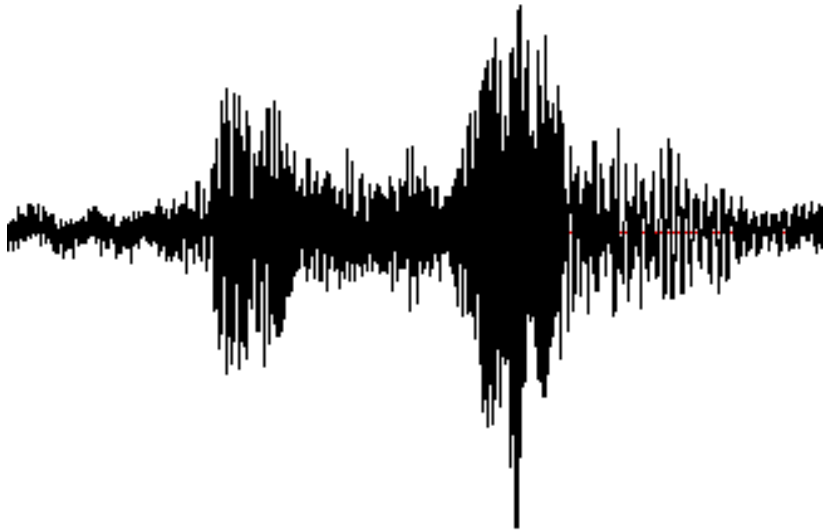


# Consonants

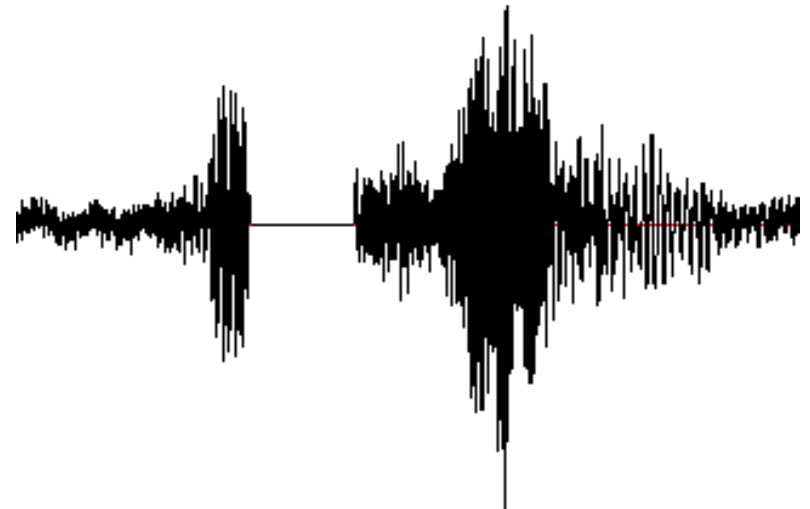
Consonants are sounds which are articulated by temporarily constricting the airflow or stopping the airflow completely.

Listen to these examples:

original recording 



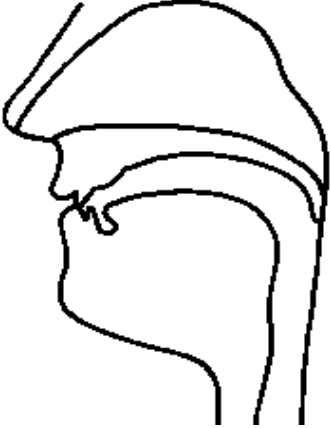


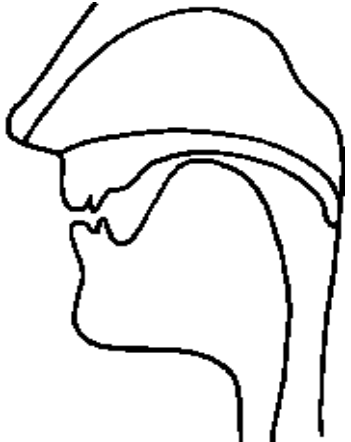
one part blanked out 



The blanked-out part sounds like a (plosive) consonant.

# Vocal Tract Shapes of Consonants

## Fricatives

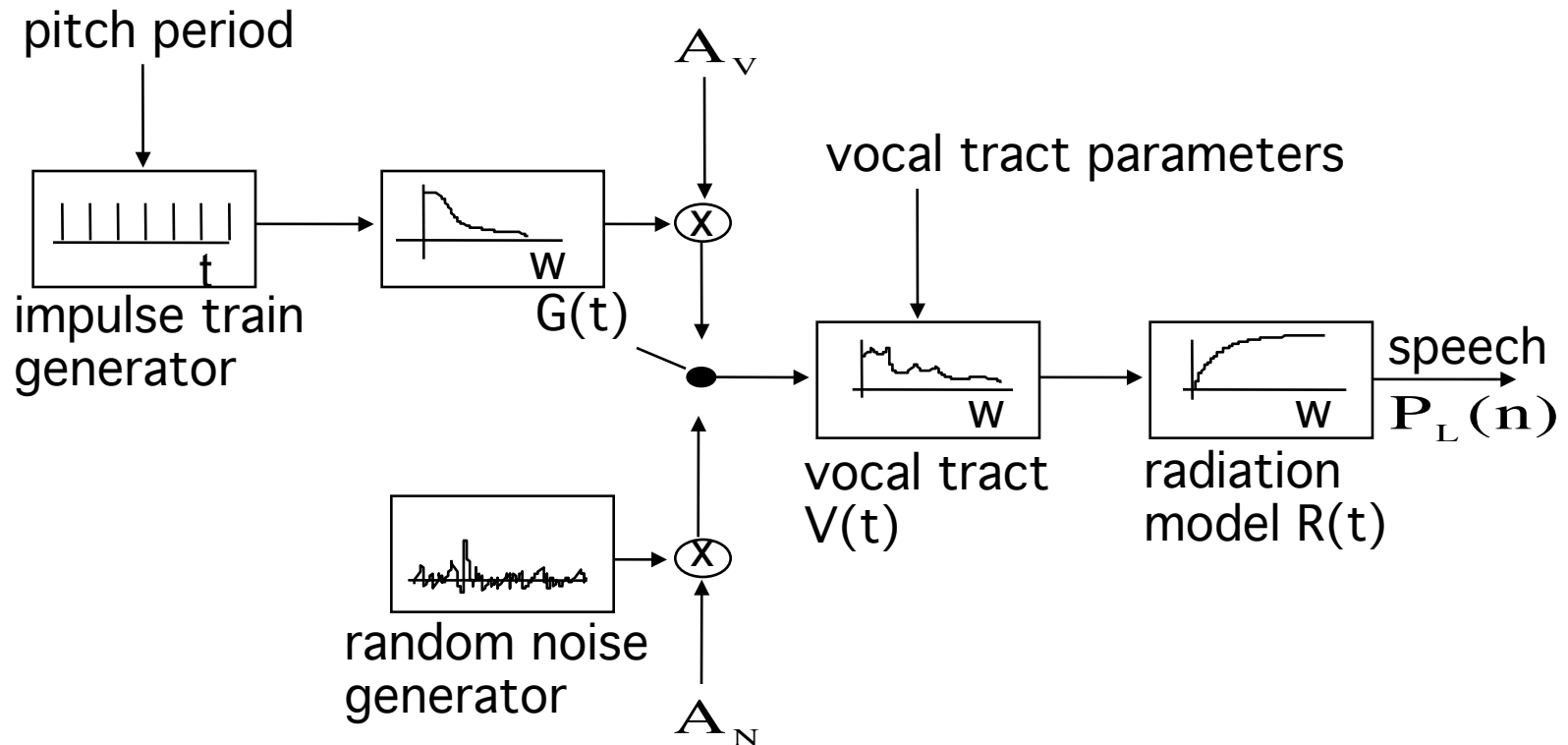
<p>Lip-Teeth Friction</p>  <p><b>FAN, VAN</b></p>	<p>Tongue-Teeth Tongue-Alveoli</p>  <p><b>SUE, ZOO</b></p>	<p>Palatal Friction Alveolar Friction</p>  <p><b>VISION, VICIOUS</b></p>	<p>Palatal Friction</p>  <p><b>YOU</b></p>
--	---	--	---

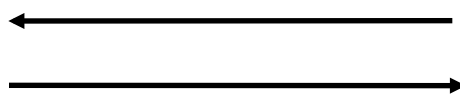
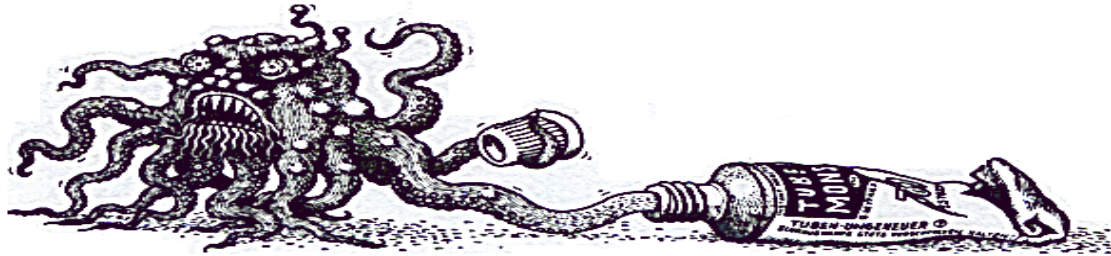
Additionally there is a glottal fricative /h/ as in **HOUSE**.

Other languages often also have aspirated velar and palatal fricatives.



# Vocal Tract Model of Speech





speech synthesis  
speech recognition



# Speech Recognition

**Alex Waibel**

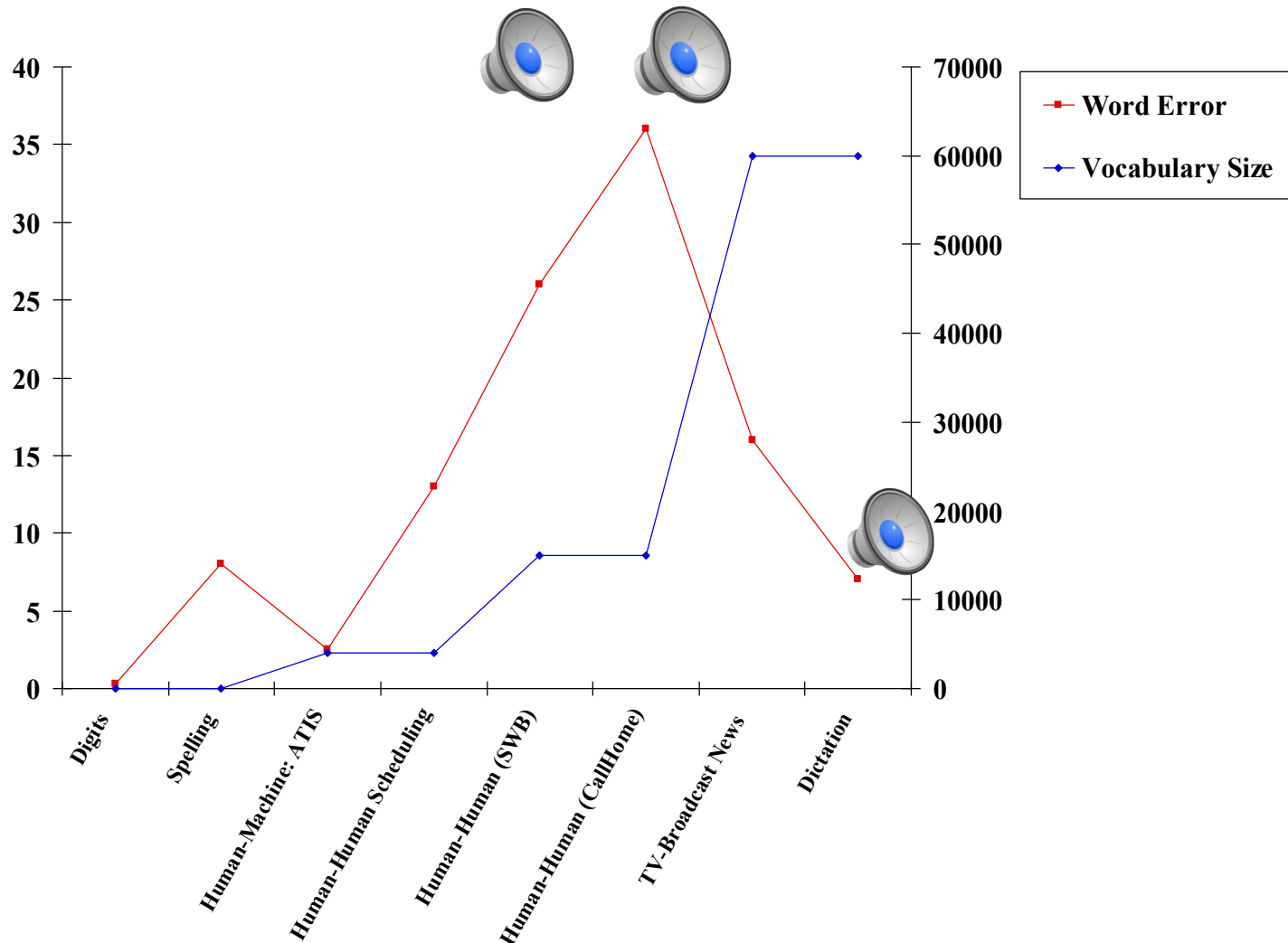
# How Good is a Recognizer?

- Word Error Rate:
  - Insertion, Deletion and Substitutions
  - $WER = 1/N * (\#Ins + \#Del + \#Sub)$
  - Determine I,D,S, by performing Alignment Search between output Hypothesis and Reference
- Perplexity
- Other Factors...

# Dimensions of Difficulty

- Noise – Environmental, Channel, Reverberation
- Speaker – Male, Female, Children, Elderly
- Acoustic Similarity – Letters, Digits,...
- Vocabulary Size – 10 → 100,000 words
- Speaking Style – Isolated, Continuous Read Speech, Spontaneous, Conversational Speech

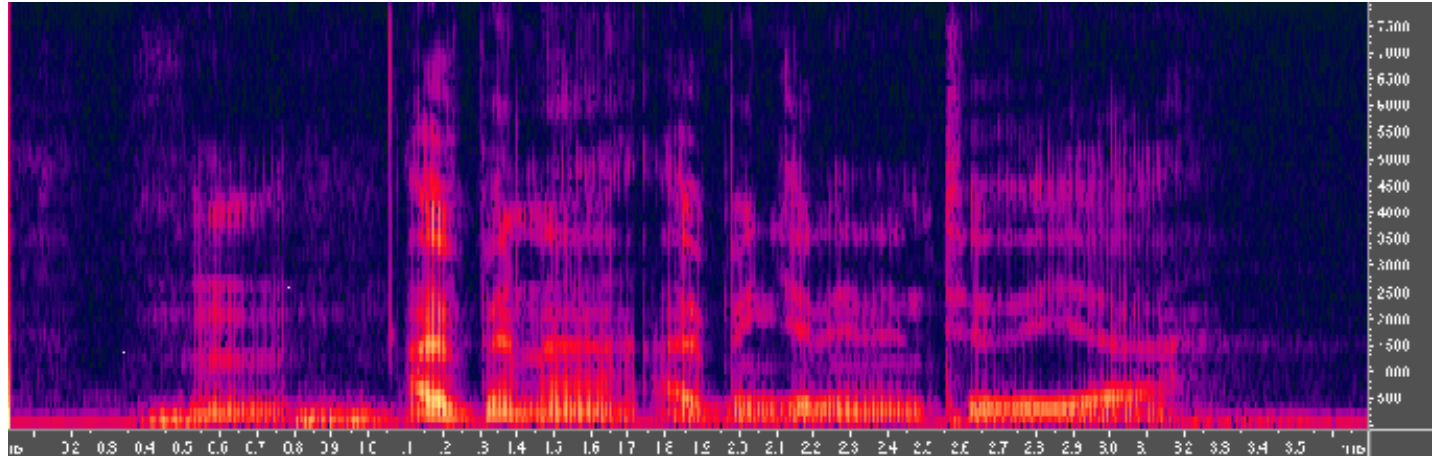
# Speech: State-of-the Art



# Sloppy Speech

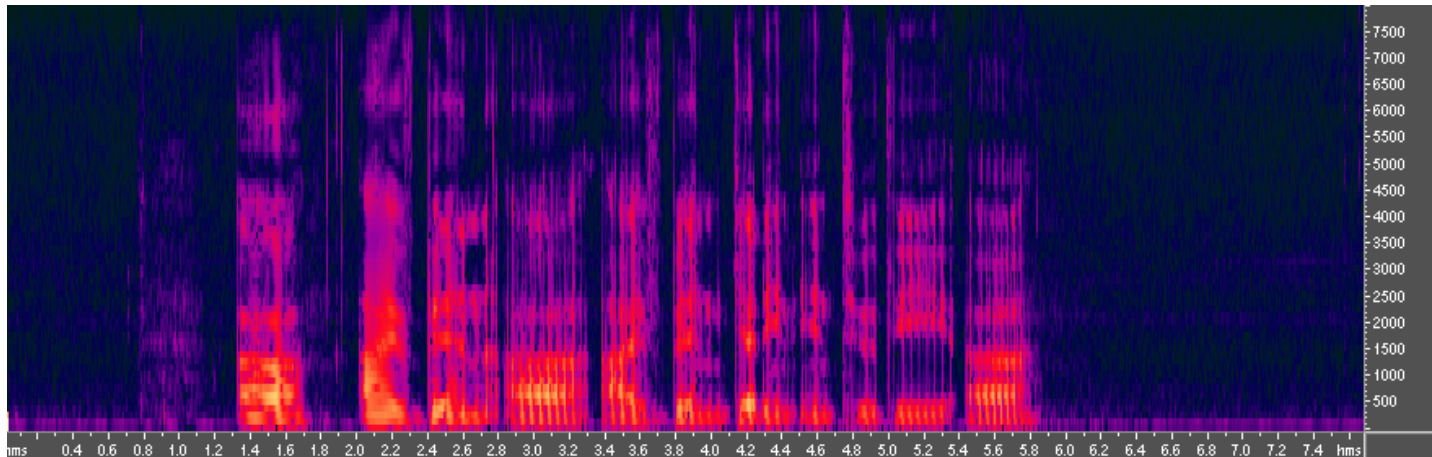
Actual Input: *"I have been I have been getting into"*

Conver-  
Sational  
Speech



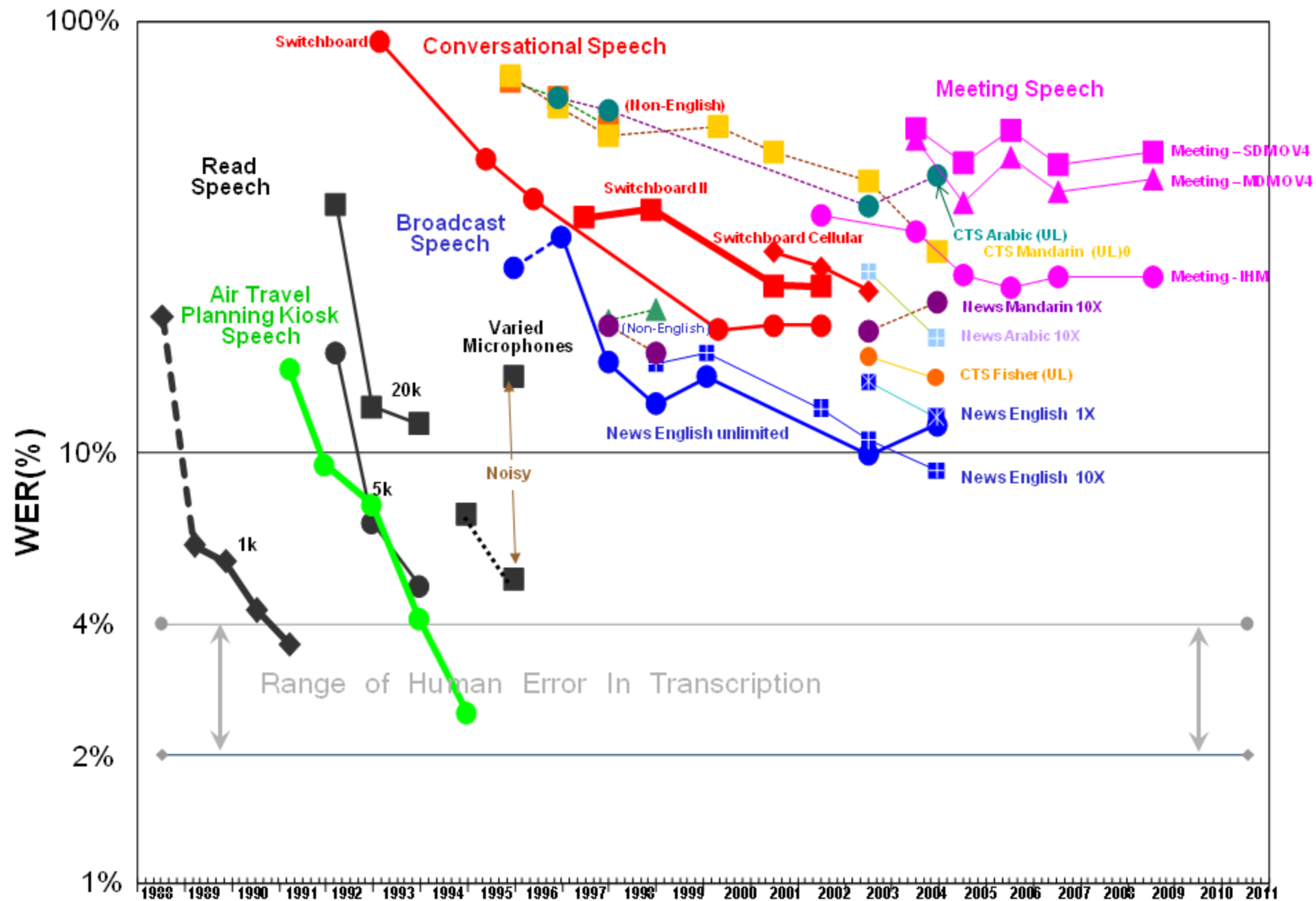
Recognition: *"and I am I being too yeah"*

Read  
Speech

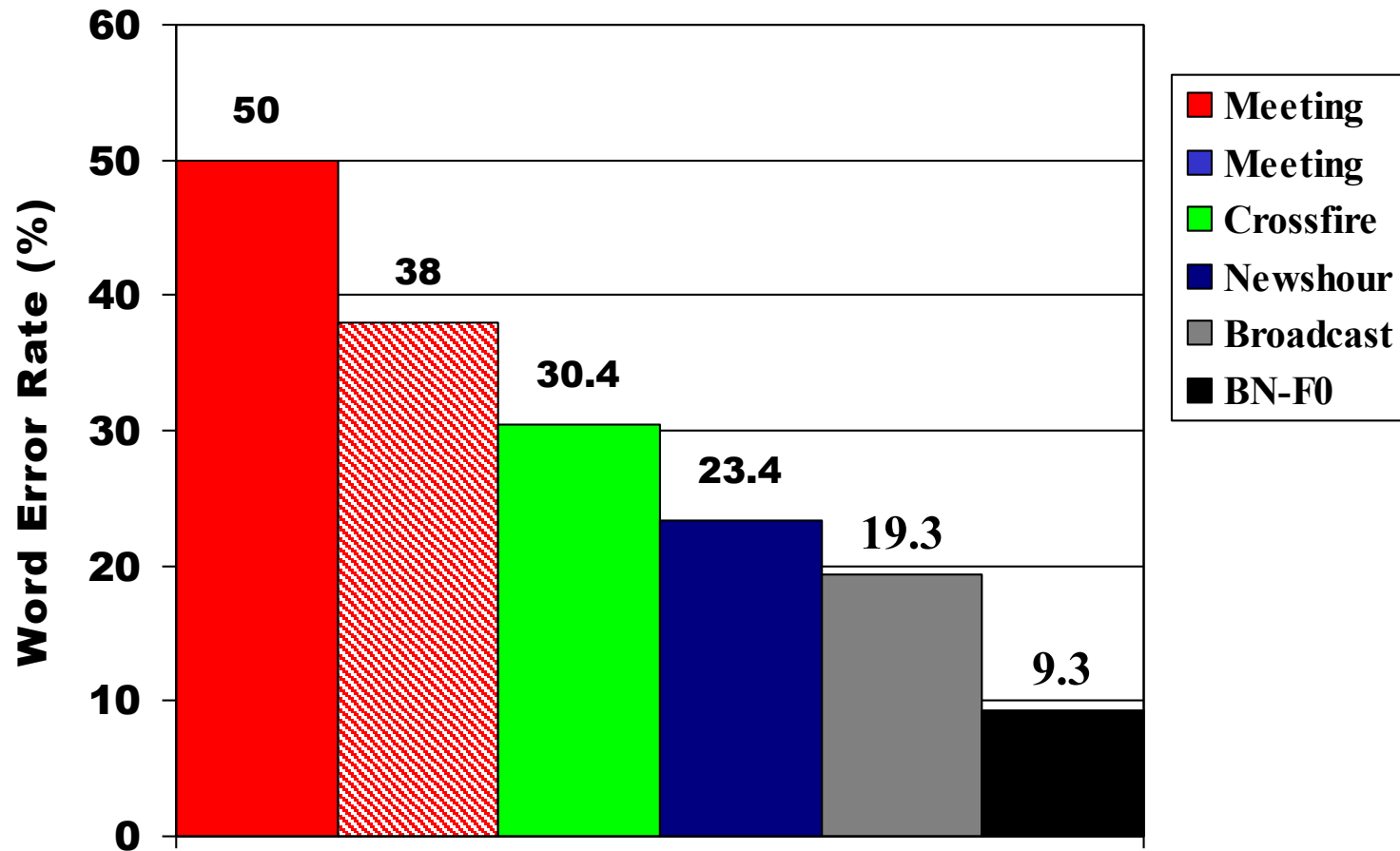


Recognition: *"I have been ties than getting into the"*

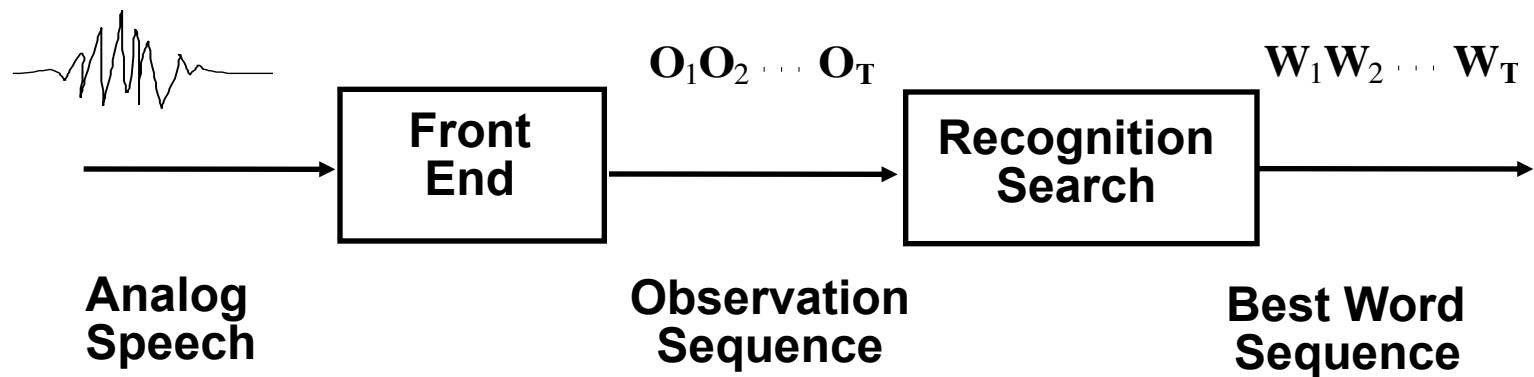
# NIST STT Benchmark Test History – May. '09



# Recognition of Speech in Meetings

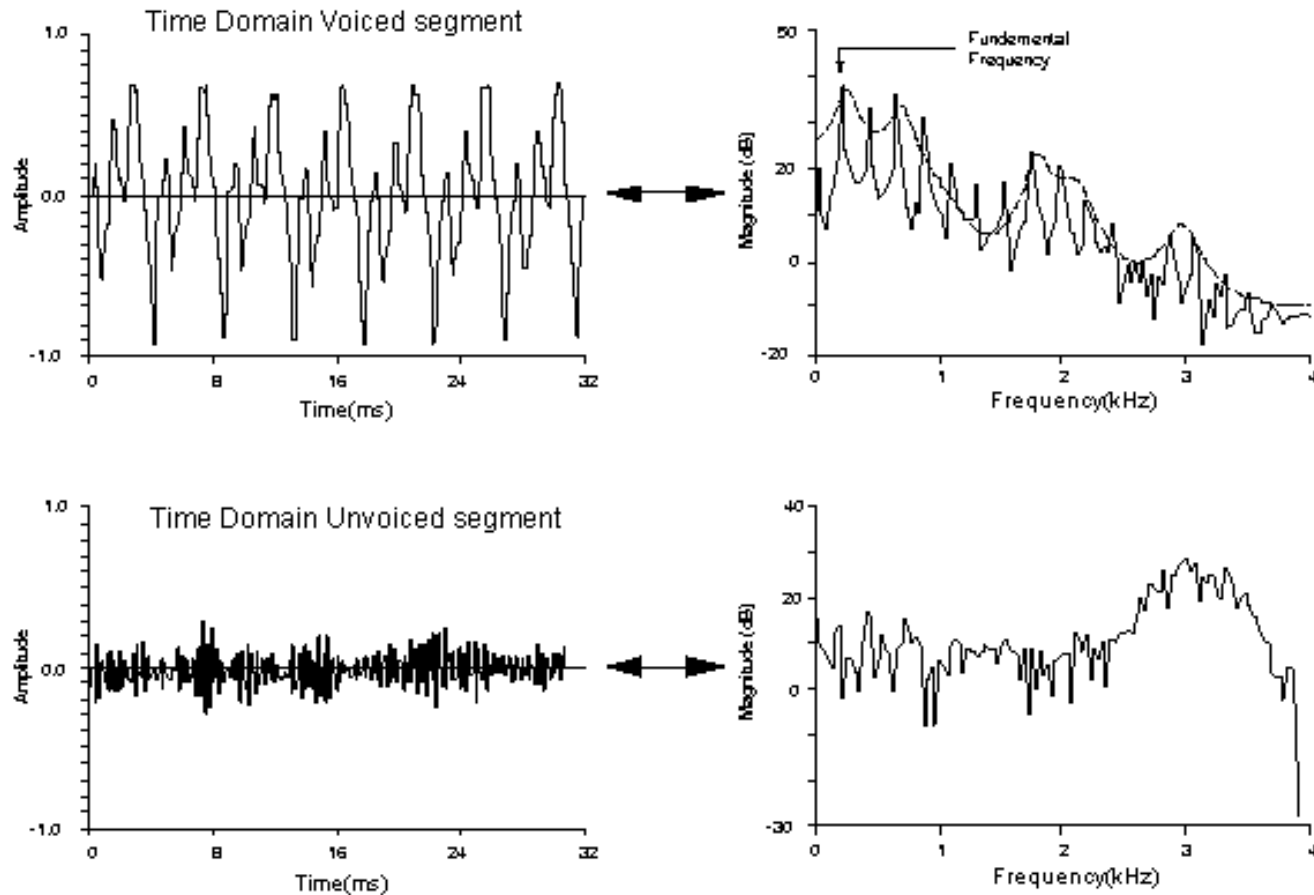


# Speech Recognition (System Overview)

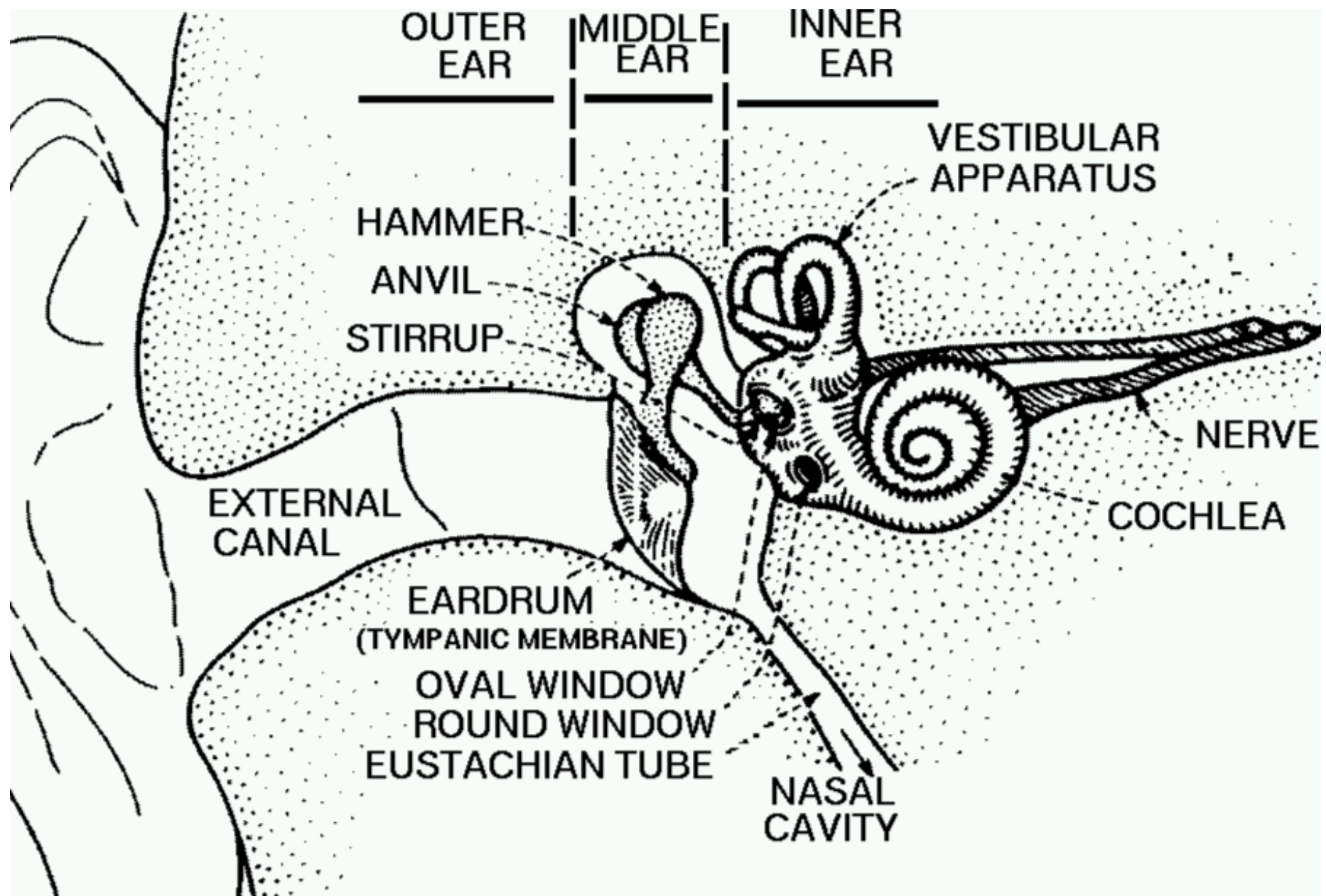




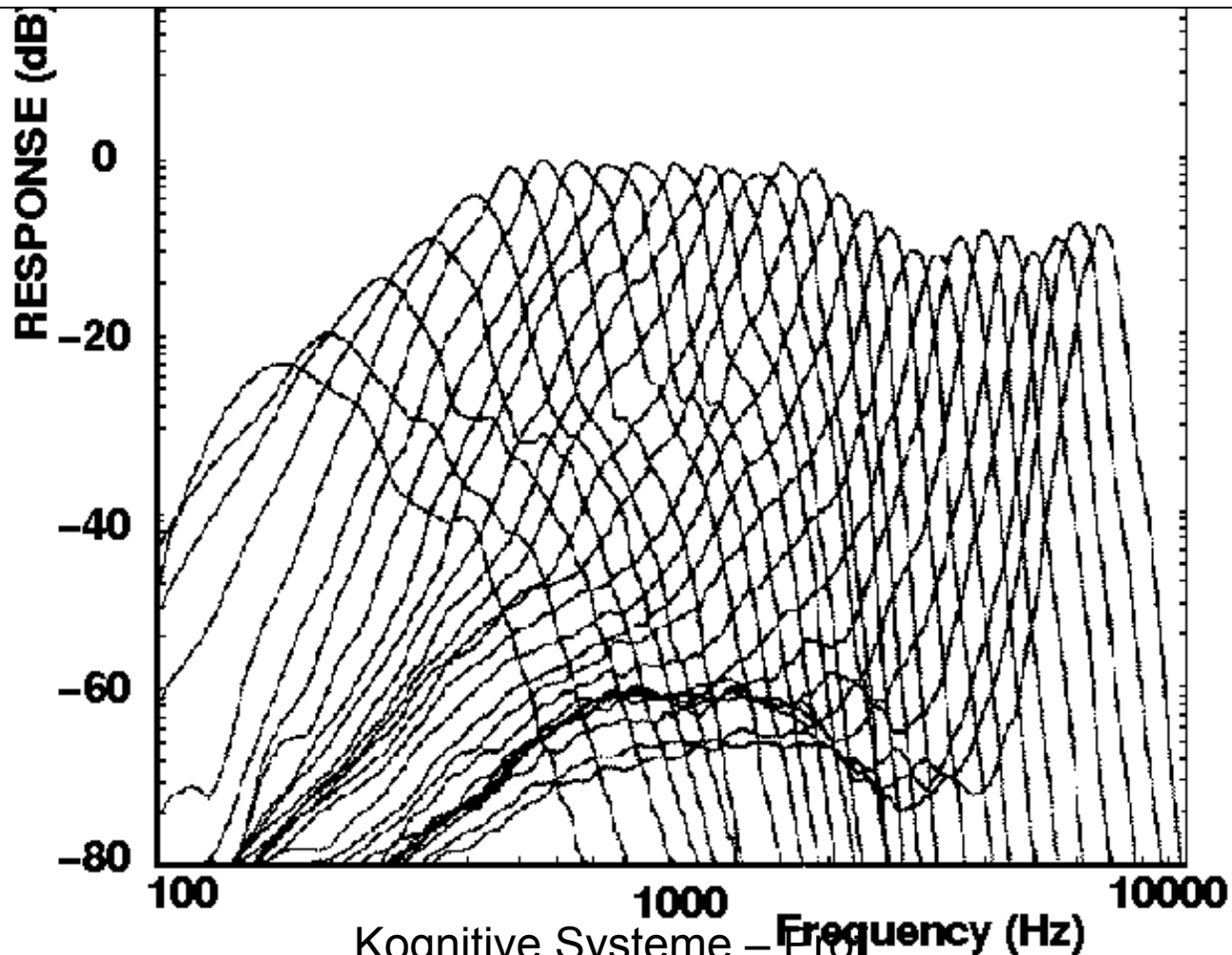
# Voiced and Unvoiced Phonemes



Voiced and Unvoiced Segments and their short-time spectra.

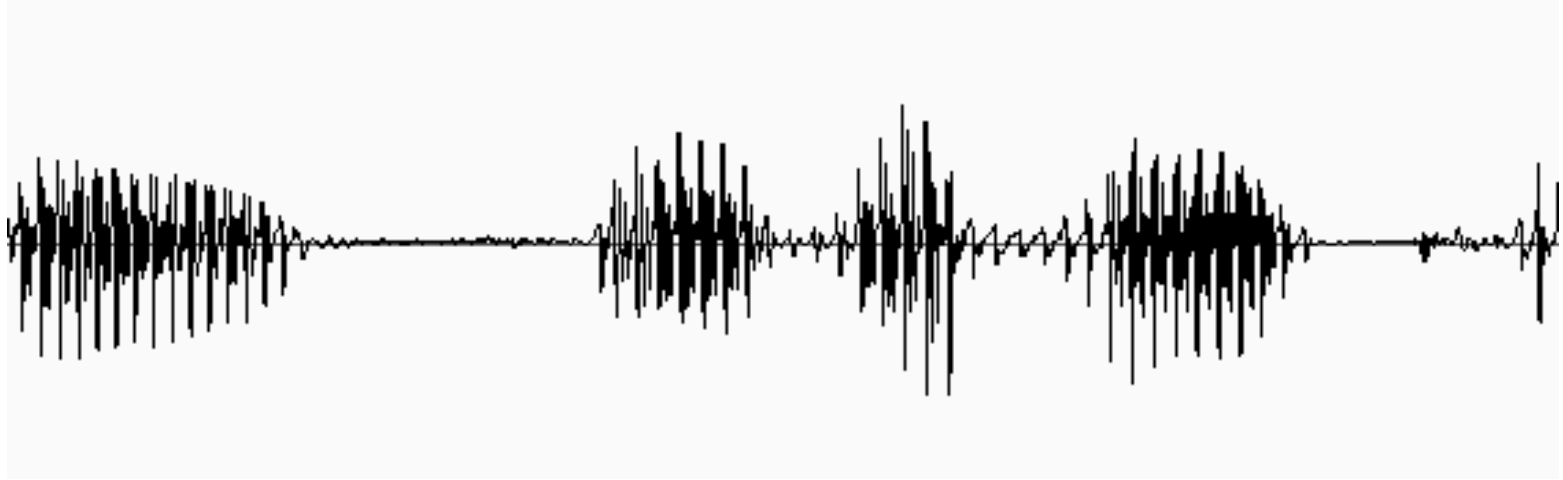


# Frequency Response of the Basilar Membrane



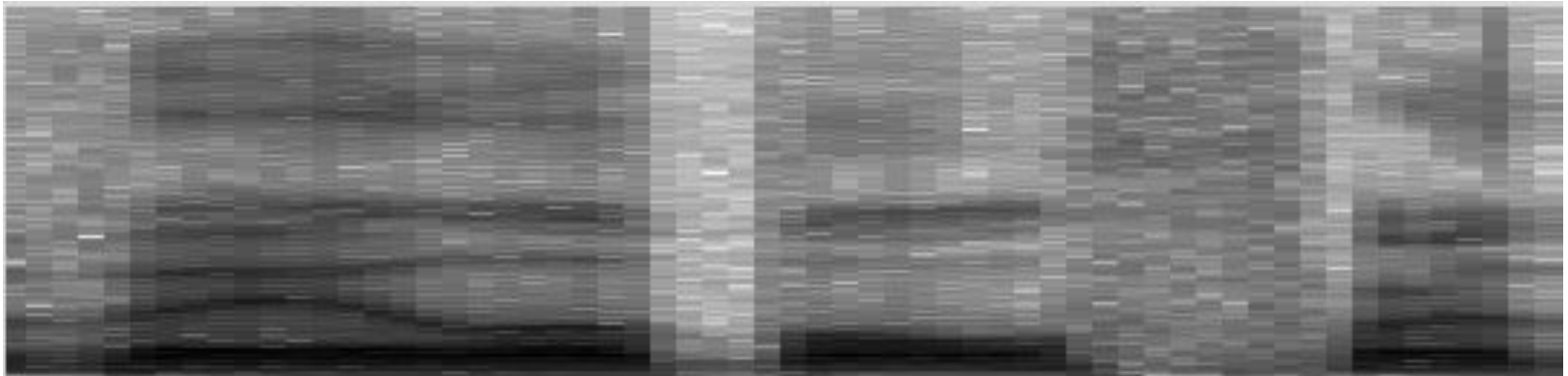
# Analog to Digital

- Sampled wave form
- Sampling Rate, Sample Resolution, Example: 16 kHz, 16 bit



# Front End Processing

- Reduce influence of undesired components --> Accuracy
- Reduce amount of data --> Speed
- Spectrum Contains most Important Information
- Most Popular Candidates: Mel-Scale FilterBank, LPC, Cepstral Coeffs



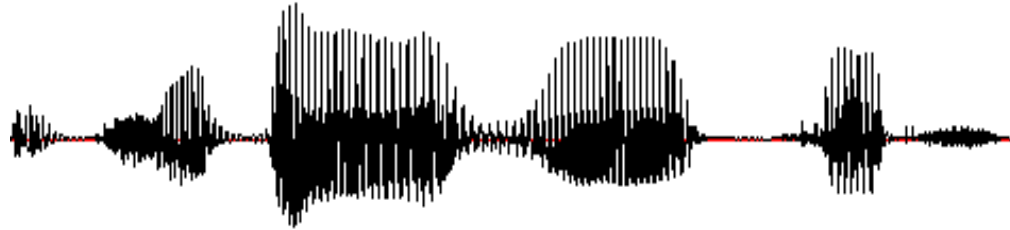
# Typical Steps

- Anti-Aliasing Filter
- AD Conversion
- Windowing
- FFT
- Compute Power-Spectrum
- Mel-Scale Filter Bank Coefficients
- Or:
  - Compute LPC or Cepstral Coefficients

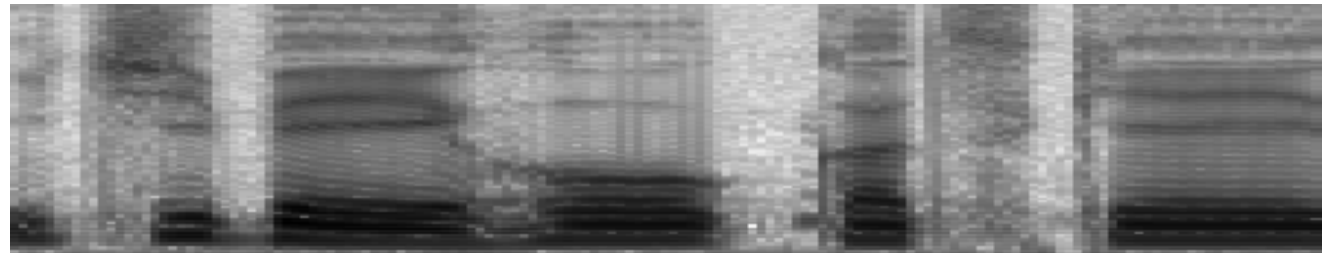


# Front End Preprocessing

Recording:



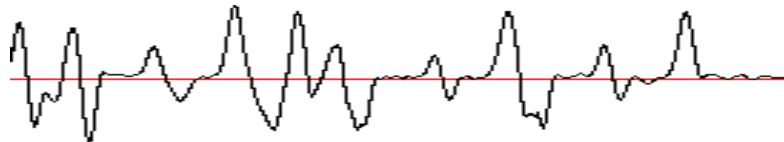
Spectrum:



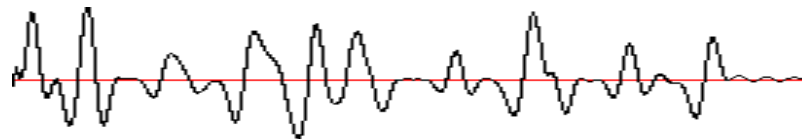
Power:



$\Delta$ -Power:



$\Delta\Delta$ -Power:



# Linear Sequence Alignment

First idea:

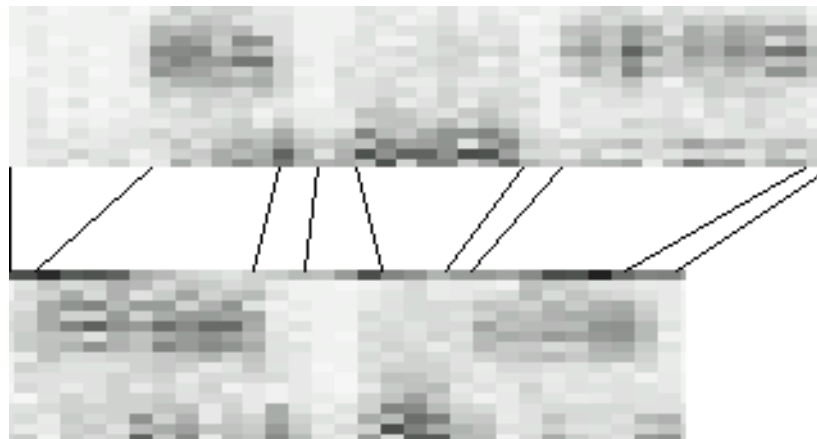
- Determine a Linear Alignment between the Frames of the Unknown Pattern and a Reference Pattern
- Compute Distance of a Word Template to a Reference Template by Computing the Sum of all Frame-to-Frame Distances
- Repeat for each Word Template



# Problem with Linear Alignment

**Linear** alignment can handle the problem of different speaking rates. But it can not handle the problem of varying speaking rates during the same utterance.

We need **Non-Linear** Alignment

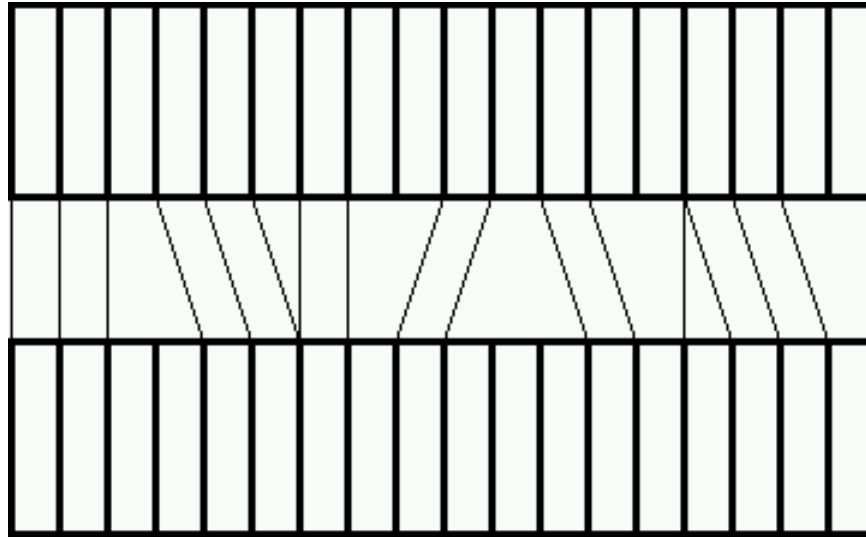


# Alignment of Speech Vectors May Be Non-Bijective

Task:

given: two sequences  $x_1, x_2, \dots, x_n$  and  $y_1, y_2, \dots, y_m$

wanted: alignment relation  $R$  (not function), where  $(i, j)$  is in  $R$  iff  $x_i$  is aligned with  $y_j$ .



It is possible that more than one  $x$  is aligned to the same  $y$  (or vice versa).

It is possible that more than an  $x$  or a  $y$  has no alignment partner at all.

# Time Warping

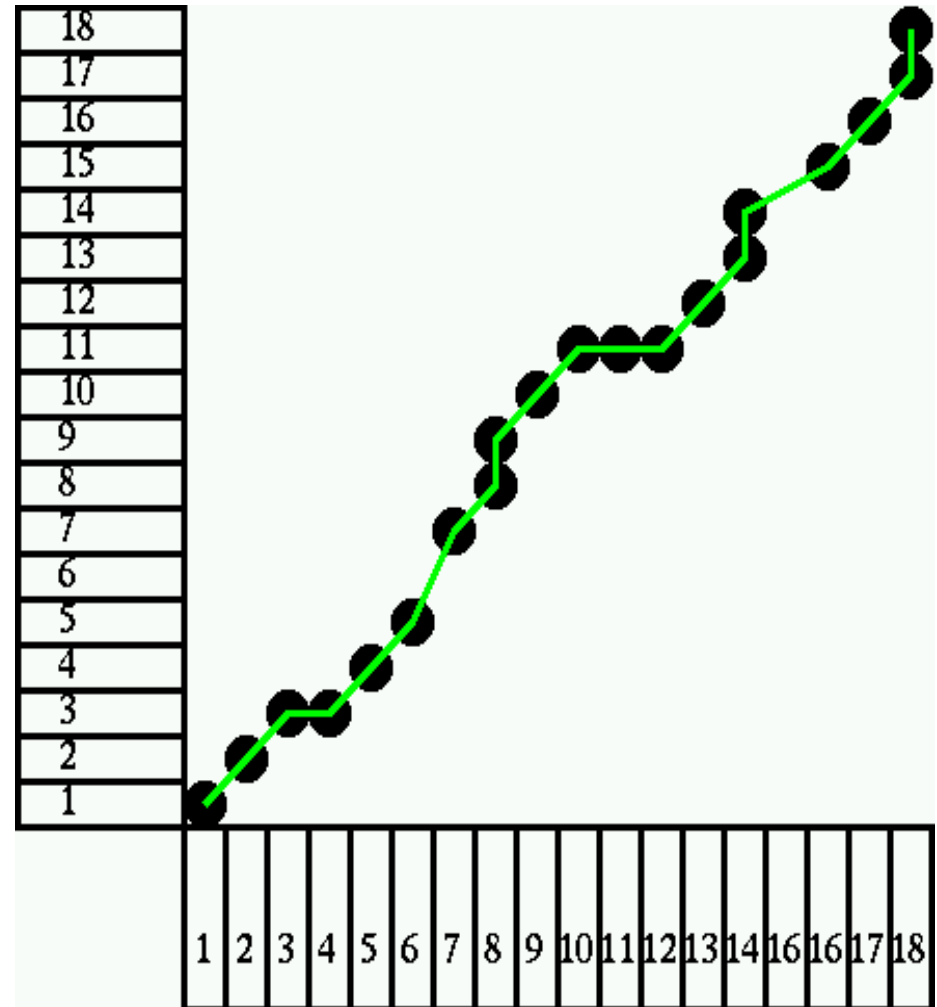
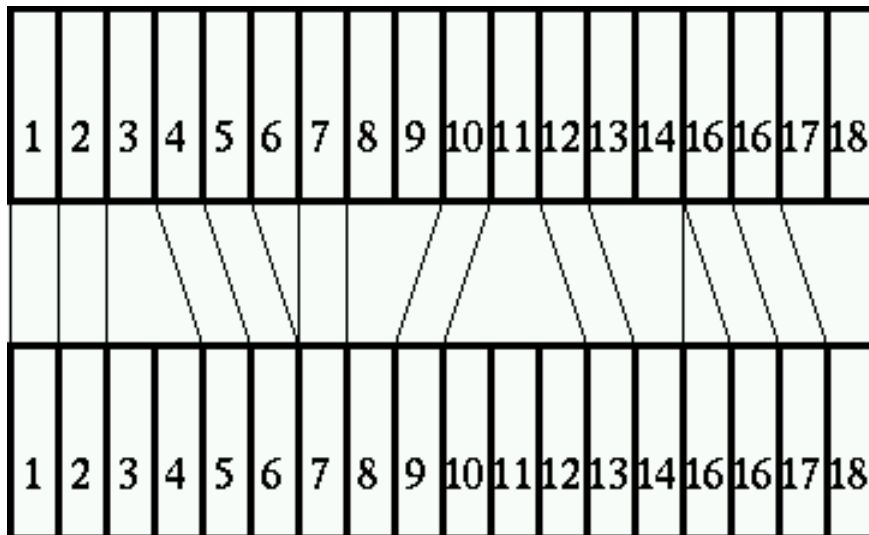
Task:

given: two sequences  $x_1, x_2, \dots, x_n$  and

$y_1, y_2, \dots, y_m$

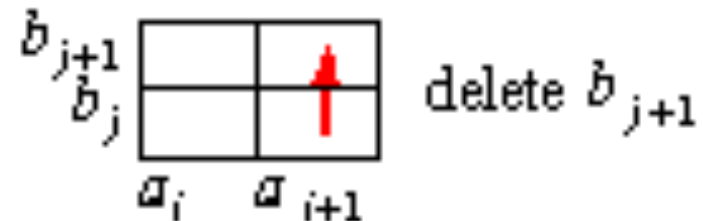
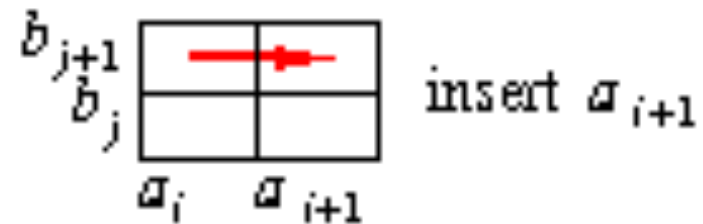
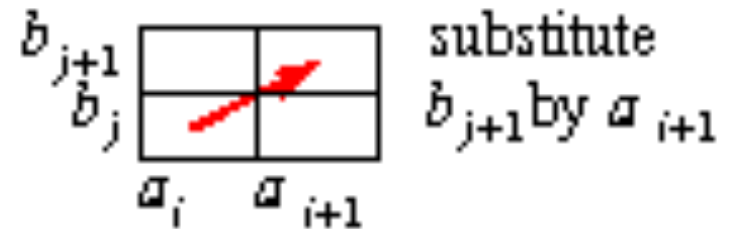
wanted: alignment relation  $R$  (not function), where  $(i, j)$  is in  $R$  iff  $x_i$  is aligned with  $y_j$ .

We are looking for a common time-axis:



# Dynamic Time Warping

- Paths
  - Insertion
- Start:
  - Initial condition at 0,0
- Inductive Step
  - Best previous path
- Solution:
  - Best path at the end





# Distance Measure between two Utterances

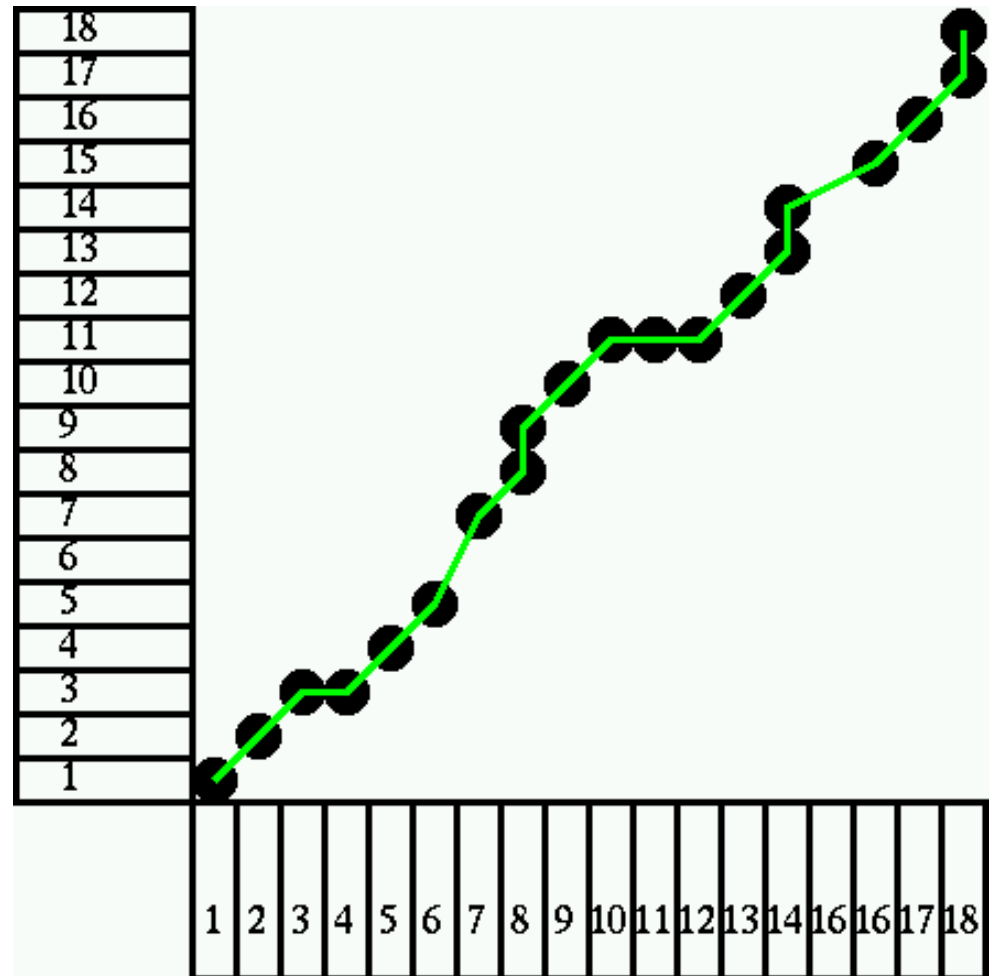
For a given **path**  $R(i,j)$ , the distance between  $x$  and  $y$  is the sum of all **local distances**  $d(x_i, y_j)$ .

In our example:

$$\begin{aligned} & d(x_1, y_1) + d(x_2, y_2) + d(x_3, y_3) + \\ & d(x_3, y_3) + d(x_5, y_4) + d(x_6, y_5) + \\ & d(x_7, y_7) + \dots \end{aligned}$$

**Question:**

How can we find a path that gives the minimal overall distance?



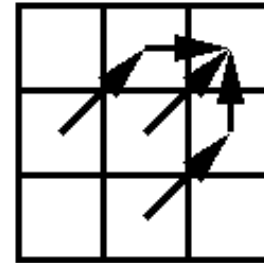
# DTW-Steps

Many different warping steps are possible and have been used.

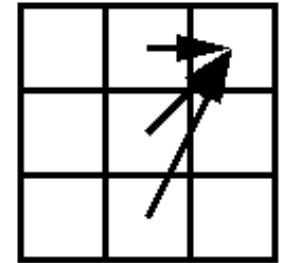
Examples:

General rule is:  
Cumulative cost of destination =  
*best-of*(cumulative cost of source + cost of step + distance in destination)

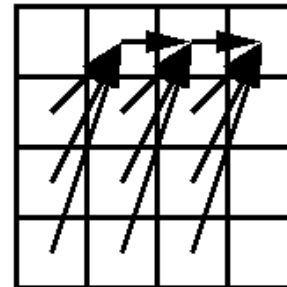
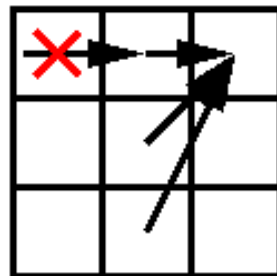
symmetric  
(editina distance)



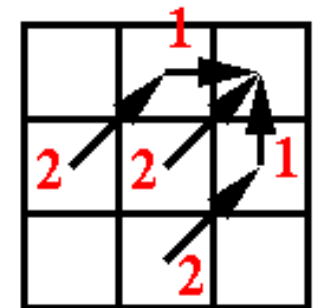
Bakis



Itakura

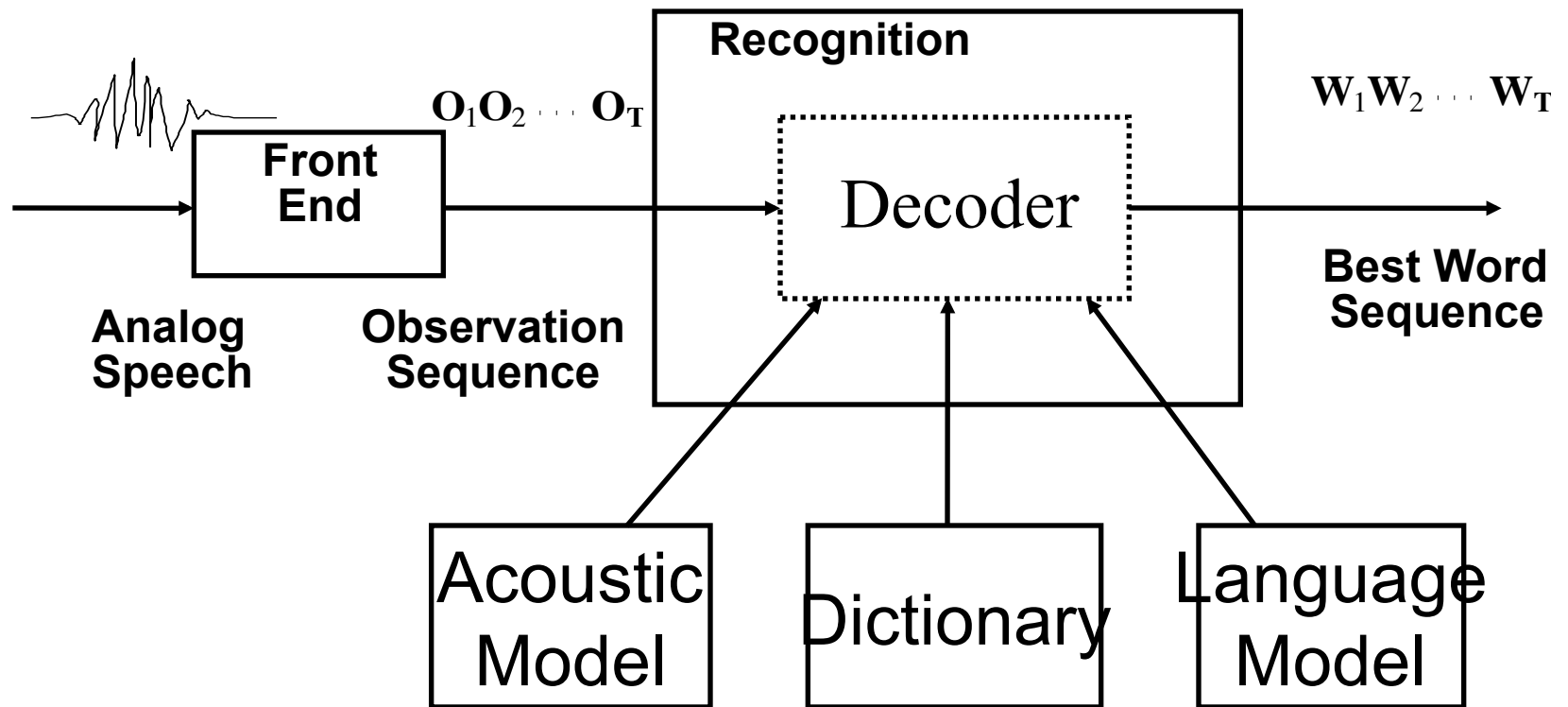


weighted



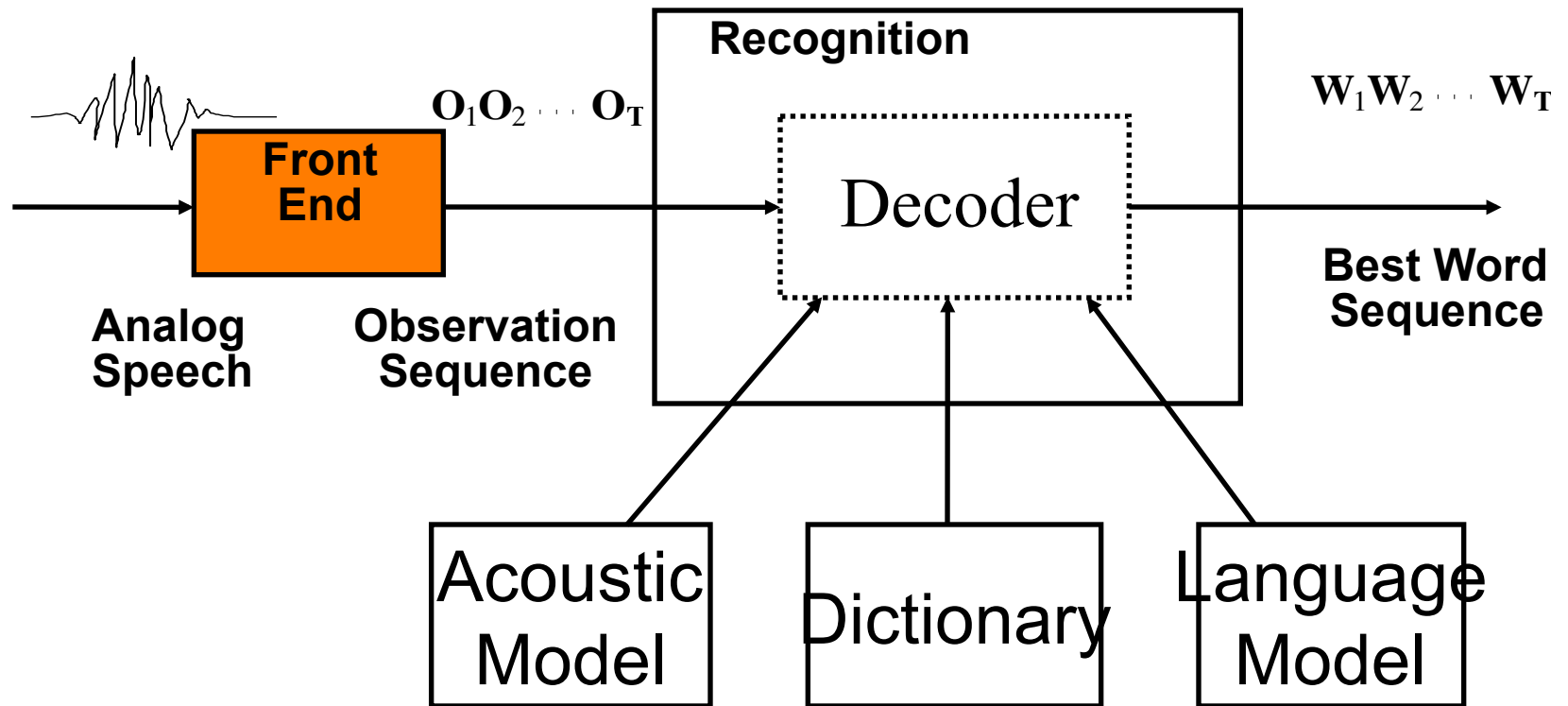
# Speech Recognition (Components)

- Recognizer Components:

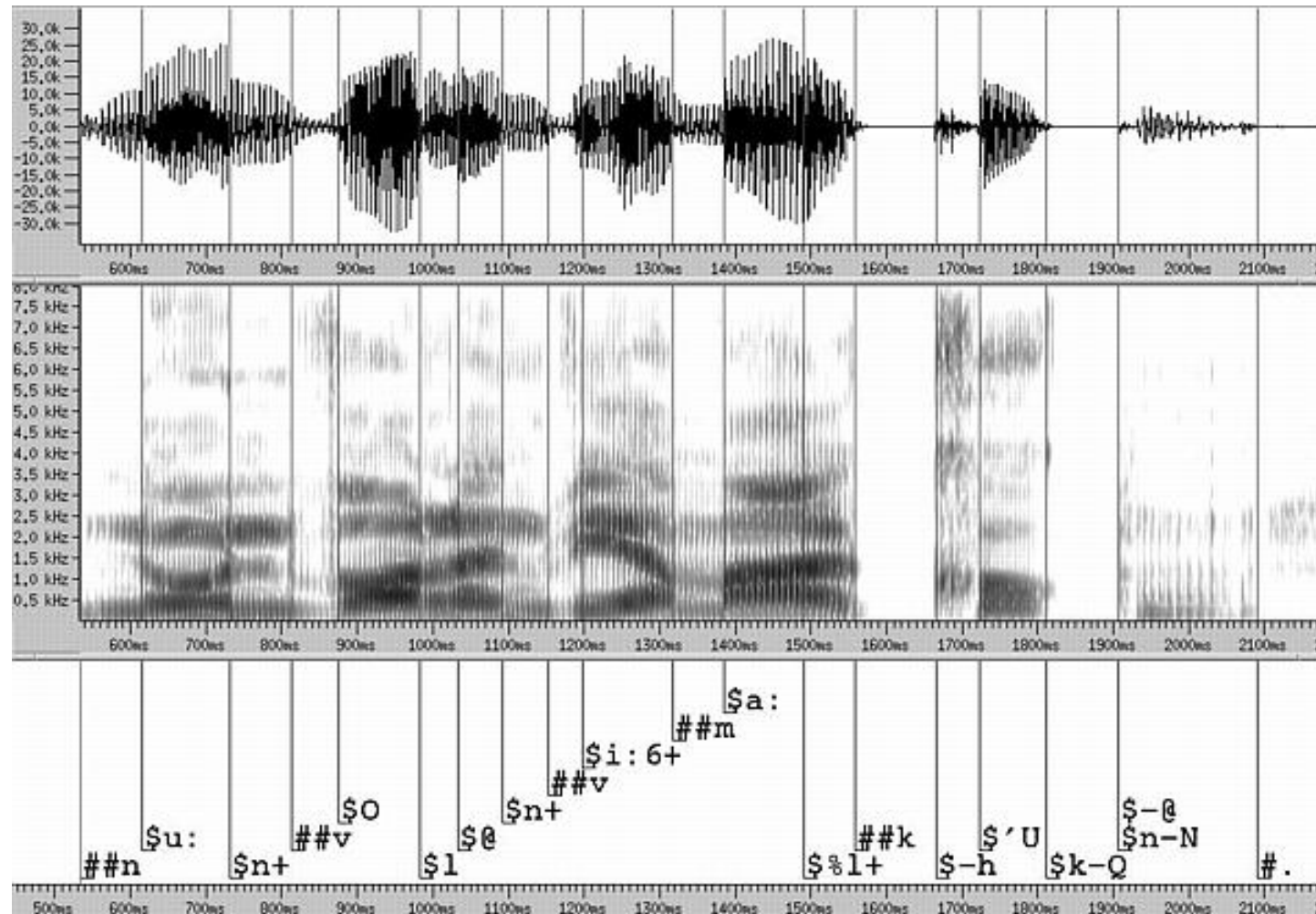


# Speech Recognition (System Components)

- Recognizer Components:



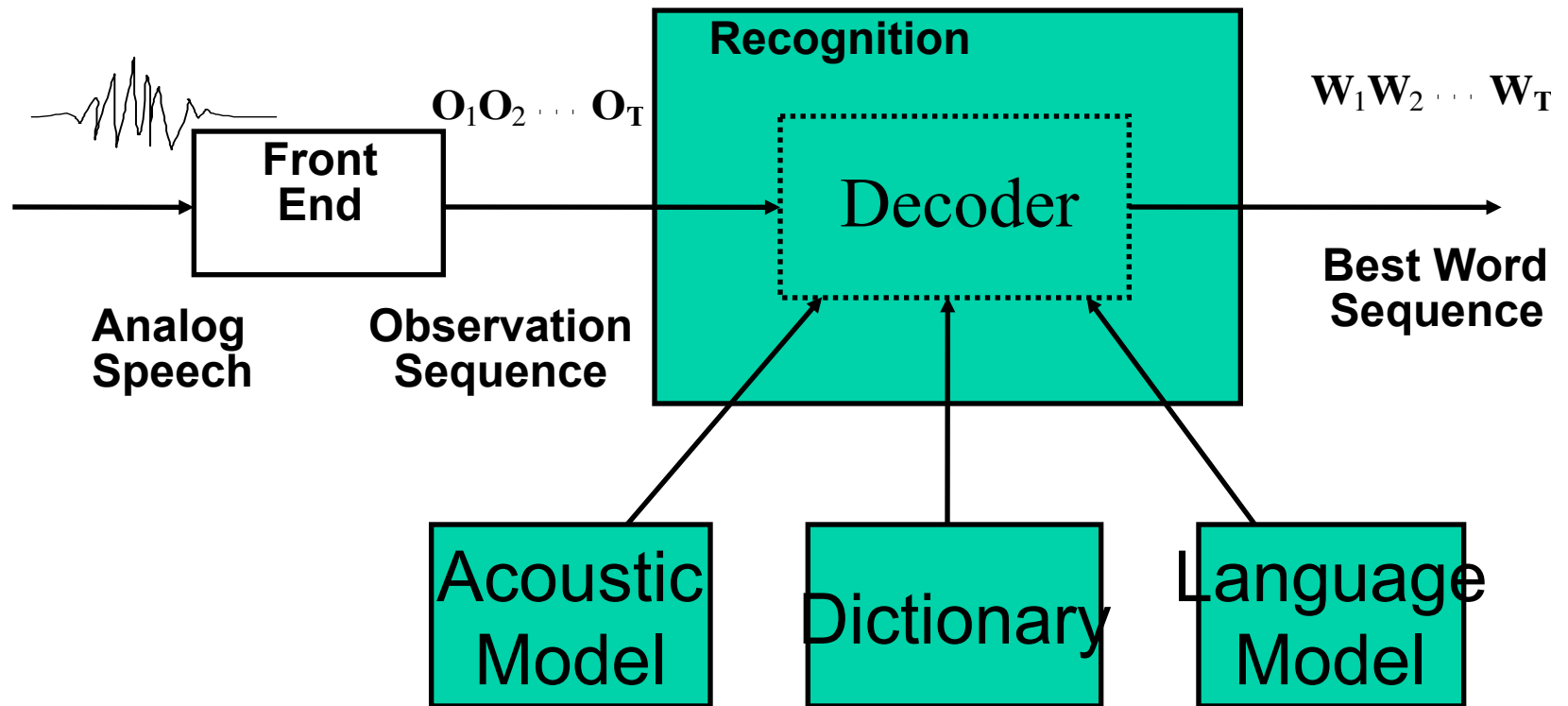
# Spectrogram



“Nu Wollen Wir Mal Gucken”

# Speech Recognition (System Components)

- Recognizer Components:





# Speech Recognition

- Goal:
  - Given acoustic data  $A = a_1, a_2, \dots, a_k$
  - Find word sequence  $W = w_1, w_2, \dots, w_n$
  - Such that  $P(W | A)$  is maximized

## Bayes Rule:

$$P(W | A) = \frac{P(A | W) \cdot P(W)}{P(A)}$$

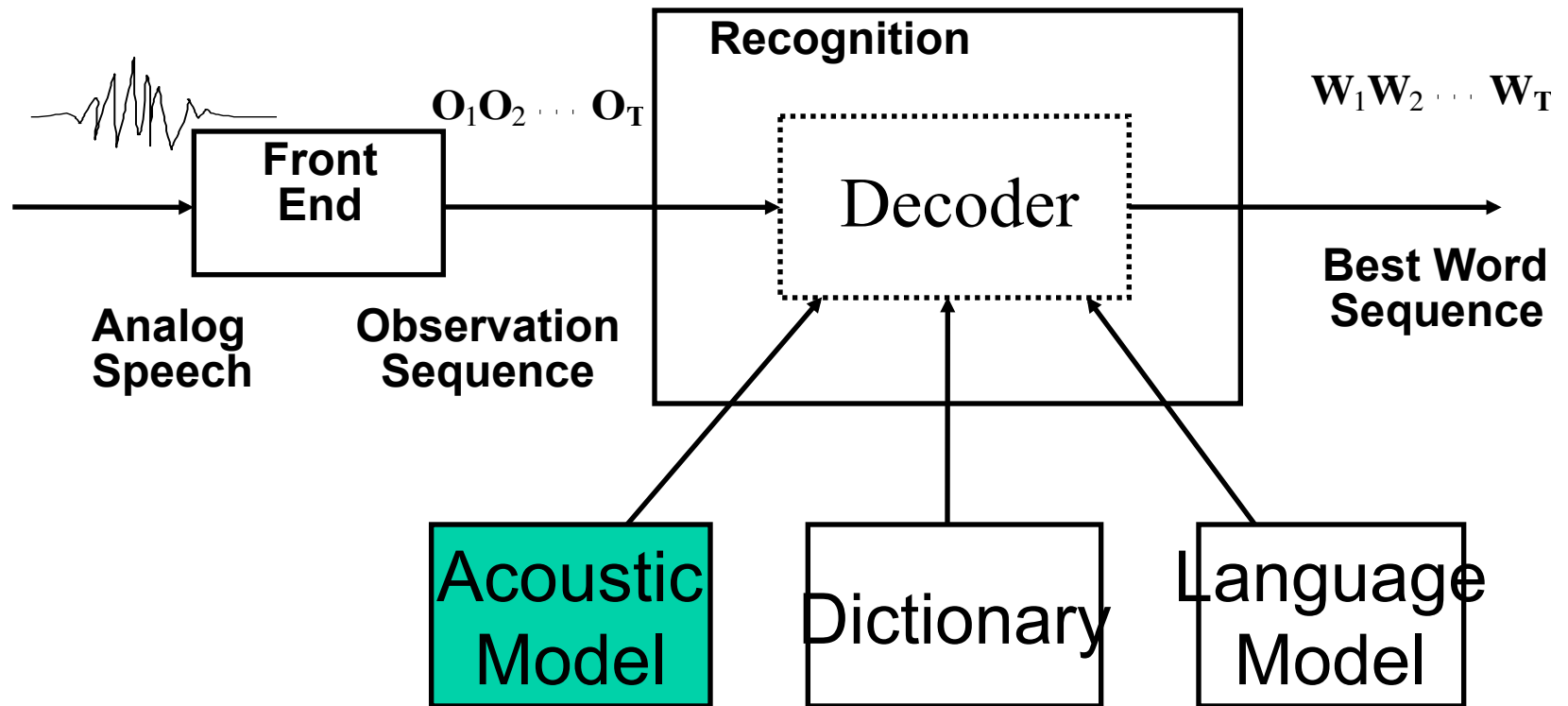
acoustic model (HMMs)  $\swarrow$

$\nwarrow$  language model

**$P(A)$  is a constant for a complete sentence**

# Speech Recognition (Components)

- Recognizer Components:



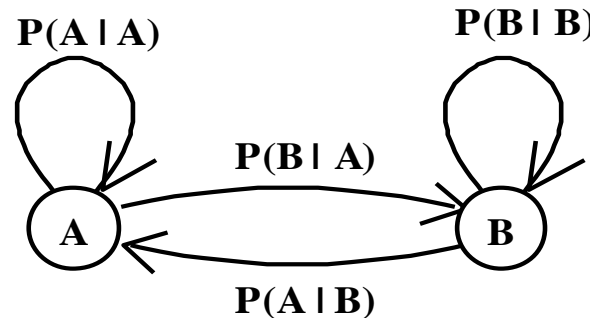
# Markov Models

**Elements:**

**States :**

**Transition probabilities :**

$$\mathbf{S} = \{S_0, S_1, \dots, S_N\}$$
$$P(q_t = S_i \mid q_{t-1} = S_j)$$



**Markov Assumption:**

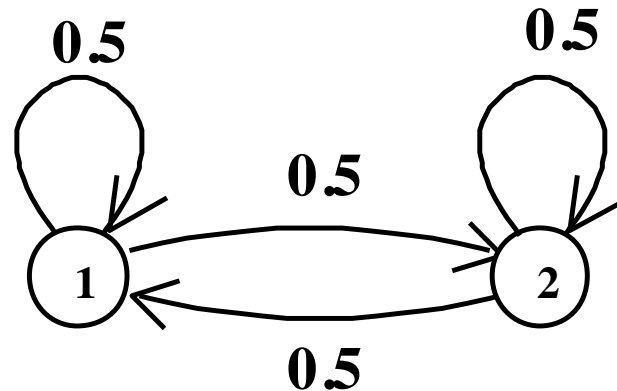
**Transition probability depends only on current state**

$$P(q_t = S_i \mid q_{t-1} = S_j, q_{t-2} = S_k, \dots) = P(q_t = S_i \mid q_{t-1} = S_j) = a_{ji}$$

$$a_{ji} \geq 0 \quad \forall j, i$$
$$\sum_{i=0}^N a_{ji} = 1 \quad \forall j$$

# Single Fair Coin

- Outcome head corresponds to state 1, tail to state 2
- Observation sequence uniquely defines state sequence



$$P(H) = 1.0$$

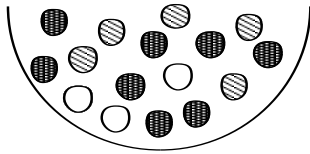
$$P(T) = 0.0$$

$$P(H) = 0.0$$

$$P(T) = 1.0$$

# Discrete Observation HMM

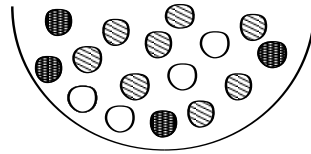
- Observation sequence: RRRBYBBYYY ... R
- not unique to state sequence



$$P(R) = 0.31$$

$$P(B) = 0.50$$

$$P(Y) = 0.19$$

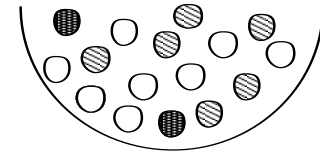


$$P(R) = 0.50$$

$$P(B) = 0.25$$

$$P(Y) = 0.25$$

...



$$P(R) = 0.38$$

$$P(B) = 0.12$$

$$P(Y) = 0.50$$

# Hidden Markov Models

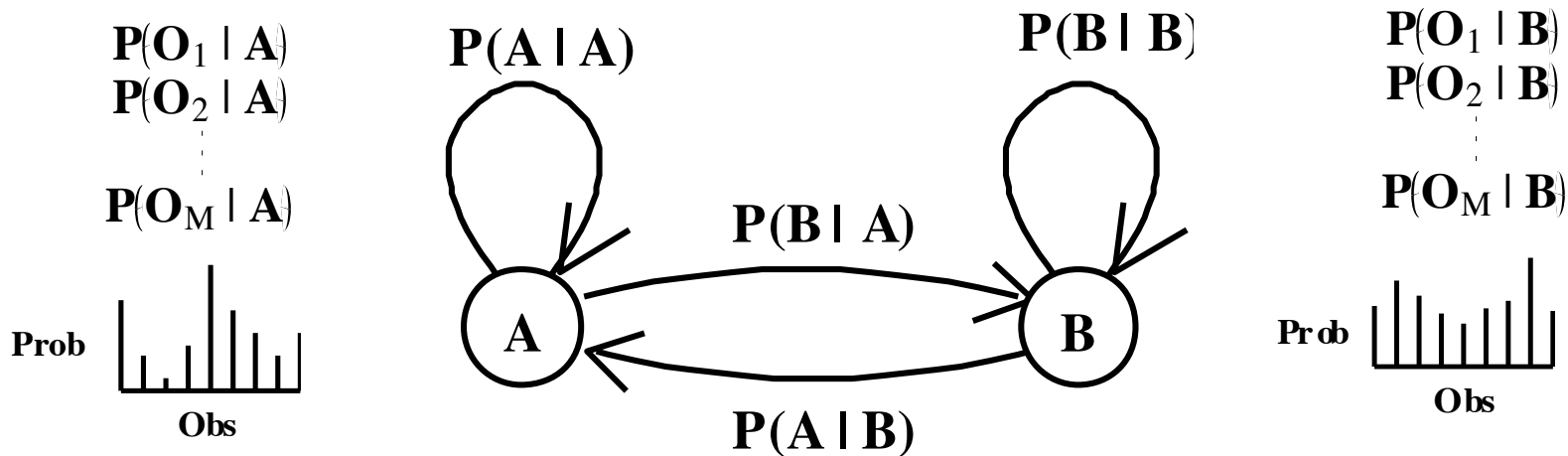
- Elements:

- States
- Transition probabilities
- Output prob distributions (at state  $j$  for symbol  $k$ )

$$S = \{S_0, S_1, \dots, S_N\}$$

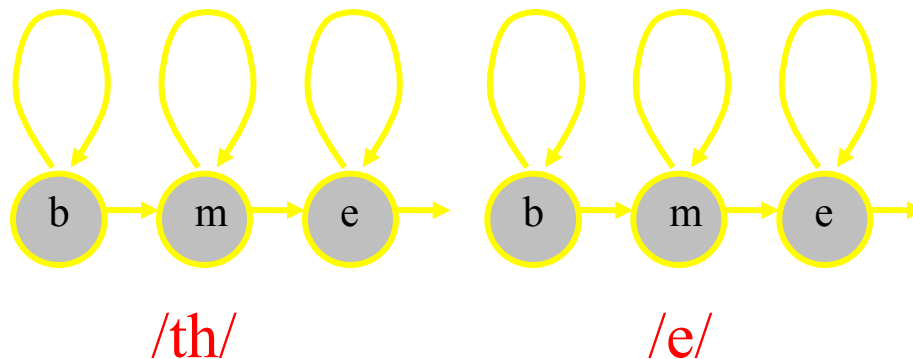
$$P(q_t = S_i \mid q_{t-1} = S_j) = a_{ji}$$

$$P(y_t = O_k \mid q_t = S_j) = b_j(k)$$



# Acoustic Modeling

- Hidden Markov Models:
  - Words, Phonemes, States
  - Observation Probability:  $P(\text{feature vector} \mid \text{state})$





# HMM Problems And Solutions

- Evaluation:
  - Problem - Compute Probability of observation sequence given a model
  - Solution - **Forward Algorithm** and **Viterbi Algorithm**
- Decoding:
  - Problem - Find state sequence which maximizes probability of observation sequence
  - Solution - **Viterbi Algorithm**
- Training:
  - Problem - Adjust model parameters to maximize probability of observed sequences
  - Solution - **Forward-Backward Algorithm**

# Evaluation

Probability of observation sequence  $\mathbf{O} = O_1 O_2 \dots O_T$   
given HMM model  $\lambda$  is :

$$P(\mathbf{O} | \lambda) = \sum_{\forall \mathbf{Q}} P(\mathbf{O}, \mathbf{Q} | \lambda)$$

where  $\mathbf{Q} = q_0 q_1 \dots q_T$  is a sequence of states

$$= \sum_{\forall q_0, \dots, q_T} a_{q_0 q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) \dots a_{q_{T-1} q_T} b_{q_T}(O_T)$$

**Not practical since the number of paths is  $O(N^T)$**

where  $N$  = number of states in model  
and  $T$  = number of observations in sequence

# The Forward Algorithm

$$\alpha_t(j) = P(O_1 O_2 \cdots O_t, q_t = S_j \mid \lambda)$$

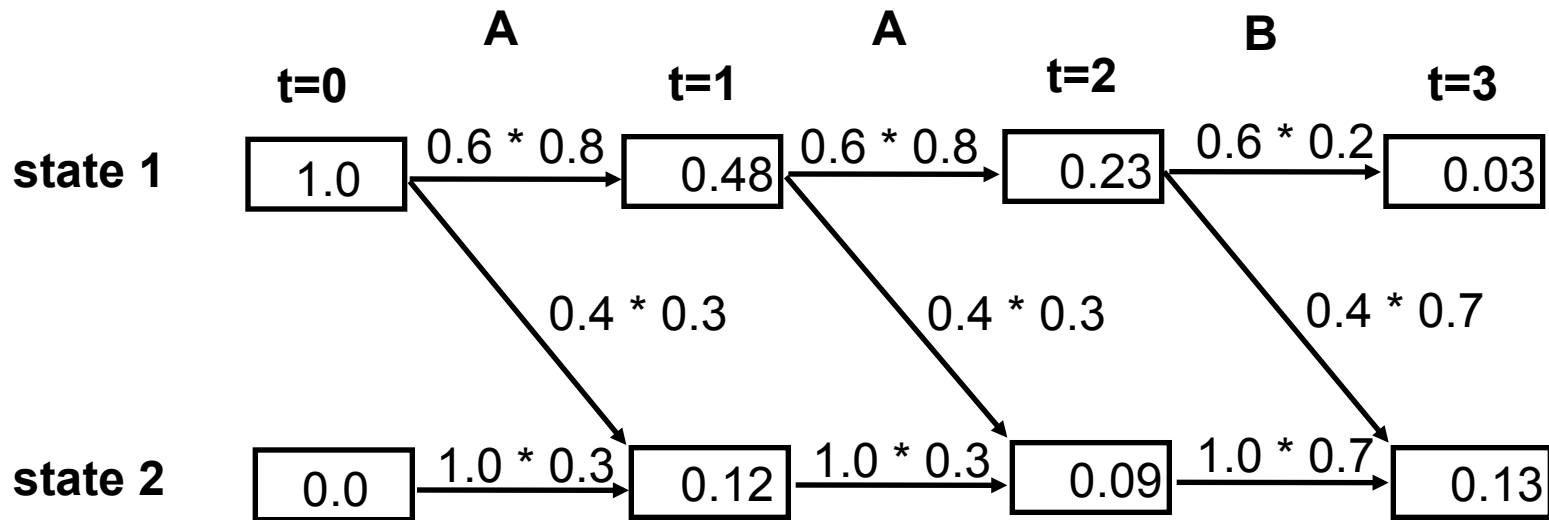
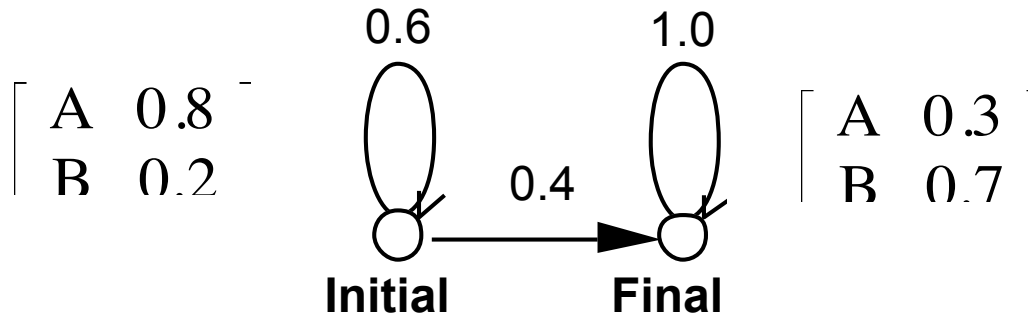
Compute  $\alpha$  recursively:

$$\alpha_0(j) = \begin{cases} 1 & \text{if } j \text{ is start state} \\ 0 & \text{otherwise} \end{cases}$$

$$\alpha_t(j) = \left[ \sum_{i=0}^N \alpha_{t-1}(i) a_{ij} \right] b_j(O_t) \quad t > 0$$

$$P(O \mid \lambda) = \alpha_T(S_N) \quad \text{Computation is } O(N^2 T)$$

# Forward Trellis



# The Backward Algorithm

$$\beta_t(i) = P(O_{t+1} O_{t+2} \dots O_T, q_t = S_i \mid \lambda)$$

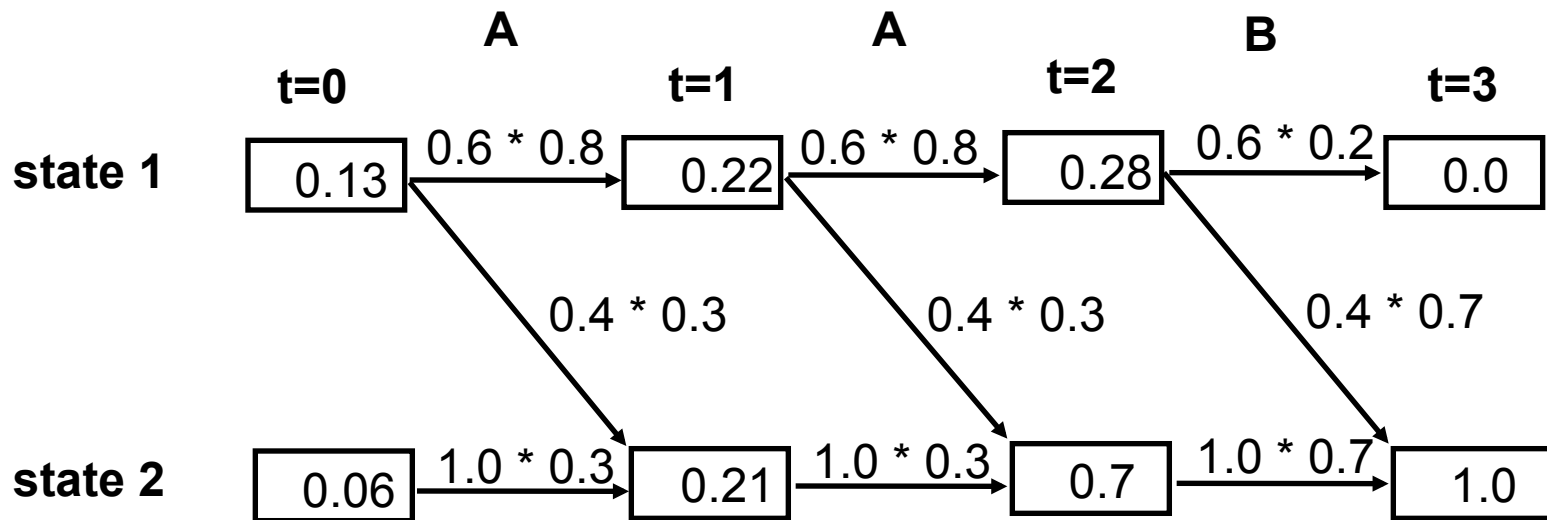
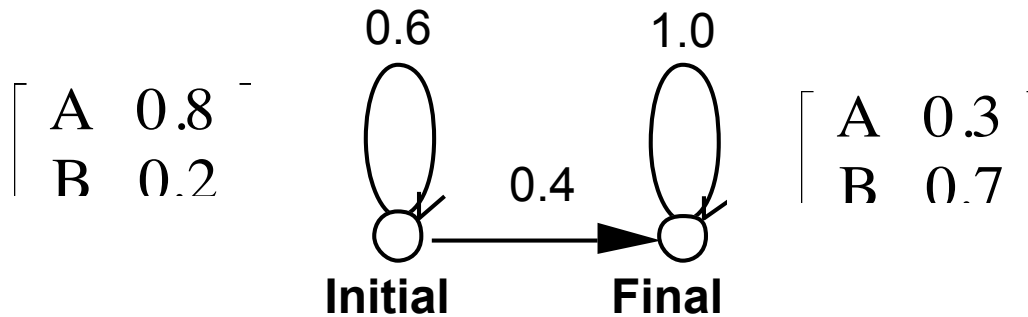
Compute  $\alpha$  recursively:

$$\beta_T(i) = \begin{cases} 1 & \text{if } i \text{ is end state} \\ 0 & \text{otherwise} \end{cases}$$

$$\beta_t(i) = \sum_{j=0}^N a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) \quad t < T$$

$$P(O \mid \lambda) = \beta_0(S_0) = \alpha_T(S_N) \quad \text{Computation is } O(N^2 T)$$

# Backward Trellis



# HMM Problems And Solutions

- Evaluation:
  - Problem - Compute Probability of observation sequence given a model
  - Solution - **Forward Algorithm** and **Viterbi Algorithm**
- Decoding:
  - Problem - Find state sequence which maximizes probability of observation sequence
  - Solution - **Viterbi Algorithm**
- Training:
  - Problem - Adjust model parameters to maximize probability of observed sequences
  - Solution - **Forward-Backward Algorithm**



# Decoding

The Viterbi Algorithm:

- Find the state sequence **Q** which maximizes **P(O, Q | λ)**
- Similar to Forward Algorithm except **MAX** instead of **SUM**

$$VP_t(i) = \text{MAX}_{q_0, \dots, q_{t-1}} P(O_1 O_2 \dots O_t, q_t=i | \lambda)$$

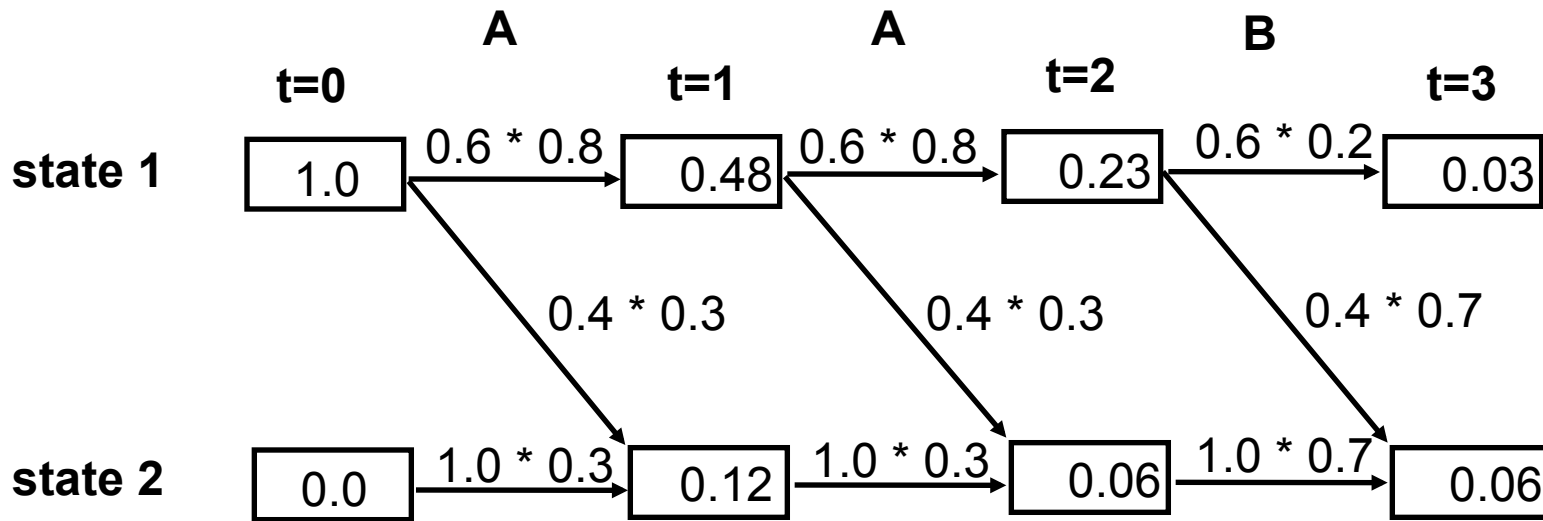
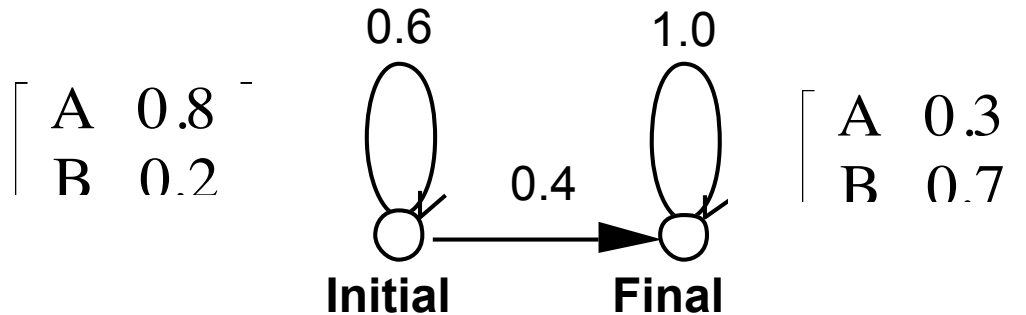
Recursive Computation:

$$VP_t(j) = \text{MAX}_{i=0, \dots, N} VP_{t-1}(i) a_{ij} b_j(O_t) \quad t > 0$$

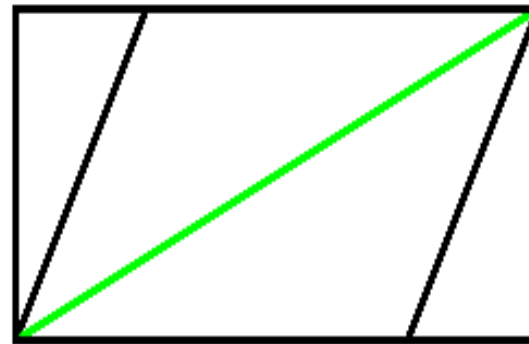
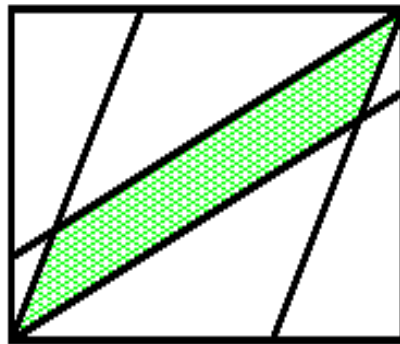
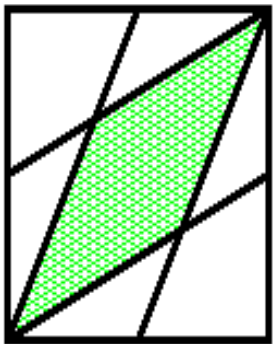
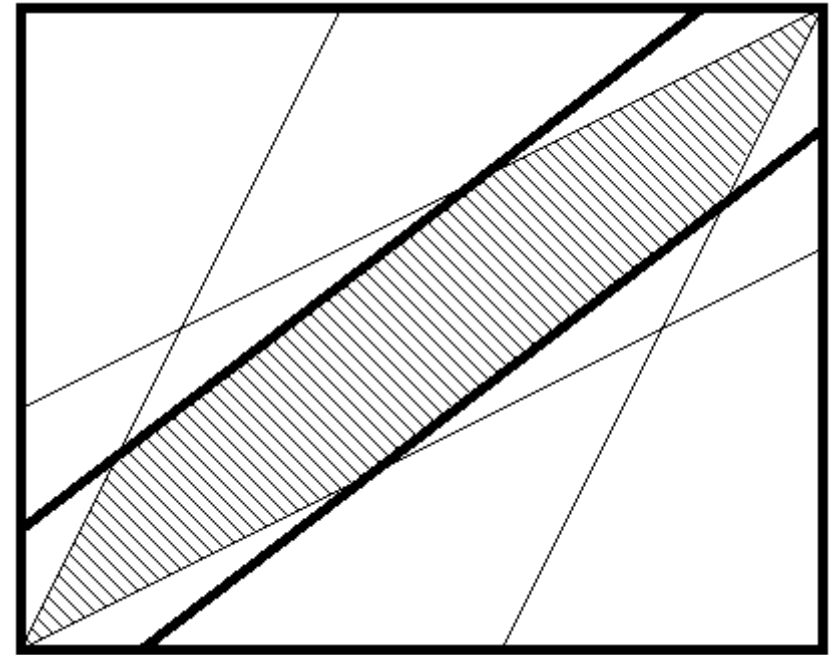
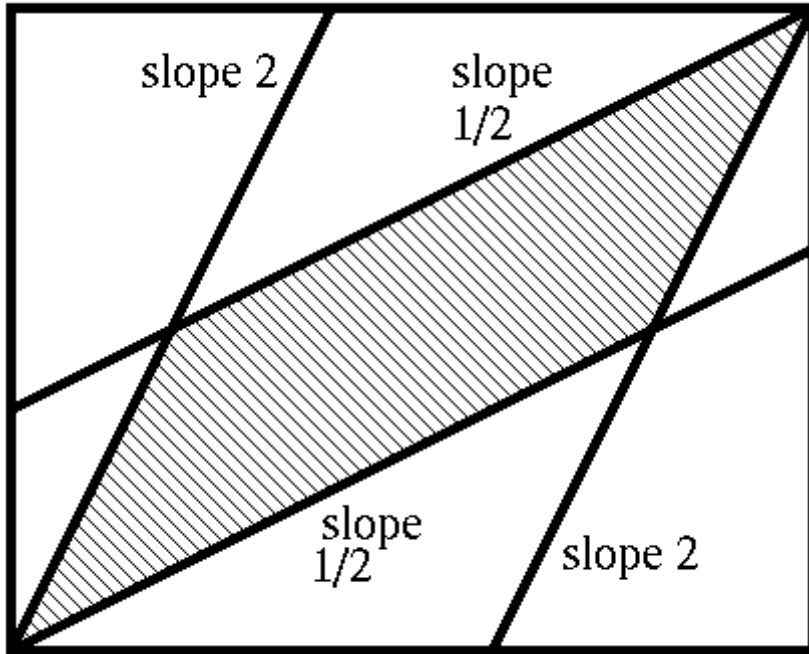
$$P(O, Q | \lambda) = VP_T(S_N)$$

Save each maximum for backtrace at end

# Viterbi Trellis



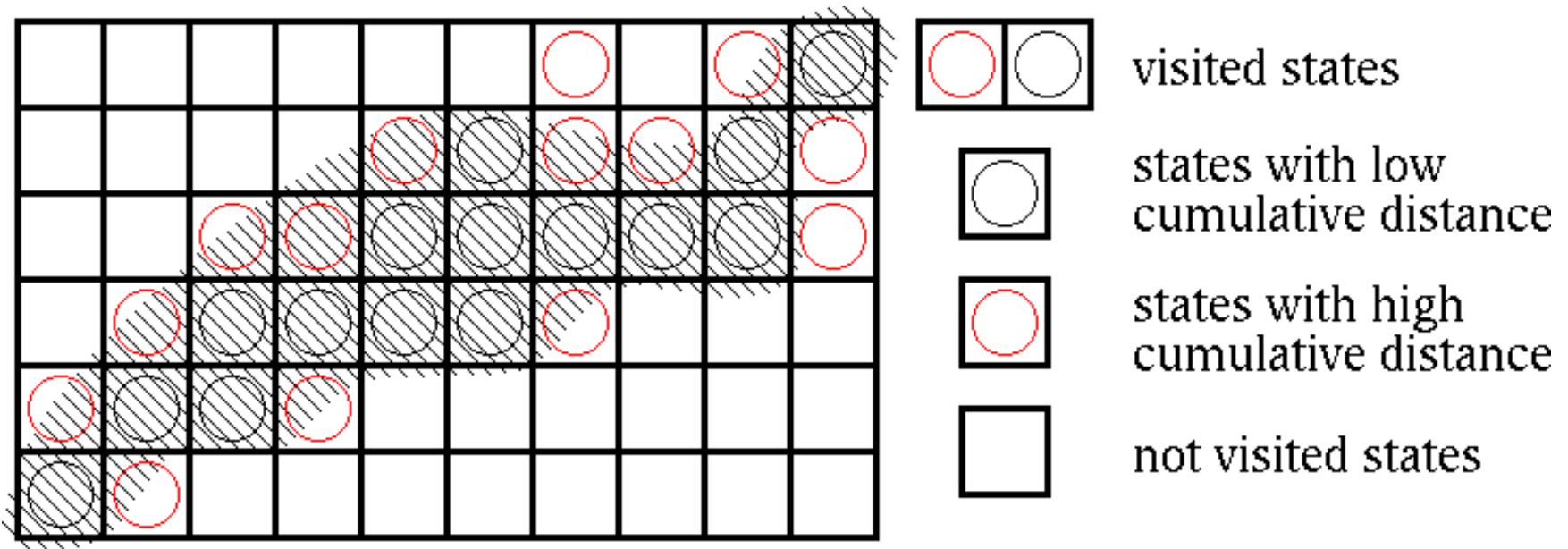
# Global Constraints for the DTW-Path



<- only one path

# Decoding with Beam Search

Idea:  
do not consider steps to be possible out of states that have "too high" cumulative distances.

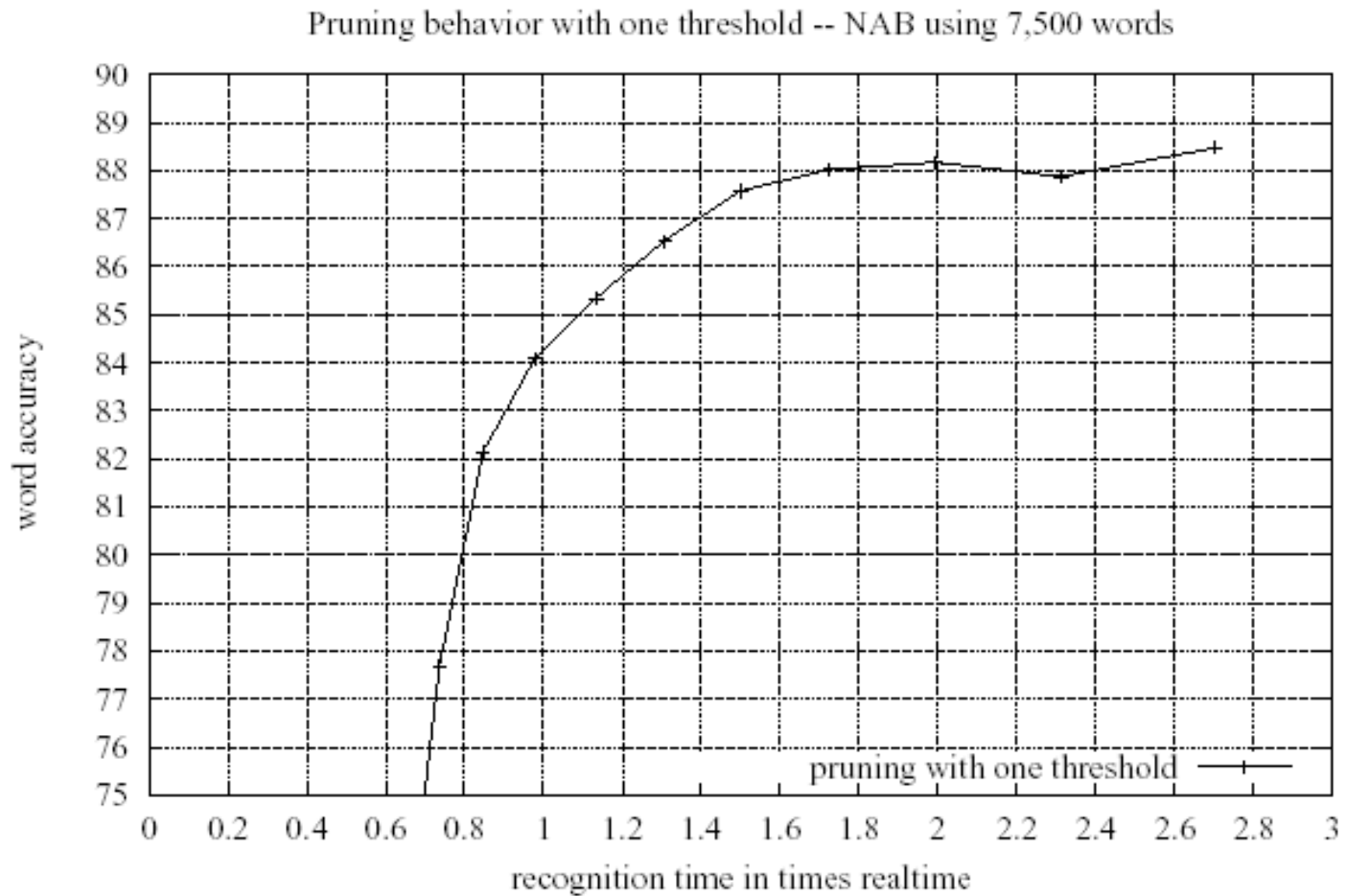


Approaches:

- "expand" only a fixed number of states per column of DTW-matrix
- expand only states that have a cumulative distance less than a factor (the beam) times the best distance so far

# Beam vs. WER

- Beispiel NAB, Beam von 170 bis 230



# HMM Problems And Solutions

- Evaluation:
  - Problem - Compute Probability of observation sequence given a model
  - Solution - **Forward Algorithm** and **Viterbi Algorithm**
- Decoding:
  - Problem - Find state sequence which maximizes probability of observation sequence
  - Solution - **Viterbi Algorithm**
- Training:
  - Problem - Adjust model parameters to maximize probability of observed sequences
  - Solution - **Forward-Backward Algorithm**

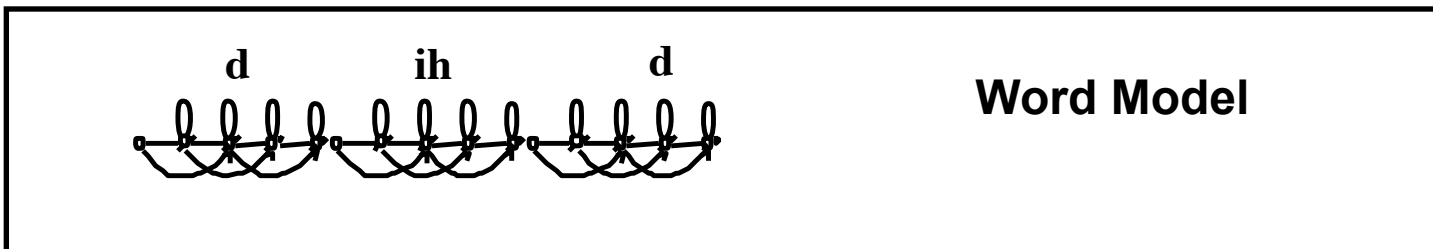
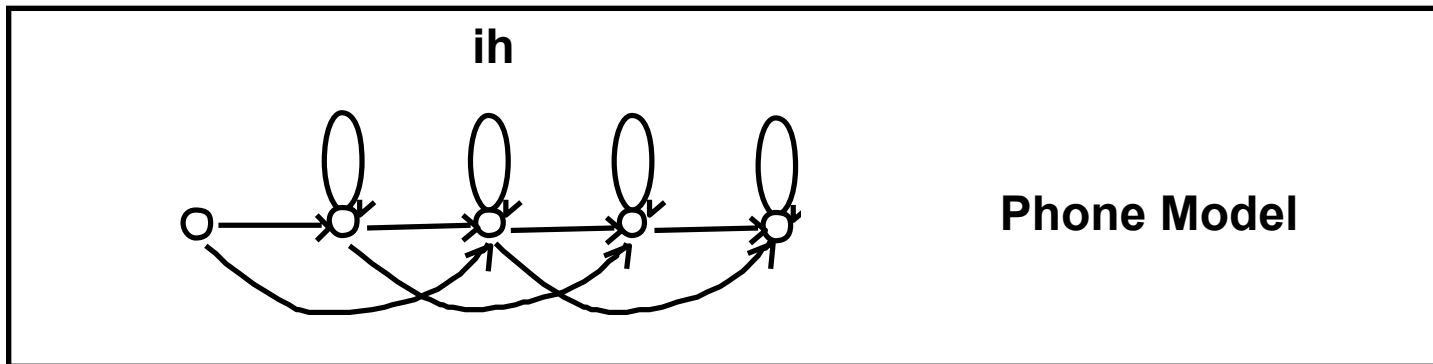
# Training HMM Parameters

- Train parameters of HMM
  - Tune  $\lambda$  to maximize  $P(O \mid \lambda)$
  - No efficient algorithm for global optimum
  - Efficient iterative algorithm finds a local optimum
- Viterbi-Training
  - Compute Viterbi-Path using Current Model
  - Reestimate Parameters Using Labels Assigned by Viterbi
- Baum-Welch (Forward-Backward) reestimation
  - Compute probabilities using current model
  - Refine  $\bar{\lambda} \rightarrow \lambda$  based on computed values
  - Use  $\alpha$  and  $\beta$  from Forward-Backward



# HMMs In Speech Recognition

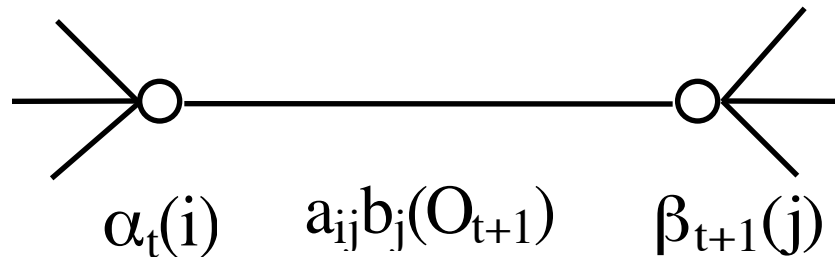
- Represent speech as a sequence of observations
- Use HMM to model some unit of speech (phone, word)
- Concatenate units into larger units



# Forward-Backward Algorithm

- Probability of transiting from  $S_i$  to  $S_j$  at time  $t$  given  $O$

$$\begin{aligned}\xi_t(i,j) &= P(q_t = S_i, q_{t+1} = S_j \mid O, \lambda) \\ &= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O \mid \lambda)}\end{aligned}$$



# Baum-Welch Reestimation

$$\bar{a}_{ij} = \frac{\text{expected number of trans from } S_i \text{ to } S_j}{\text{expected number of trans from } S_i}$$

$$= \frac{\sum_{t=1}^T \xi_t(i,j)}{\sum_{t=1}^T \sum_{j=0}^N \xi_t(i,j)}$$

$$\bar{b}_j(k) = \frac{\text{expected number of times in state } j \text{ with symbol } k}{\text{expected number of times in state } j}$$

$$= \frac{\sum_{t: O_t=k} \sum_{i=0}^N \xi_t(i,j)}{\sum_{t=1}^T \sum_{i=0}^N \xi_t(i,j)}$$

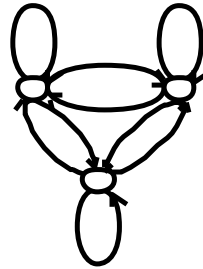
# Convergence of FB Algorithm

1. Initialize  $\lambda = (A, B)$
2. Compute  $\alpha$ ,  $\beta$ , and  $\xi$
3. Estimate  $\bar{\lambda} = (\bar{A}, \bar{B})$  from  $\xi$
4. Replace  $\lambda$  with  $\bar{\lambda}$
5. If not converged go to 2

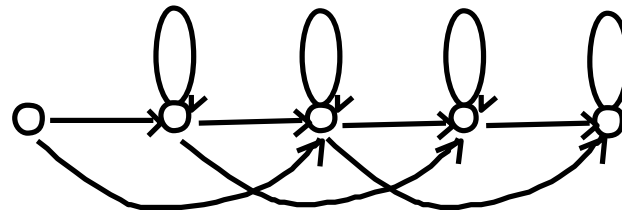
It can be shown that  $P(O | \bar{\lambda}) > P(O | \lambda)$  unless  $\bar{\lambda} = \lambda$

# Model Topologies

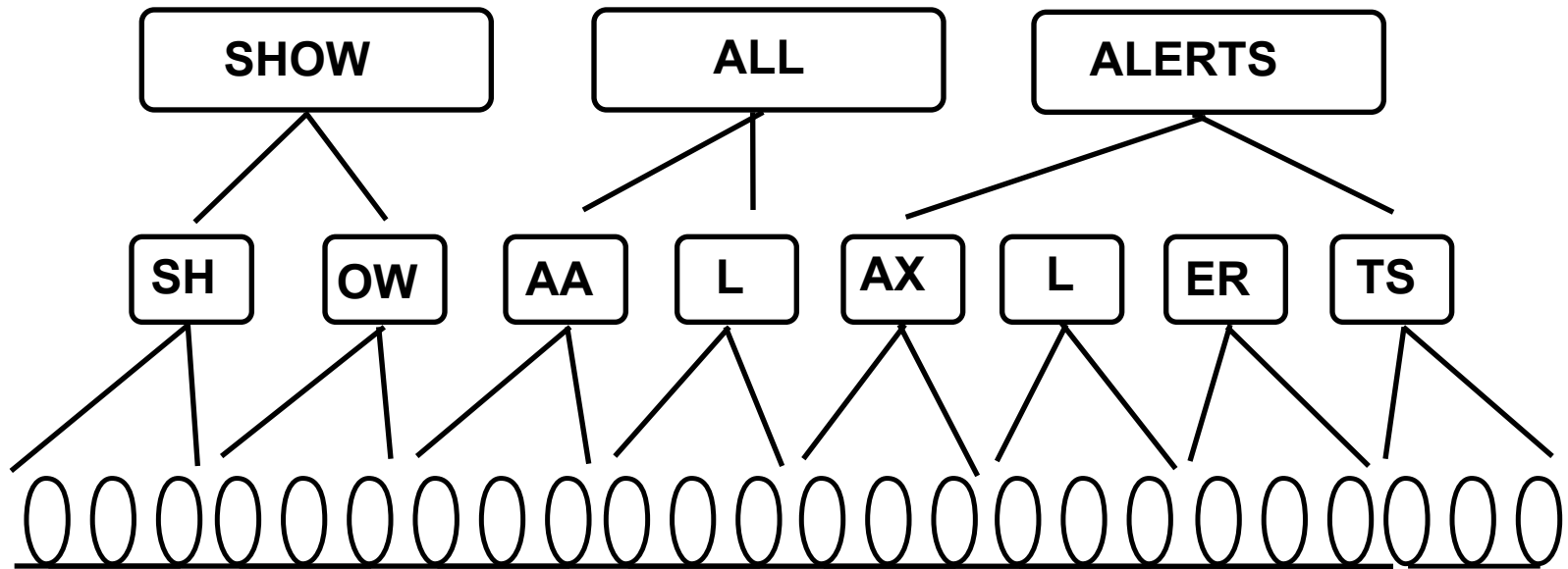
**Ergodic** - Fully connected, each state has transition to every other state



**Left-to-Right** - Transitions only to states with higher index than current state. Inherently impose temporal order. These most often used for speech.



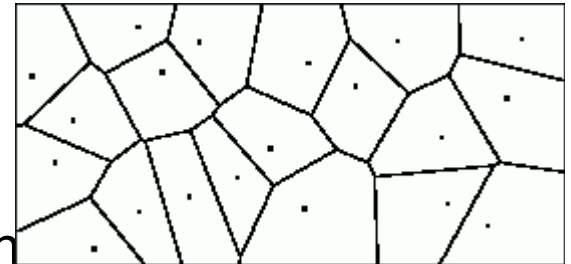
# Forward-Backward Training for Continuous Speech



# Discrete HMM's

## Vector Quantization

- Want to create a discrete set of areas that we can label
- choose a number of prototypical reference vectors
- the set of points to which a reference vector is the closest is called the vector's **Voronoi region**
- every Voronoi region is convex
- Give each Voronoi region a label, „code“
- Set of codes is the „code book“
- During Classification, assign label of the region new sample falls
- Sequence of labels represents „Observations“ of a Discrete HMM





# Acoustic Modeling

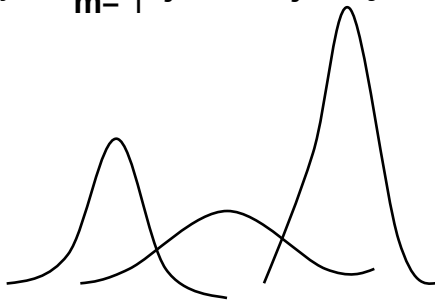
- How to Model Emission Probabilities
  - Discrete
  - Continuous
    - Mixture Gaussians
    - Neural Nets
  - Semi-Continuous, Tied Mixtures
- The Problem of Context
  - The Markov Assumption is really not Good!
  - Context-Dependent Phones
    - Tri-Phones
    - Poly-Phones

# Acoustic Modeling

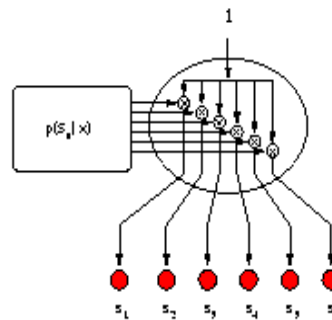
Emission Probabilities can be Estimated by Alternate Methods:

## Mixture of Gaussians Networks

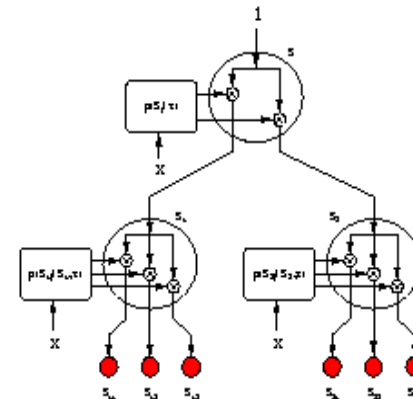
$$b_j(\mathbf{x}) = \sum_{m=1}^M c_m N[\mathbf{x}, \mu_{jm}, U_{jm}]$$



## Neural Networks

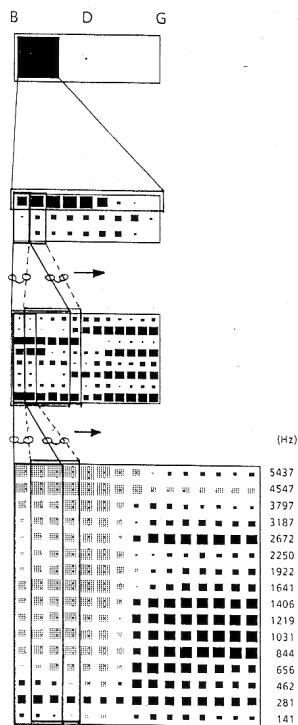


## Hierarchies of Neural



# Neural Nets.... (then & now: more, bigger, deeper)

(1987)



TDNN: Waibel '87

(1989)

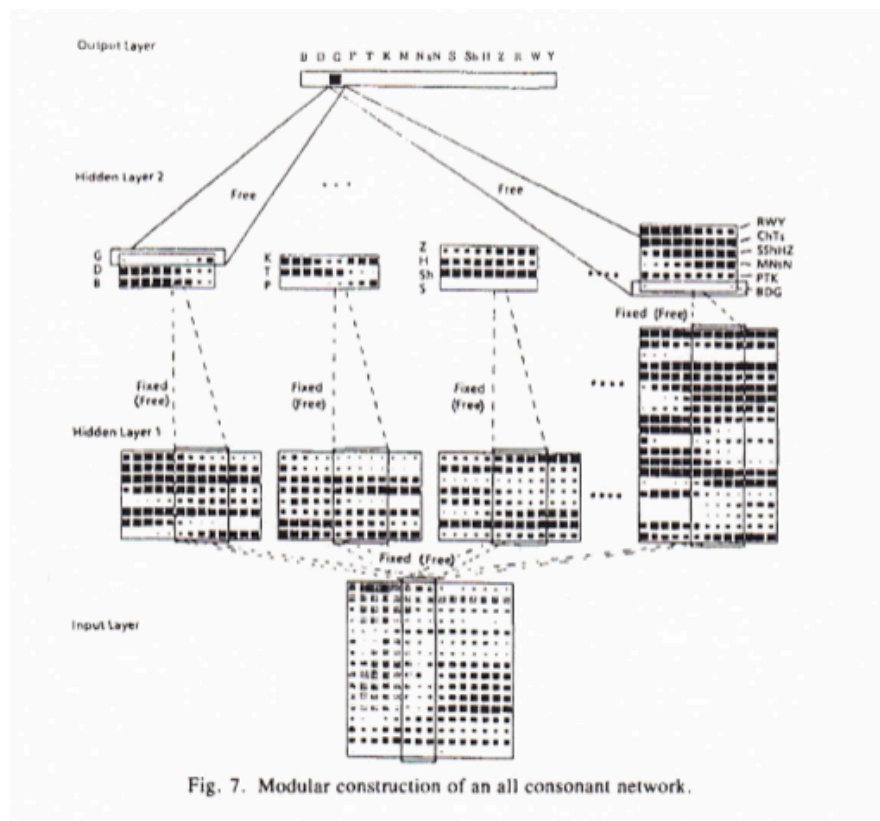
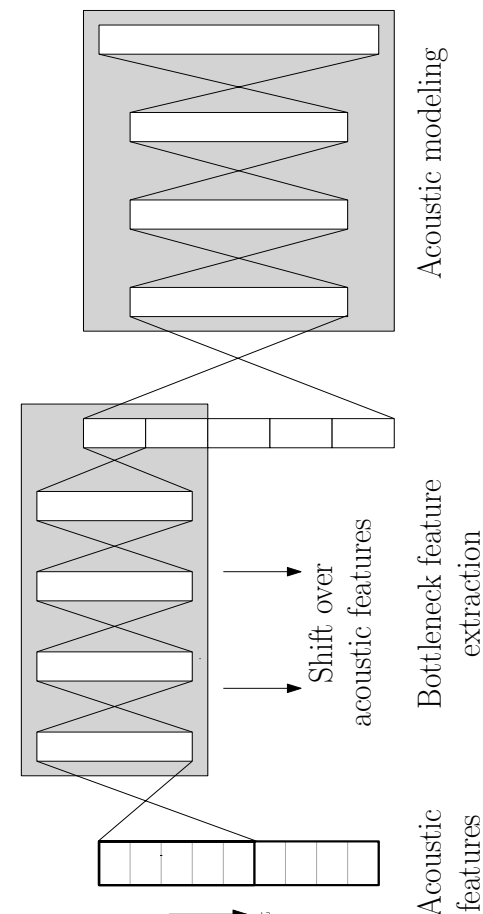


Fig. 7. Modular construction of an all consonant network.

Modular (deep) TDNN: Waibel '87

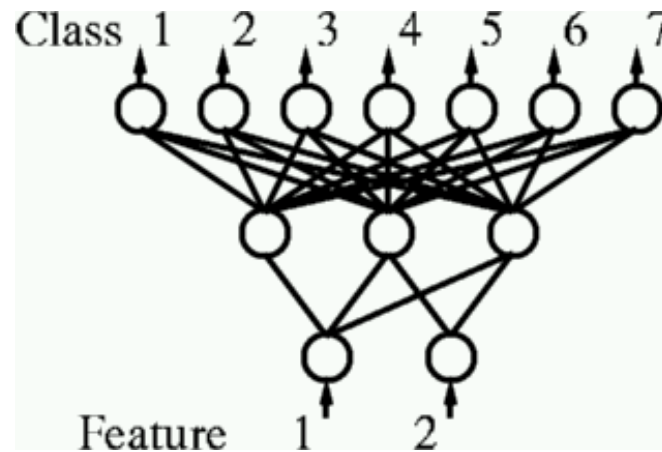
(2013)



Bigger, Deeper:  
Waibel et al. Babel, 2013

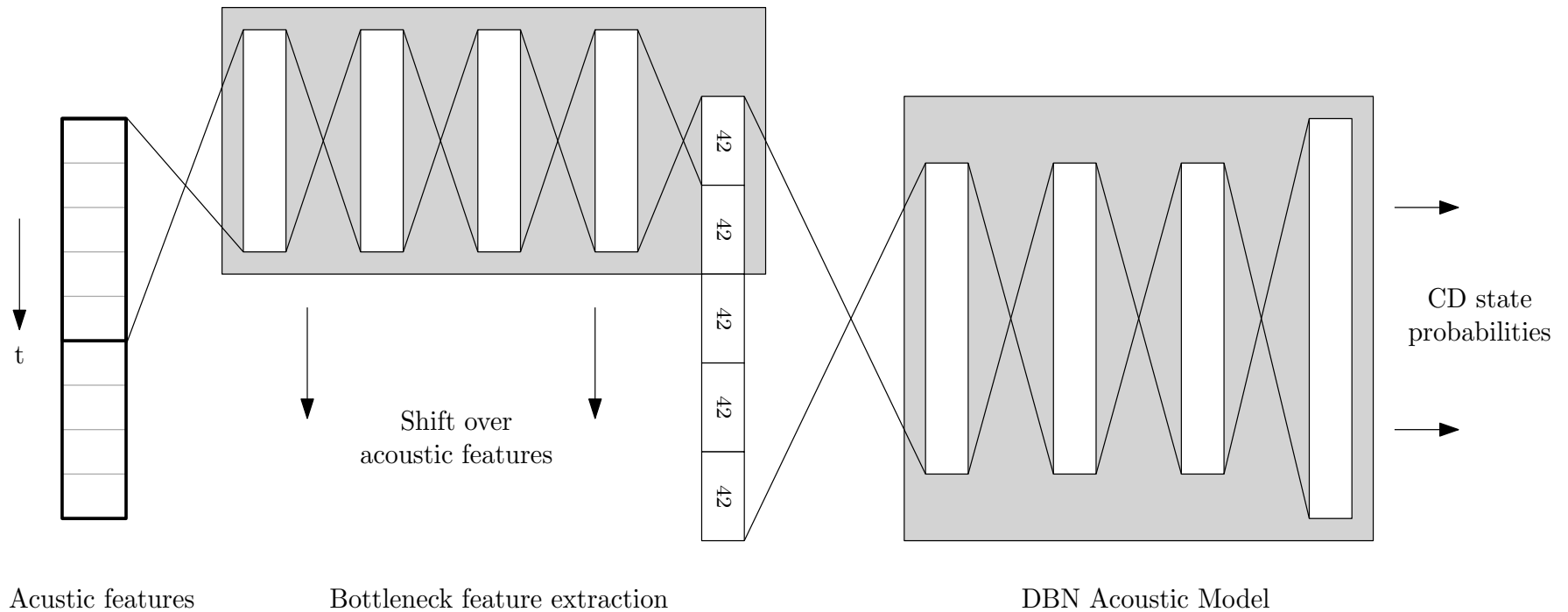
# Neural Net Approaches to Pattern Classification

most common  
approach:  
Multi-Layer  
Perceptron  
**MLP**



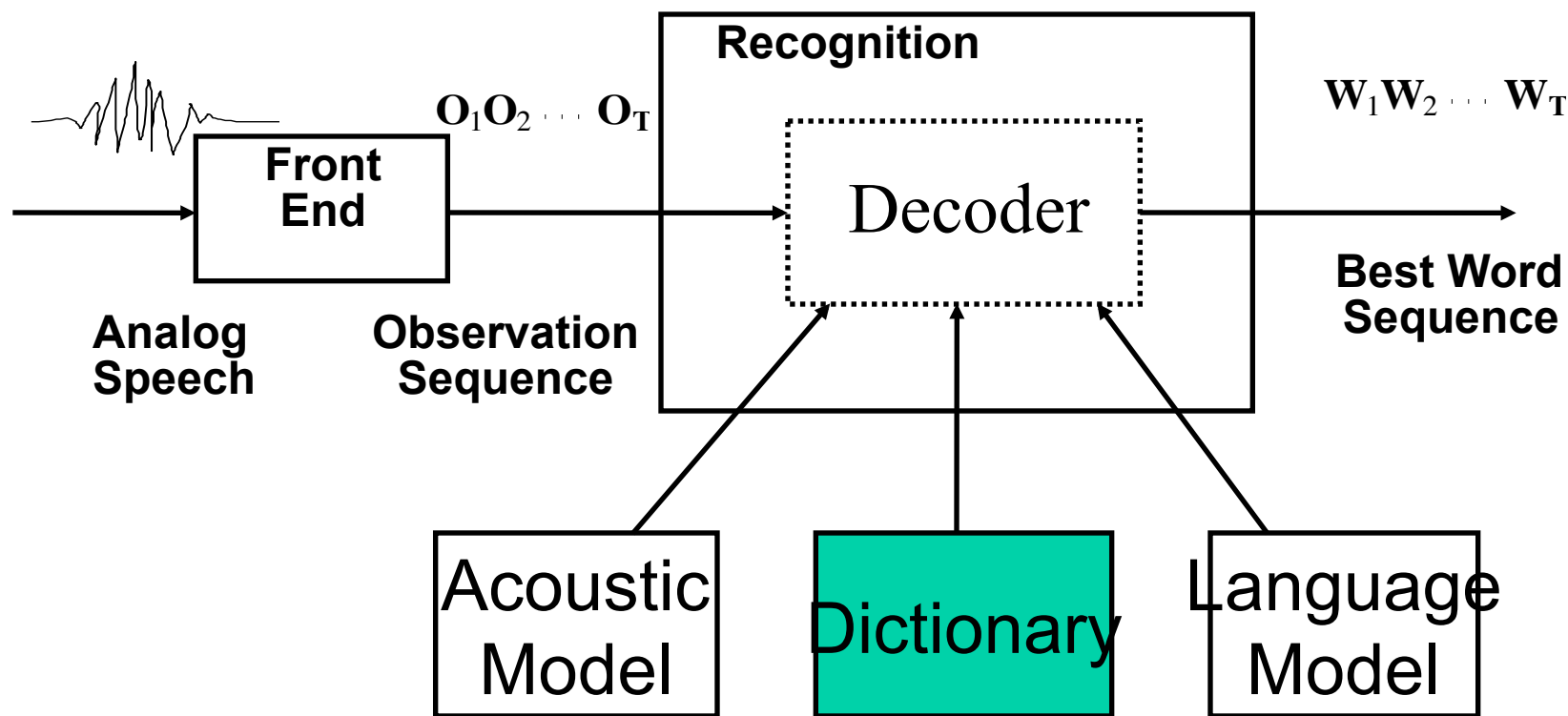
- we can prove that a MLP can approximate the probabilities  $P(\text{Class} | \text{Pattern})$
- most common training procedure: **error backpropagation**

# Recent Neural Acoustic Models



# Speech Recognition (System Components)

- Recognizer Components:

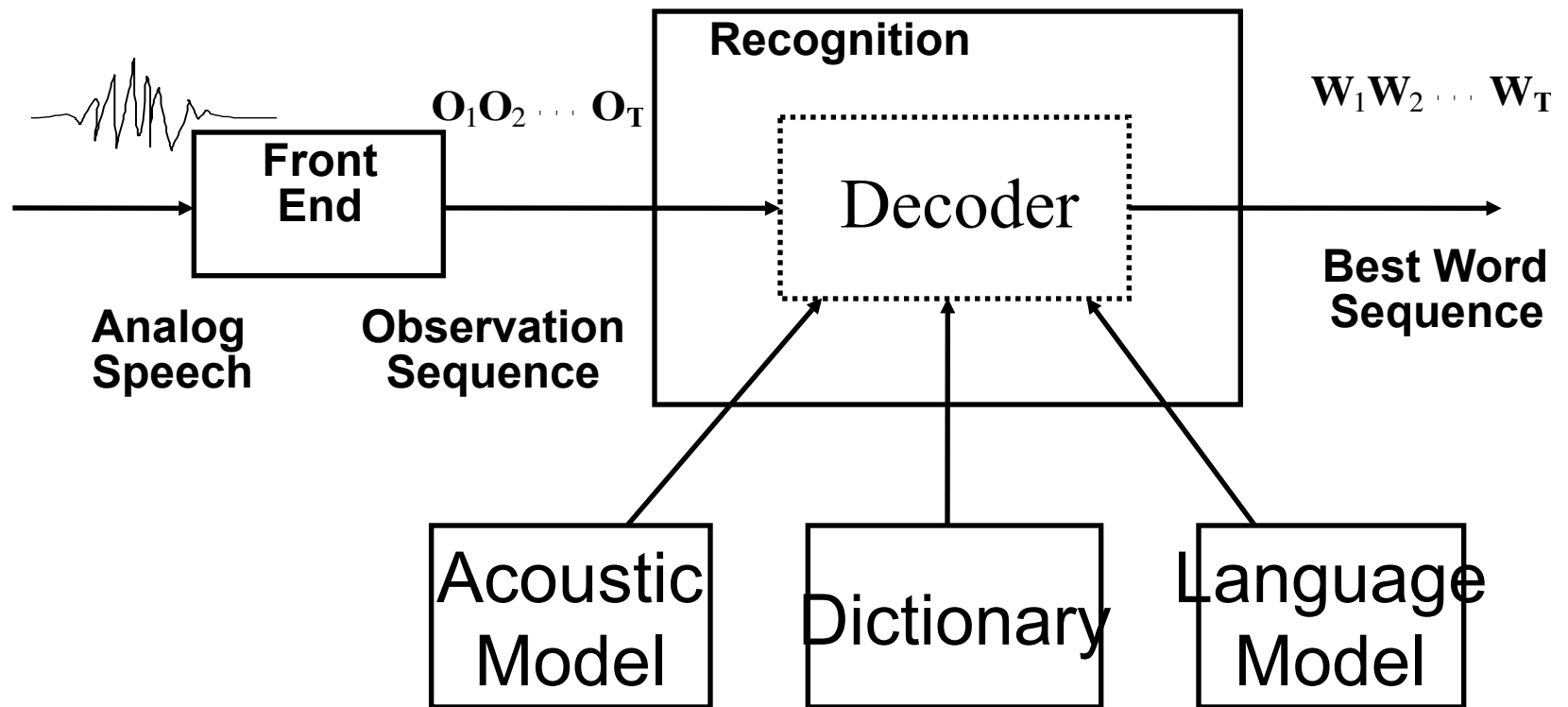


# Dictionaries

- Word Dictionaries
  - Words and Word Models
  - Assign Certain Number of States to Each Word Model
- Phonetic Dictionaries
  - Convert Orthography to Phoneme Strings
  - Represent Alternate Pronunciations  
“Zero”, “Oh”, “Because”, “Cause”
  - Multiwords: “Did You” → “Didjah”
- Tree-Structured Dictionary
  - Faster Search, Faster Overall Run-Time

# Speech Recognition (Components)

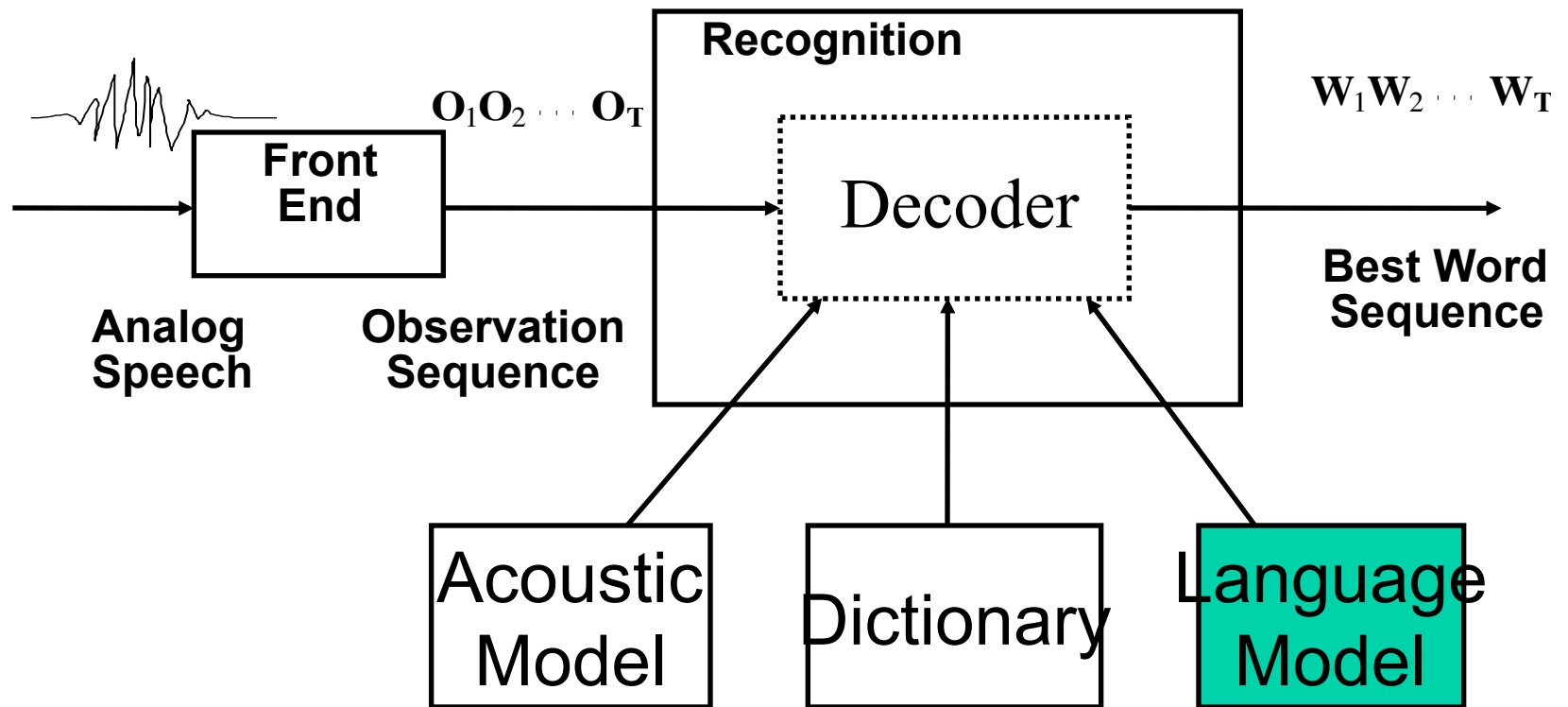
- Recognizer Components:





# Speech Recognition (System Components)

- Recognizer Components:



# Speech Recognition

- Goal:
  - Given acoustic data  $A = a_1, a_2, \dots, a_k$
  - Find word sequence  $W = w_1, w_2, \dots, w_n$
  - Such that  $P(W | A)$  is maximized

## Bayes Rule:

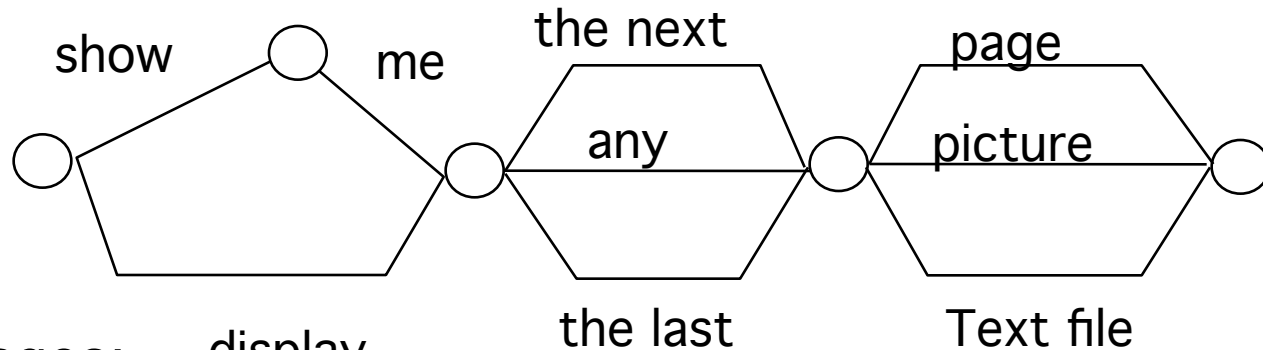
$$P(W | A) = \frac{P(A | W) \cdot P(W)}{P(A)}$$

acoustic model (HMMs)  $\swarrow$   $\nwarrow$  language model

**$P(A)$  is a constant for a complete sentence**

# Language Models: Grammar Based

- Write Grammar of Possible Sentence Patterns



- Advantages:
  - display
  - Long History / Context
  - Don't Need Large Text Database (Rapid Prototyping)
- Problem:
  - Work to Write Grammars
  - Rigid: Only Programmed Patterns can be Recognized

# Language Models: N-Grams

- Predict Next Word based on History
- History is Approximated by Past two or three (generally  $n$ ) past words
  - Everything before word  $w_{i-n}$  is placed into an equivalence class
- Then Probability of next word is given by
  - Trigram:  $P(w_i | w_{i-1}, w_{i-2})$
  - Bigrams:  $P(w_i | w_{i-1})$
  - Unigrams:  $P(w_i)$
- Advantage:
  - Trainable on Large Text Databases
  - Prediction 'Soft' (Probabilities)
  - Can be Directly Combined with Acoustic Model
- Problem:
  - Need Large Text Database for each Domain

# Objective Estimation of Language Model Quality

A language model is better than an alternative one, if the probability  $P(\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n)$  with which it would generate a test corpus  $W$  is larger.

But

$$P(\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n) = \prod_{i=1}^n Q(\mathbf{w}_i / \Psi(\mathbf{w}_1, \dots, \mathbf{w}_{i-1}))$$

so a good quality measure is the LOGPROB

$$\hat{H}(\mathbf{W}) = -\frac{1}{n} \sum_{i=1}^n \log_2 Q(\mathbf{w}_i / \Psi(\mathbf{w}_1, \dots, \mathbf{w}_{i-1}))$$

# Objective Estimation of Language Model Quality

If words were generated by the language "mechanism" uniformly at random from a vocabulary of size  $V$ , then

$$Q(\mathbf{w}_i / \Psi(\mathbf{w}_1, \dots, \mathbf{w}_{i-1})) = \frac{1}{V}$$

and

$$2^{\mathbb{H}(\mathbf{w})} = 2^{\log V} = V$$

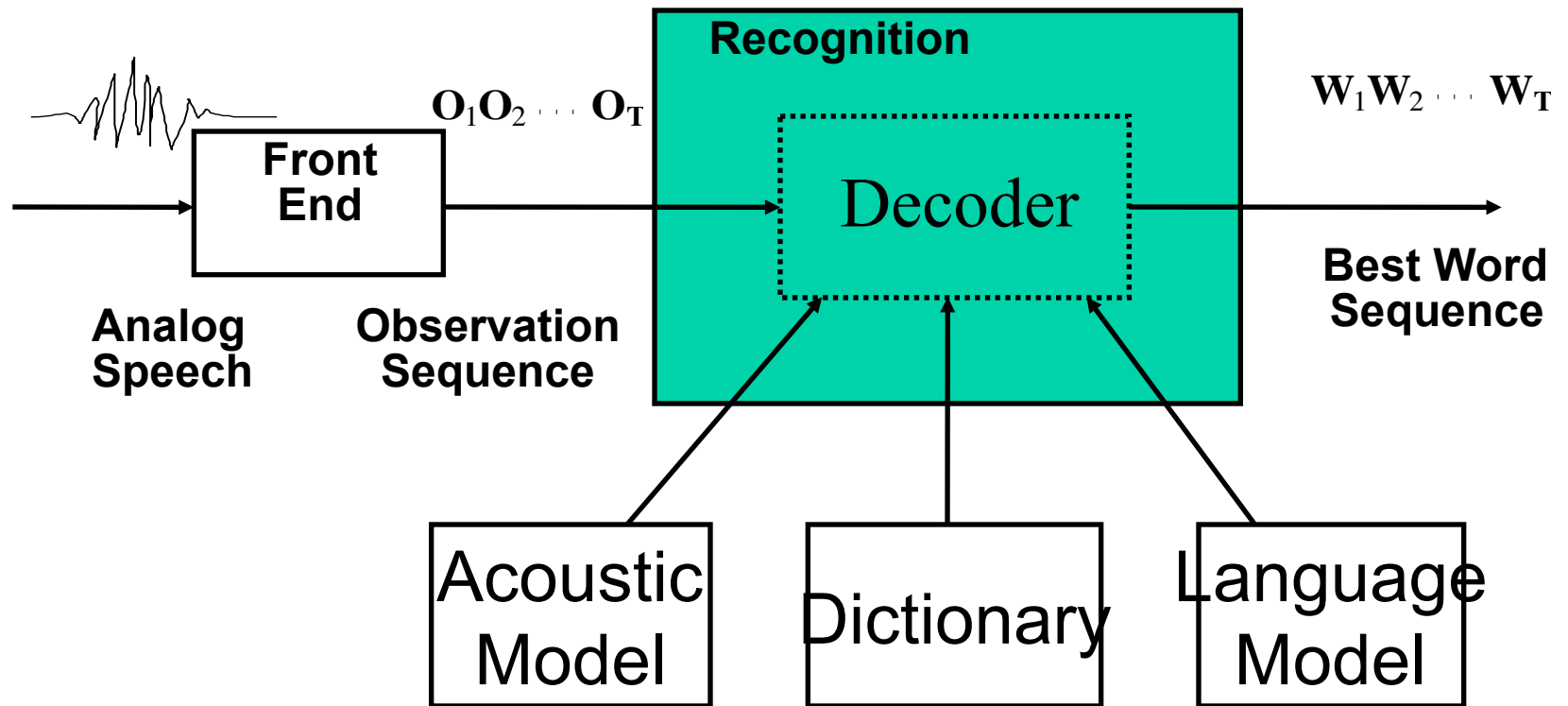
We can thus define the PERPLEXITY of the language model as:

$$PP(\mathbf{W}) = 2^{\mathbb{H}(\mathbf{w})}$$

and interpret it as the "branching factor" of the language, when  $\Psi$  is available.

# Speech Recognition (System Components)

- Recognizer Components:

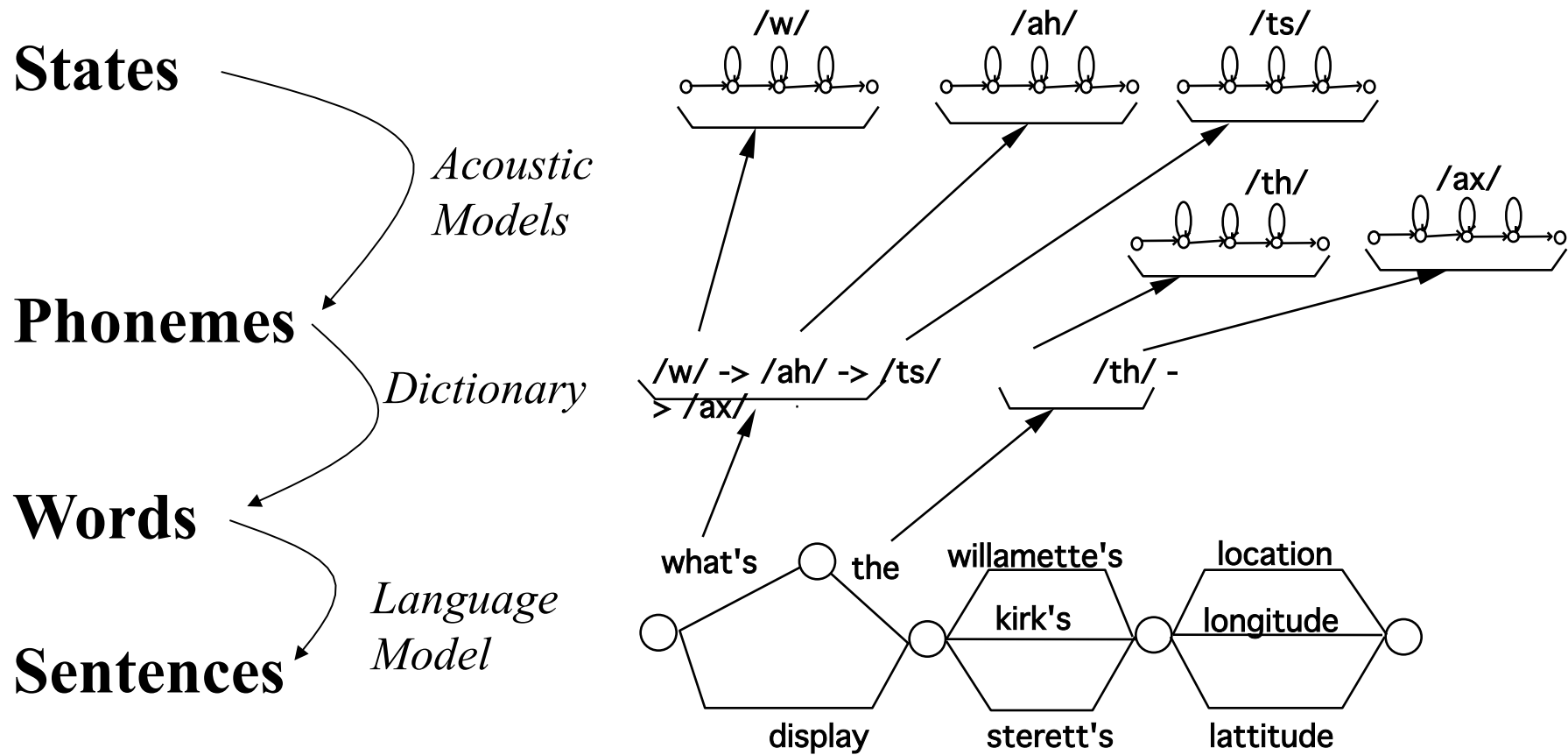


# Decoding

- Problem: Find most likely Word Sequence
- Acoustics Provide a Score for each Word Hypothesis
- Language Model Provides Constraints
  - Grammar
  - N-Gram
- Search finds most likely Word Sequence Using **Both**
  - Acoustic Scores and
  - Language Model Constraints

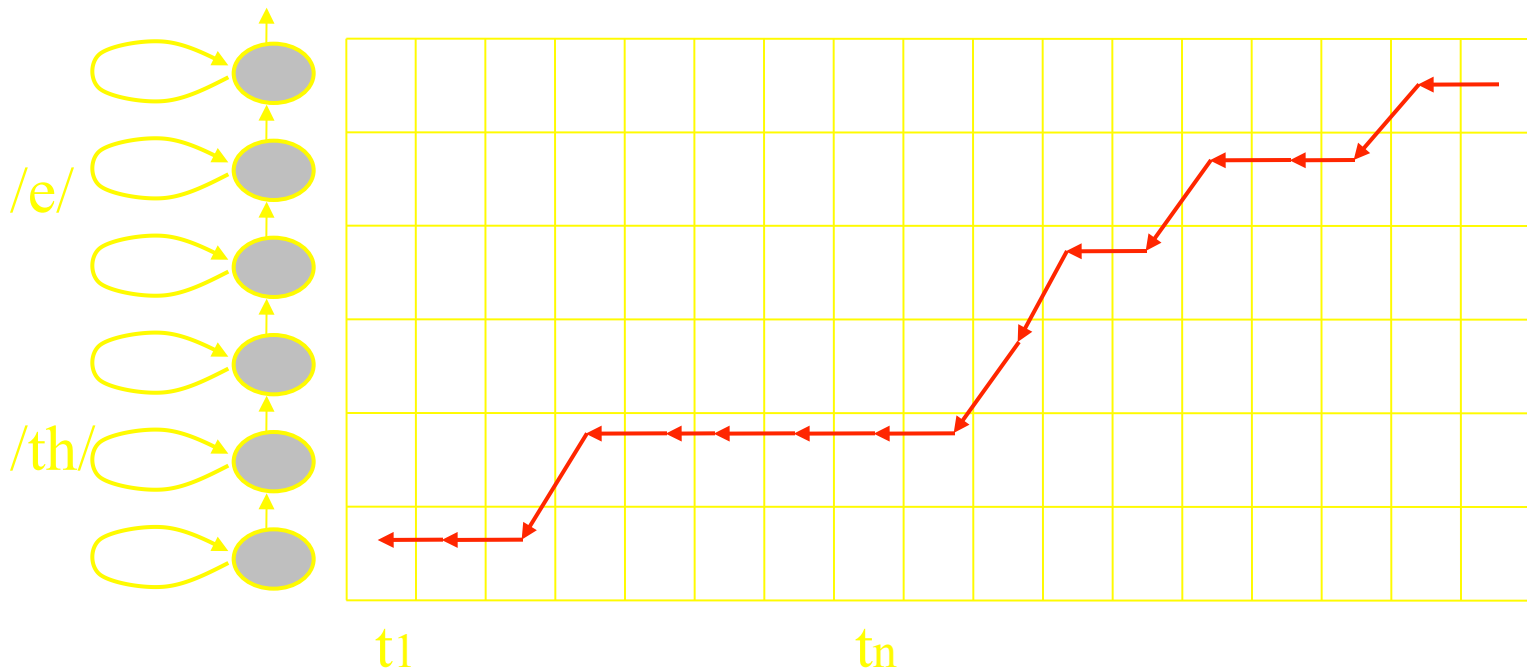


# Decoder - Assembling the Pieces



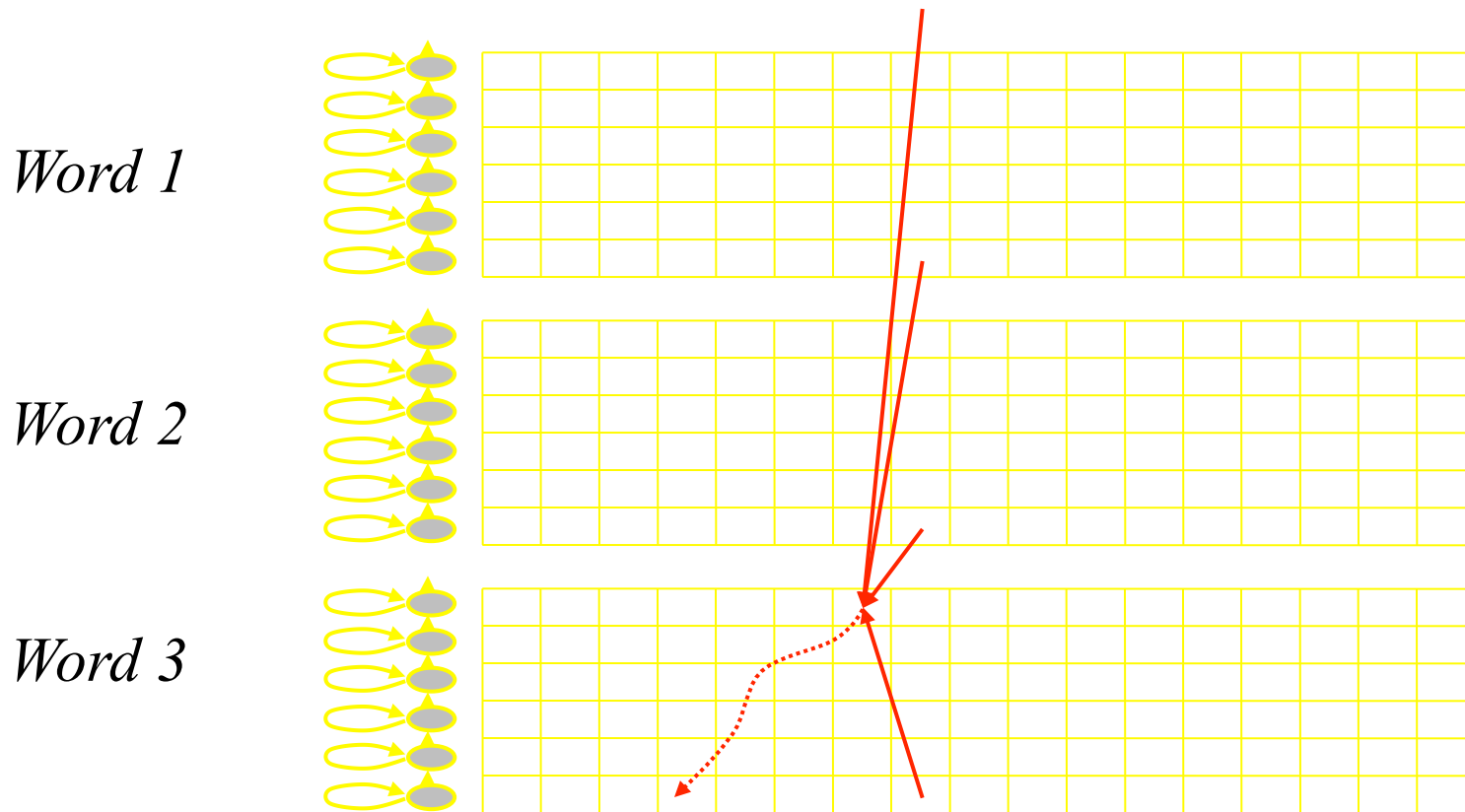
# Search

- Find Best Association between Word and Signal
- Efficiently Compose Words from Phones Using Dictionary

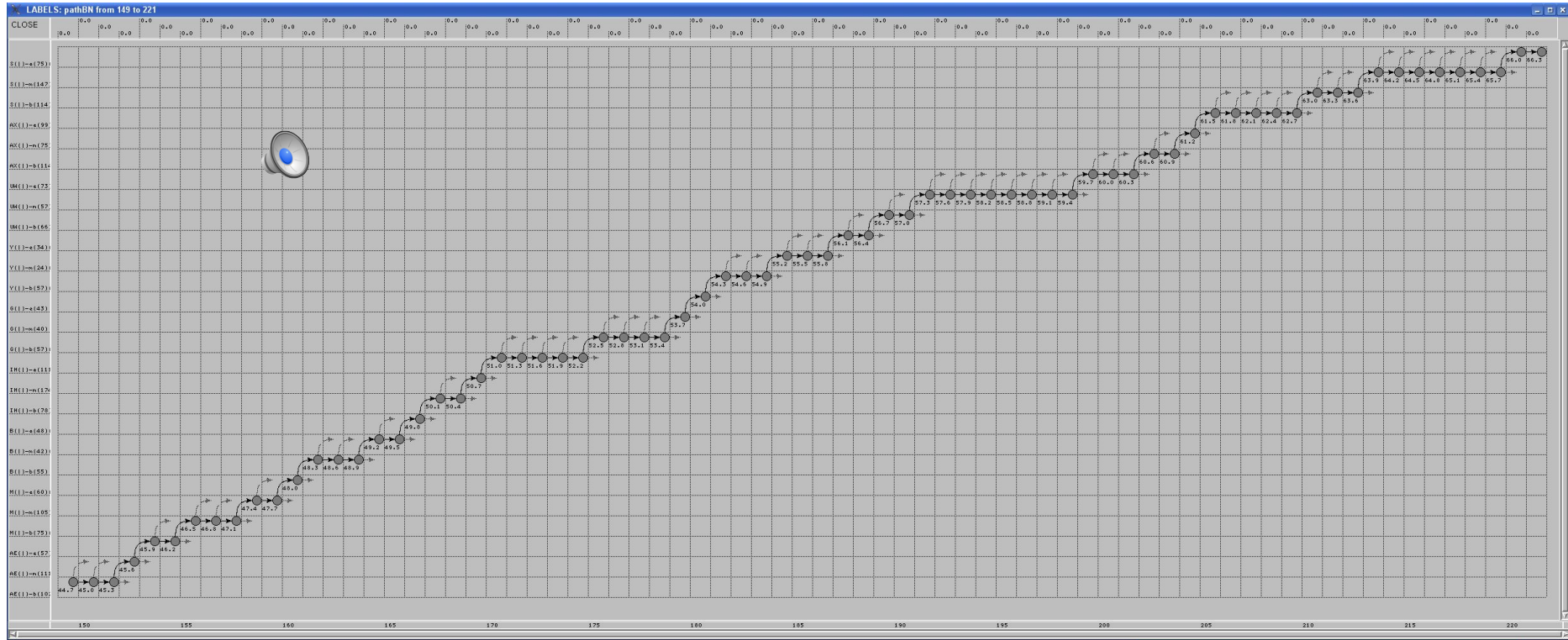


# Search - continuous speech

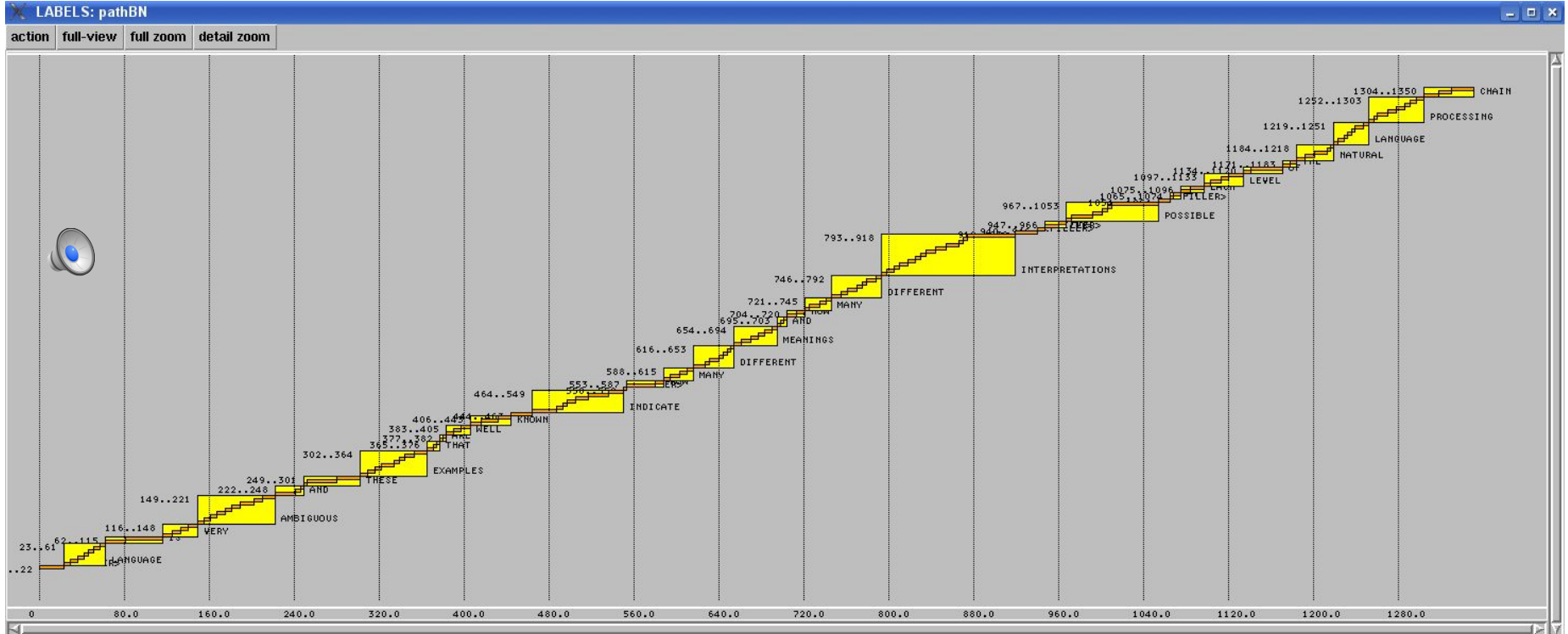
- Compose Sentences from Words Using Language Model



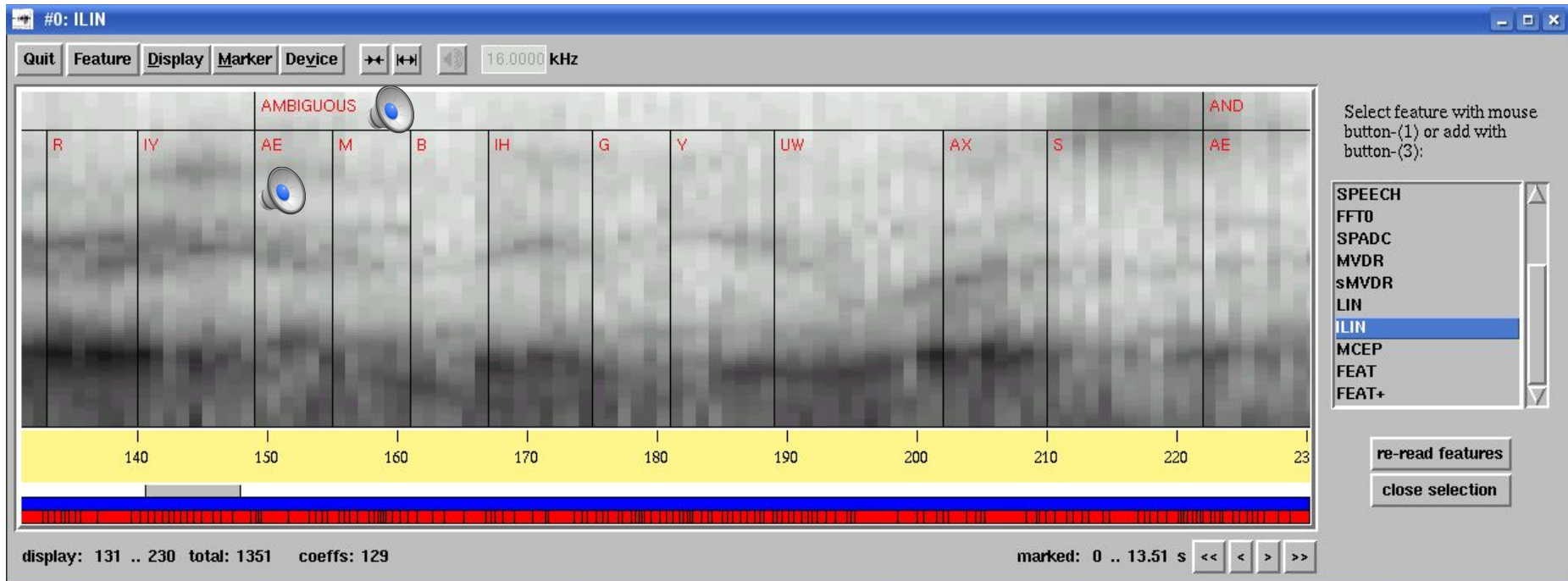
# Viterbi Alignment



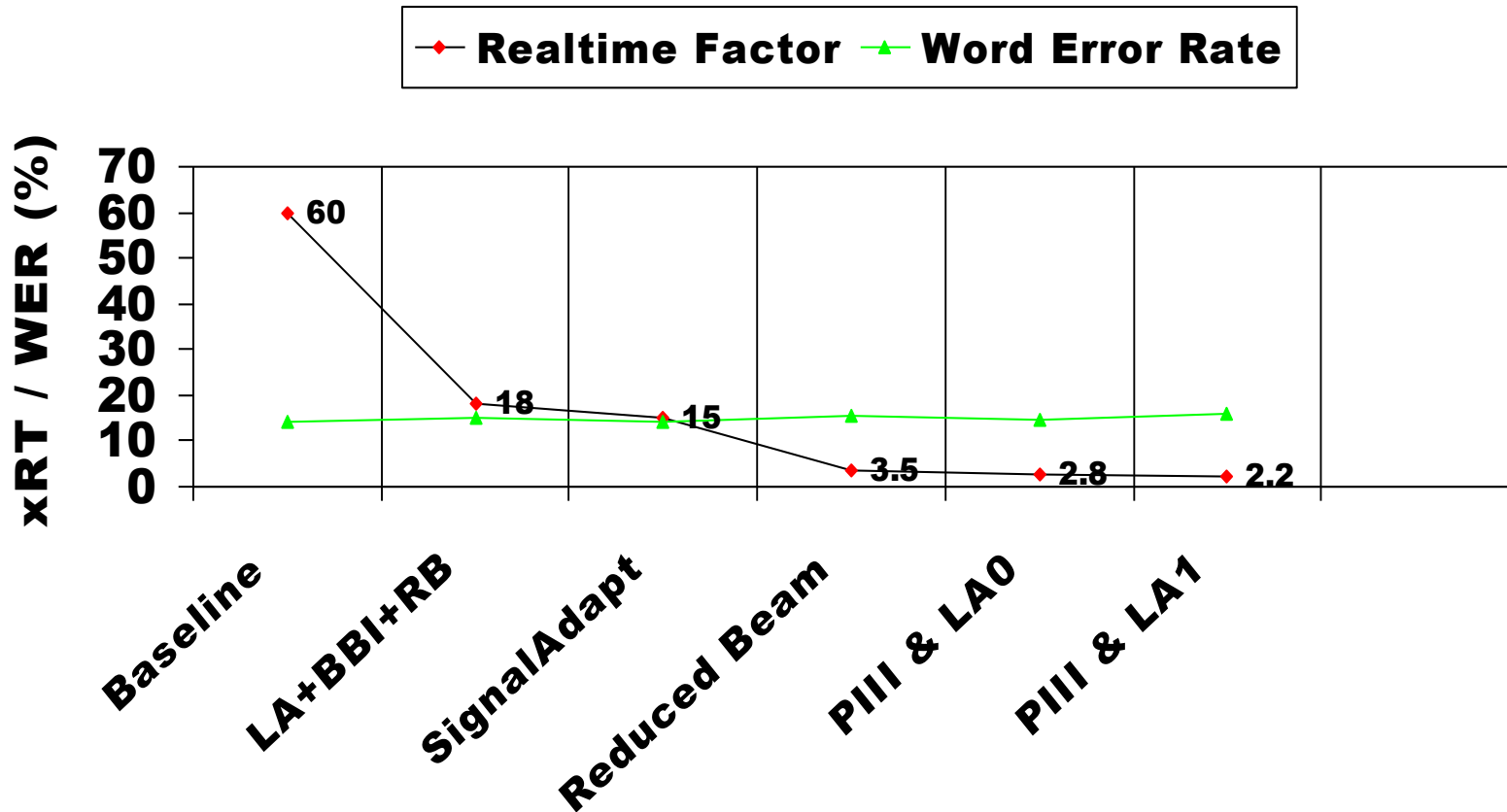
# Viterbi Alignment



# Viterbi Alignment



# Improving Speed on Cooperative Speech



# Measuring Recognizer Performance

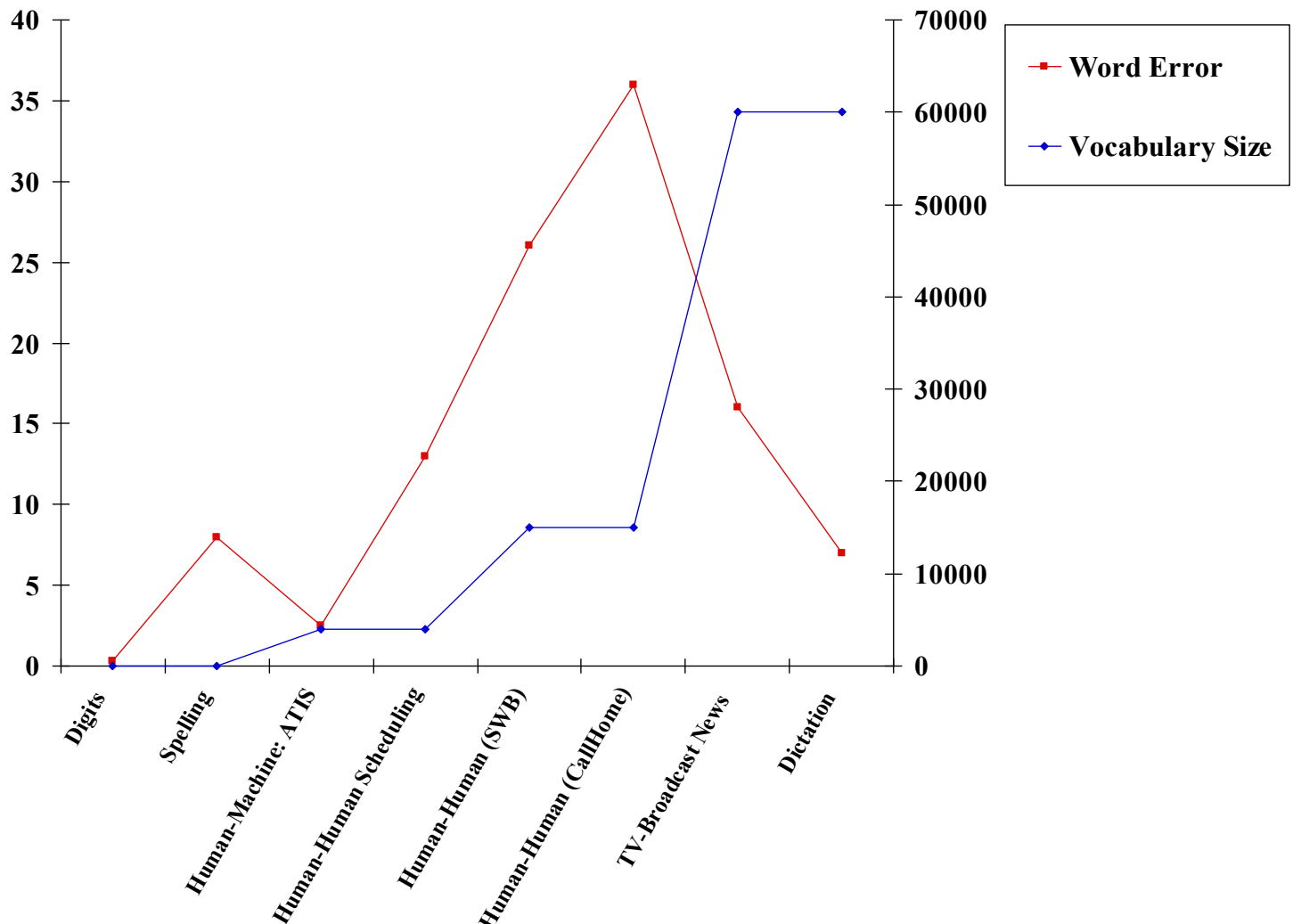
- Word Error Rate:
  - Insertion, Deletion and Substitutions
  - $WER = 1/N * (\#Ins + \#Del + \#Sub)$
  - Determine I,D,S, by performing Alignment Search between output Hypothesis and Reference
- Perplexity
- Other Factors...



# Factors Affecting Recognizer Performance

- Speaker: Dependent / Independent
- Speaking Style: Isolated / Continuous / Spontaneous / Conversational
  - Sloppy, Coarticulated, Reduced
- Size: Small / Medium / Large Vocabulary
- Confusability: Small (digits), Large (Spelling)
- Sound Quality
  - Noise: Office, Car, Telephone, .....
  - Microphone: Close Speaking, Lapel, Table Top, Room, ...

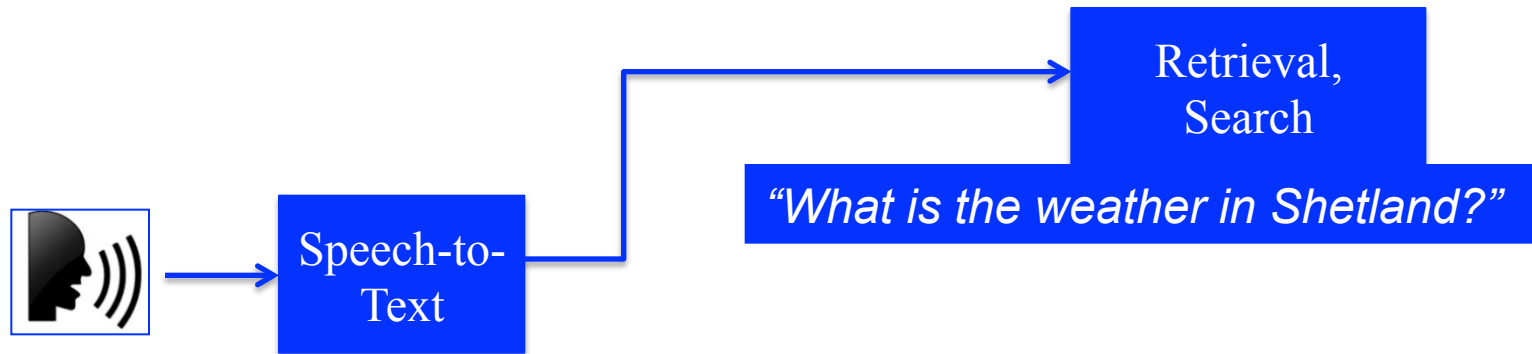
# Error Rates in Perspective: The State of the Art



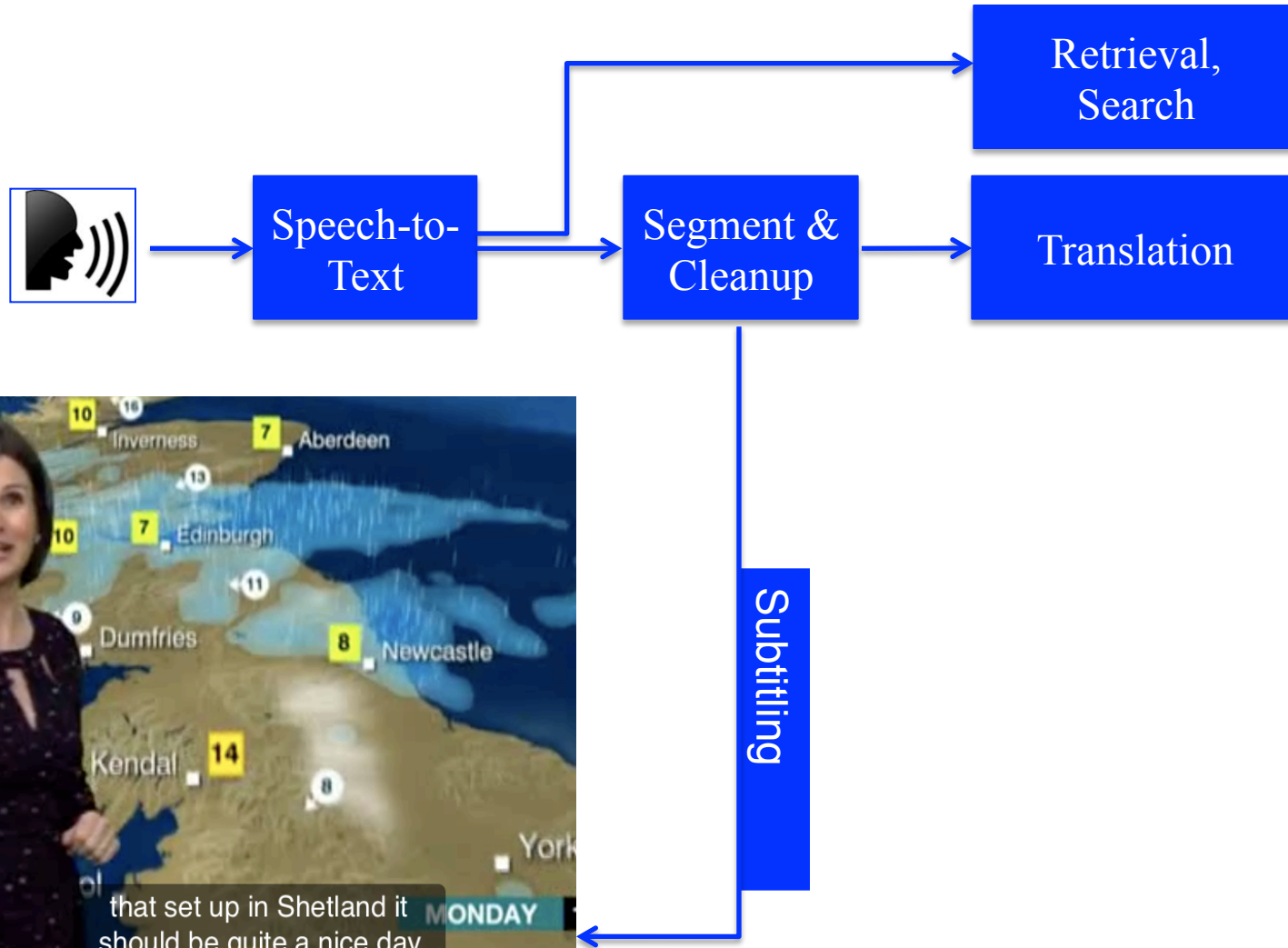
# How Good Does it Have to be?

<u>Use Case</u>	<u>Needed WER</u>	<u>Task</u>	<u>Current WER</u>
Retrieval	30-50%	TV Announcer	3-5%
KeyWord Spotting	30-50%	BroadCast News	5-20%
Analytics	50-80%	Lectures	8-15%
Subtitles (Gisting)	20%	Spontaneous Queries	10-20%
Translation (Comprehension)	<15%	Telephone Conversations	30-50%
TV-Subtitles (Production)	5%	Meetings	40-60%

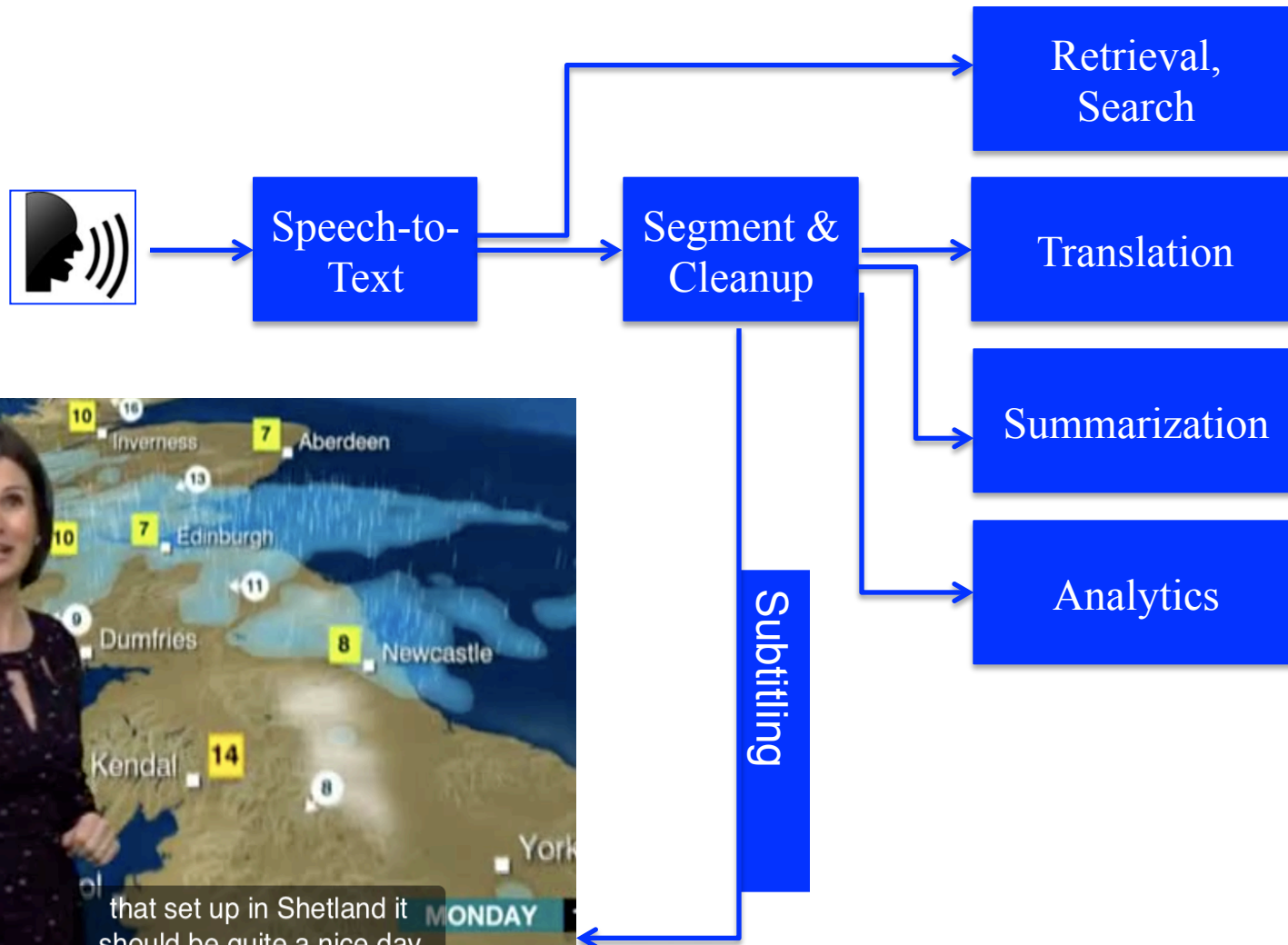
# Speech Deployment



# Speech Deployment



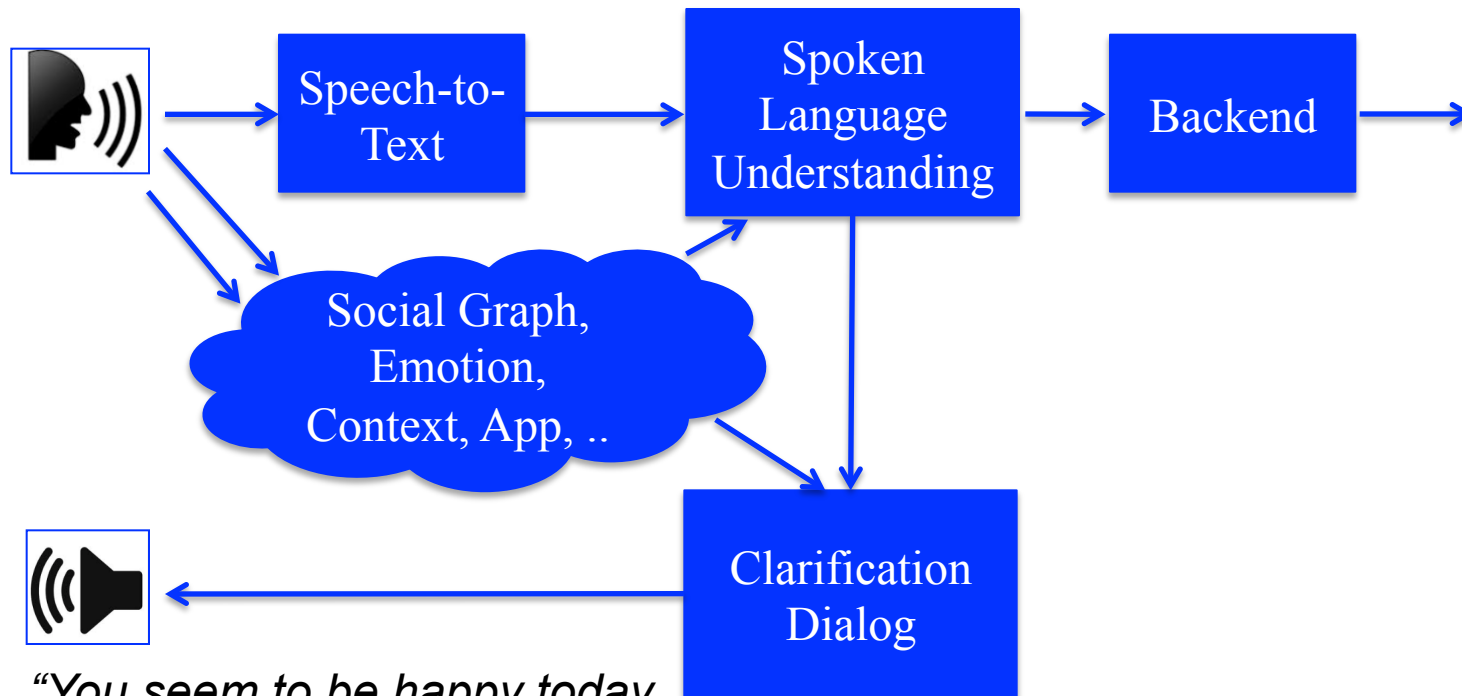
# Speech Deployment



# Voice Agents

*“Er.. Hum.. I’d like to.. Er. I want to ..hum..  
want to post this picture.....!”*

<power-on;  
Action: Post;  
Audience: Friends>



*“You seem to be happy today..  
May I assume you want to post  
these pictures to your family?”*

# Natural Language Processing & Machine Translation



# Natural Language Processing

- Natural Language – Processing natural language as used by human beings, in contrast to formal languages
  - Language is ambiguous
  - Language is dynamic and changes over time
- Goal:
  - How to Build Systems that can interact in natural language with human user
  - How to Build Systems that can *Understand* human language

# Natural Language Processing

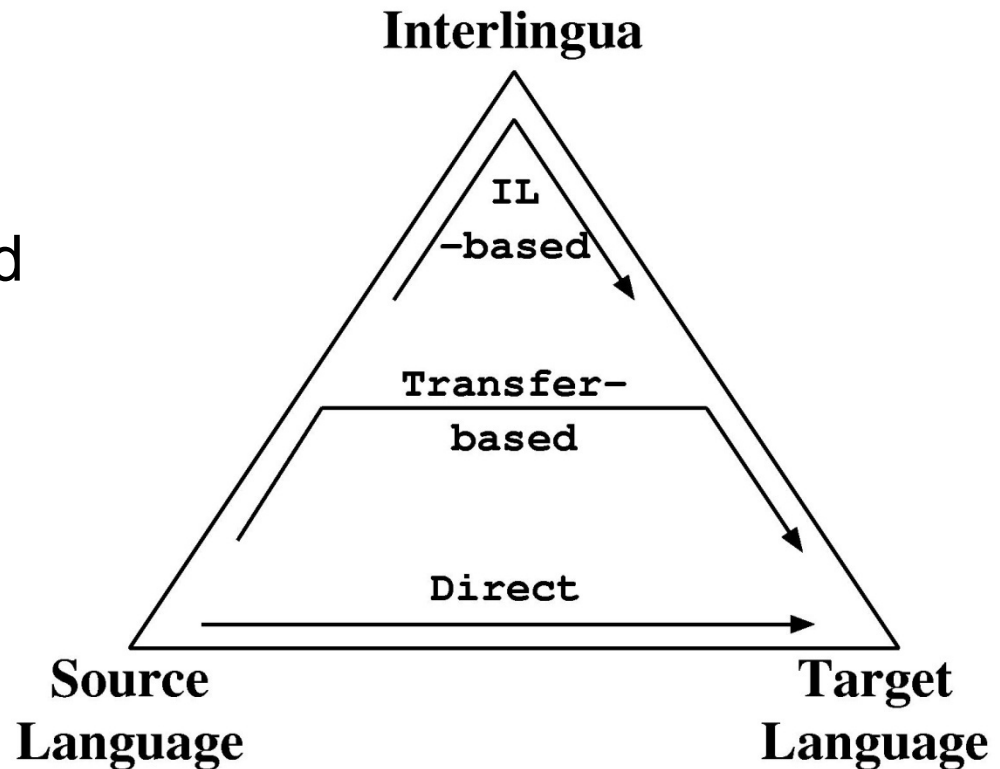
- Areas of Concern:
  - Question Answering
  - Information Retrieval, Information Access
  - Interactive Assistance
  - Natural Command Interpreters
  - Error Checking in Text Processing
  - Speech Recognition
  - Text Generation
  - Dialog Processing
  - Discourse Processing
  - Summarization, Gisting
  - Named Entity Detection
  - Machine Translation

# Natural Language Processing

- Scientific Tools/Methods:
  - Computational Analysis of
    - Grammar: Parsing
    - Morphology: Morph Decomposition
    - Part of Speech: POS tracking
  - Resources:
    - Grammars
    - Ontologies
    - Tree-Banks
  - Approach:
    - Handwritten Grammars (need programming)
    - Statistical Analysis (need data)
    - Trend toward statistics.. Data is cheaper...

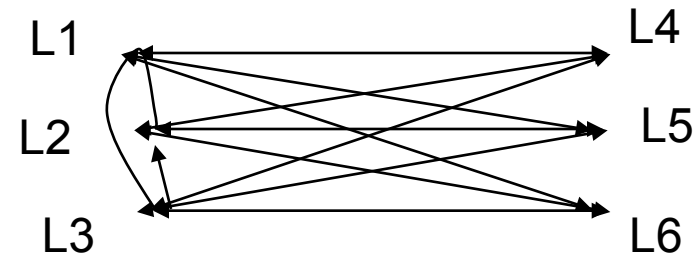
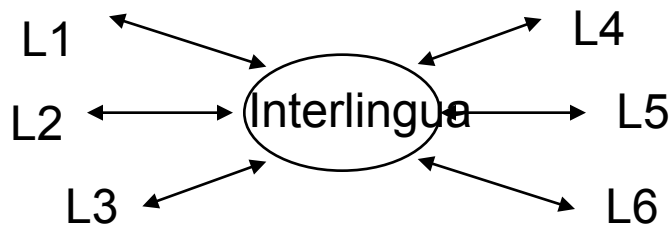
# Machine Translation: Approaches

- Interlingua based
- Transfer based
- Direct
  - Example based
  - Statistical



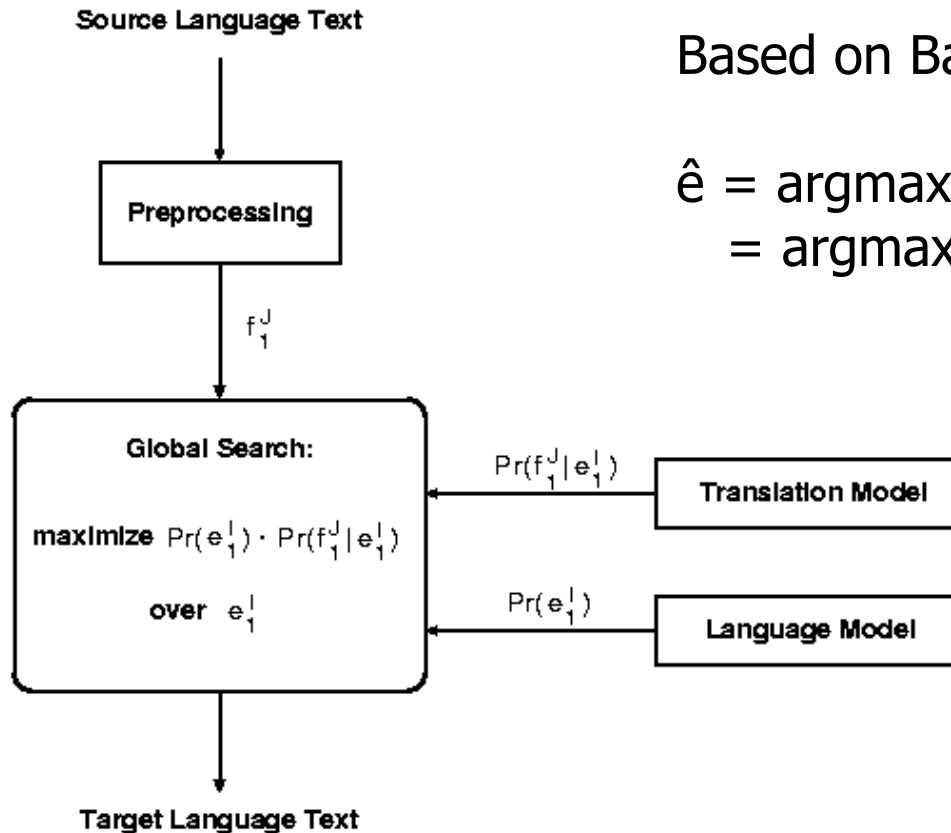
# MT - Interlingua Approach

- Need only N parser/generators instead of  $N^2$



- Rapid Addition of New Output Language
- Can generate culturally / contextually appropriate interpretation
- Eliminate Disfluencies, Clean-Up Language
- Generate Paraphrase in Own Language for Verification

# Statistical Machine Translation



Based on Bayes' Decision Rule:

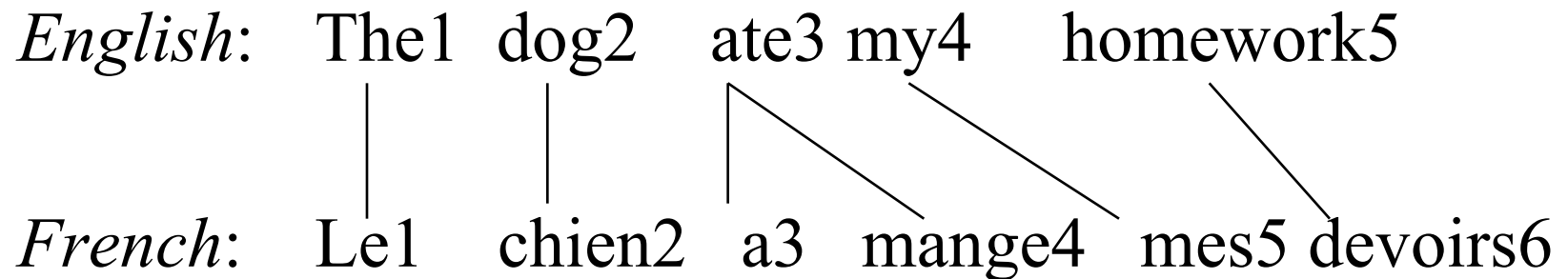
$$\begin{aligned}\hat{e} &= \operatorname{argmax}\{ p(e | f) \} \\ &= \operatorname{argmax}\{ p(e) p(f | e) \}\end{aligned}$$

# Alignment

- **Alignment:** relation between individual words of sentences in source and target language

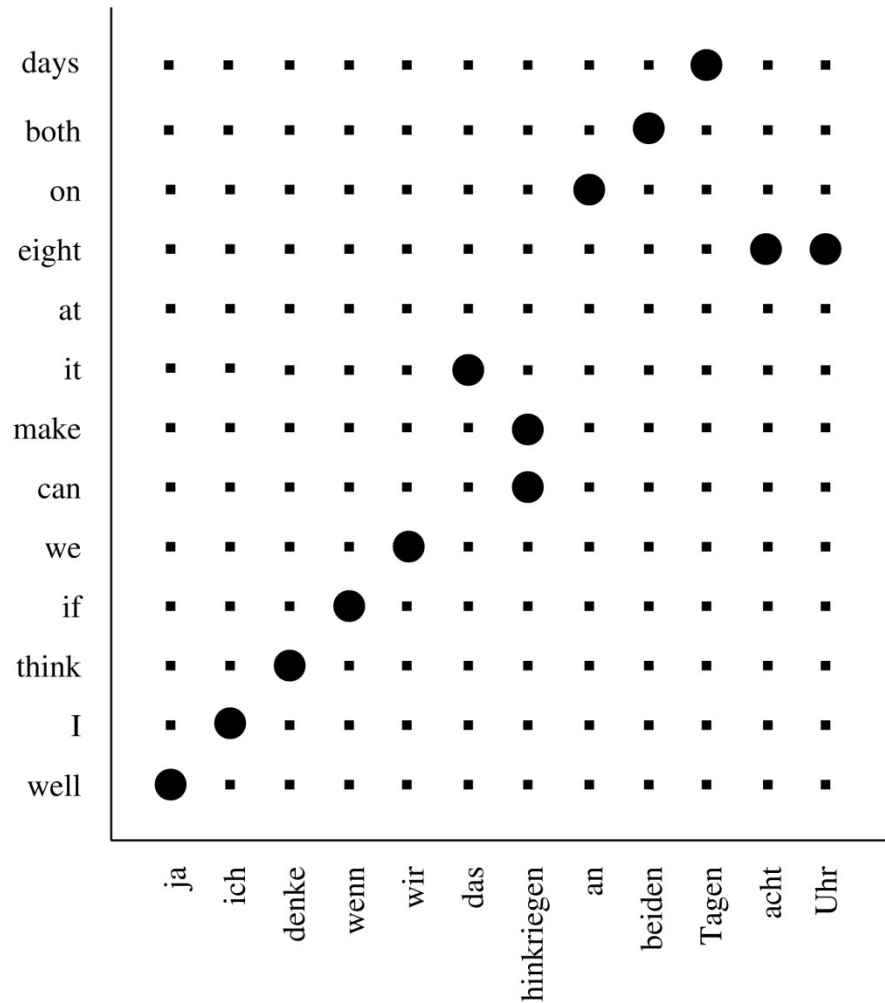
- **Example:**

*English:* The<sup>1</sup> dog<sup>2</sup> ate<sup>3</sup> my<sup>4</sup> homework<sup>5</sup>  
*French:* Le<sup>1</sup> chien<sup>2</sup> a<sup>3</sup> mange<sup>4</sup> mes<sup>5</sup> devoirs<sup>6</sup>



- **Formal definition** for alignment  $a$ : set of pairs  $\langle i, k \rangle$   
= connection between  $k$ -th word in target language  
and  $i$ -th word in source language

# Alignment Example





# Tasks in SMT

- Modeling

Build statistical models which capture characteristic features of translation equivalences and of the target language

- Word alignment models to extract word translations
- Different methods to extract phrase translations

- Training

Train translation model on bilingual corpus, train language model on monolingual corpus

- Training corpus size matters:  $\geq 10$  million words desired
- But SMT works also on very small corpora

- Decoding

Find best translation for new sentences according to models





# Language Modeling in Automatic Speech Recognition

We are concerned about the rules for legal / likely word sequences.  
Why?

- **improve speech recognizer**  
add another information source

- **disambiguate homophones**  
find out that "I OWE YOU TOO" is more likely than "EYE O U TWO"

- **search space reduction**  
when vocabulary is  $n$  words, don't consider all  $nk$  possible  $k$ -word sequences

- **analysis**  
analyse utterance to *understand* what has been said

# Language Modeling in Automatic Speech Recognition

Remember the fundamental problem of speech recognition:

**Given:** An Observation  $X = x_1, x_2, \dots, x_T$  **Wanted:**  $W' = w'_1, w'_2, \dots, w'_n$   
with highest likelihood:

$$W' = \operatorname{argmax}_W P(W | X) = \frac{p(X | W) \cdot P(W)}{p(X)} = \operatorname{argmax}_W p(X | W) \cdot P(W)$$

This poses four problems to the speech recognizer:

- What is  $X$ ? The problem of preprocessing.
- What is  $p(X | W)$ . The acoustic modeling.
- What is  $P(W)$ . The language modeling.
- How do we find the  $\operatorname{argmax}_W$ ? The search problem.

# Deterministic vs. Stochastic Language Models

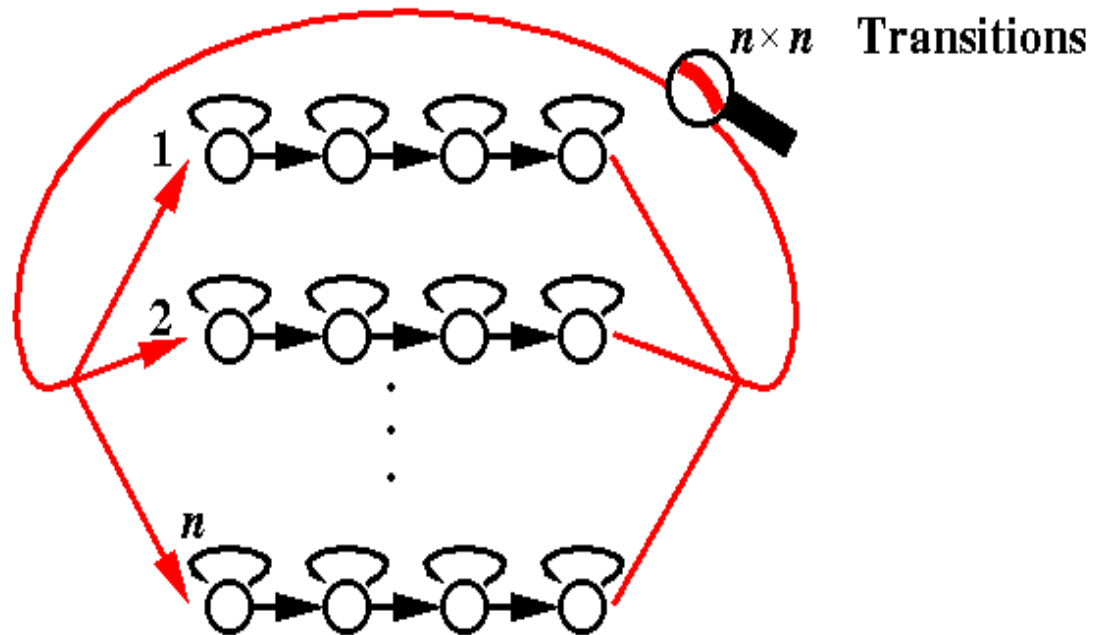
In HMM-recognizers, the language model is responsible for the computation of the word-to-word transition probabilities.

These can be computed on the fly, and may depend on more than just the previous word.

LMs can be

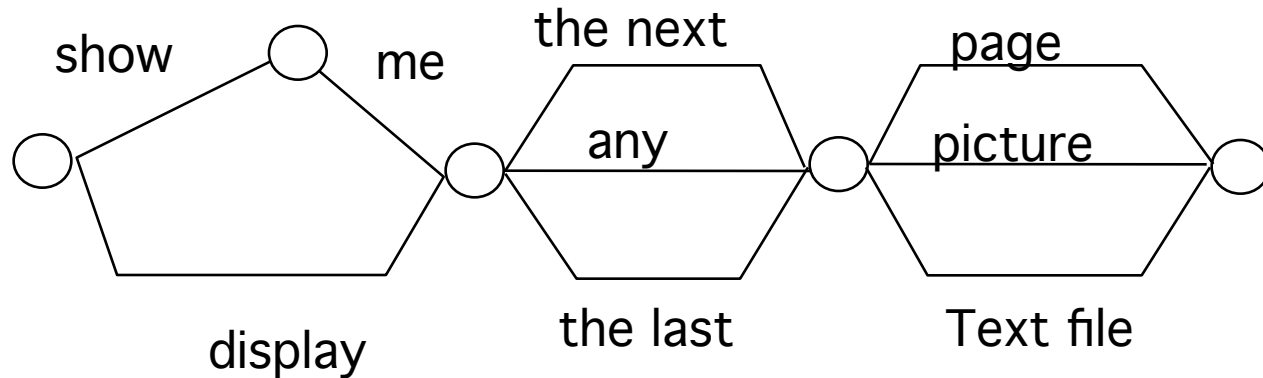
**deterministic:**  $P(w_j|w_i) = 0.0$  or  $1.0/n$  (e.g. finite state grammars)

**stochastic:** transition probabilities are in the range 0.0 to 1.0



# Language Models: Grammar Based

- Write Grammar of Possible Sentence Patterns



- Advantages:
  - Long History / Context
  - Don't Need Large Text Database (Rapid Prototyping)
- Problem:
  - Work to Write Grammars
  - Rigid: Only Programmed Patterns can be Recognized

# Probabilities of Word Sequences

The language model computes:

$$P(W) = P(w_1 w_2 \dots w_n) = P(w_1) \cdot P(w_2 | w_1) \cdot P(w_3 | w_1 w_2) \cdot \dots \cdot P(w_n | w_1 w_2 \dots w_{n-1})$$

Such language models usually only consider the past (the history) to predict a word. In principle it is also possible to "predict" a word from the future part of a hypothesis (not suitable for runon recognition).

Above, we have omitted the prior probabilities for the length of the word sequence:  $P(|W|=n)$ .

Alternatively, we can introduce a special symbol \$ that indicates the end of a word sequence

(for all  $i > n$ :  $w_i = \$$ ). Then we can, in theory, write

$$P(w | \text{history}) = \begin{cases} f(\text{history } w) / f(\text{history}) & \text{if } w \text{ is not } \$ \text{ and } \$ \text{ not in history} \\ f(\text{history}) & \text{if } w = \$ \text{ and } \$ \text{ not in history} \\ 0 & \text{if } w \text{ is not } \$ \text{ and } \$ \text{ in history} \\ 1 & \text{if } w = \$ \text{ and } \$ \text{ in history} \end{cases}$$



# Probabilities of Word Sequences

When computing  $P(w \mid \text{history})$ , we easily see:

For a vocabulary of 64,000 words and average sentence lengths of 25 words (typical for Wall Street Journal), we end up with a huge number of possible histories ( $64,000^{25} > 10^{120}$ ).

So it is impossible to precompute a special  $P(w \mid \text{history})$  for every history.

Two possible solutions:

- compute  $P(w \mid \text{history})$  "on the fly" (rarely used, very expensive)
- replace the history by one out of a limited feasible number of classes  
such that  $P'(w \mid \text{history}) = P(w \mid \mathbf{C}(\text{history}))$

# Classification of Word Sequence Histories

We can use different equivalence classes using information about:

- grammatical content (phrases like noun-phrase, etc.)
- POS = part of speech of previous word(s) (e.g. subject, participle, etc.)
- semantic meaning of previous word(s)
- context similarity (word(sequence)s that are often observed in similar contexts are treated equally, e.g. weekdays, people's names etc.)
- apply some kind of automatic clustering (agglomerative or divisive)
- $n$ -grams:  $P'(w_k \mid w_1 w_2 \dots w_{k-1}) = P(w_k \mid w_{k-(n-1)} w_{k-n-2} \dots w_{k-1})$
- bigrams:  $P'(w_k \mid w_1 w_2 \dots w_{k-1}) = P(w_k \mid w_{k-1})$
- trigrams:  $P'(w_k \mid w_1 w_2 \dots w_{k-1}) = P(w_k \mid w_{k-2} w_{k-1})$
- unigrams: no history = prior probabilities of word observation

# A Word Guessing Game

What do we learn from the word guessing game?

- for some histories the number of expected words is rather small.
- for some histories we can make virtually no prediction about the next word.
- the more words fit at some point the more difficult it is to recognize the correct one (more errors are possible)
- the difficulty of recognizing a word sequence is correlated with the "branching degree"

# Bigrams and Trigrams

Are Bigrams / Trigrams any good?

First experiment:

- 1.5 million words used for training
- 300,000 words used for testing
- restricted to 1,000 most frequent words

=> 23% of trigrams occurring in test corpus were absent from training corpus

Second experiment (bag of words):

- take any meaningful 10-word sentence (from dictation task)
- scramble the words into an arbitrary order
- find most probable order with trigram model

=> 63% perfect word-by-word reconstruction  
79% reconstruction that preserves meaning

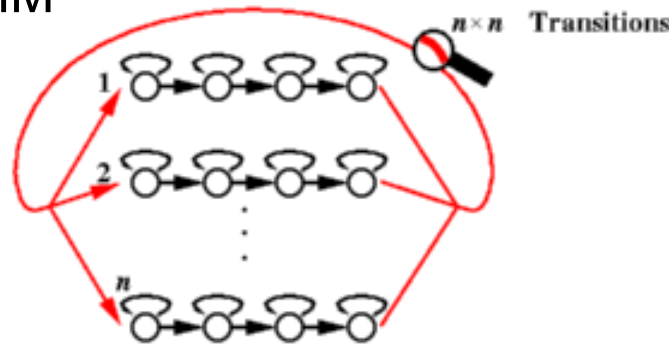
# The Bag of Words Experiment

Most likely **trigram sequences** from randomly scrambled **dictated sentence**:

- I expect that the output will improve with experience.  
I expect that the output will improve with experience.
- would I report directly to you?  
I would report directly to you?.
- now let me mention some of the disadvantages.  
let me mention some of the disadvantages now.
- these people have a fairly large rate of turnover.  
of these people have a fairly large turnover rate.
- exactly how this might be done is not clear.  
clear is not exactly how this might be done.

# Bigrams vs. Trigrams

Bigrams can be easily incorporated into an HMM recognizer:



For trigrams, we need a larger history. What if a word can have many predecessors?

Typical solution for incorporating trigrams:

- use time asynchronous search (easier to handle long history)
- for time-synchronous search: use "poor man's trigrams", i.e. consider only the predecessor's *best* predecessor instead of

Other disadvantages of trigrams compared to bigrams:

- coverage of test data is smaller than with bigrams
- estimation of  $P(w_k | w_{k-2} w_{k-1})$  is more difficult

Typical error reductions: **bigrams** 30%-50%, **trigrams** 10%-20%, **fourgrams** 3%

# Interpolation of Language Model Parameters

The standard approach to estimate  $P(w | \text{history})$  is to use a large amount of training text (There's no data like more data.) and count the occurrences of the history and the occurrences of history followed by  $w$ .

Then we estimate:

$$P(w | \text{history}) = \frac{\#(\text{history } w)}{\#(\text{history})}$$

But have a look at 300,000,000 words of Wall Street Journal text: In this very large amount of text, there are around 65,000,000 word triples that occur only once in the entire text. How do we estimate these trigrams?

How do we estimate trigrams that do not occur in the training text at all?

Solution: Estimate more frequently observable  $n$ -grams, or  $n$ -classes, and interpolate poorly estimated parameters with robustly estimated parameters



# Interpolation of Language Model Parameters

Interpolating specialized parameters with more general parameters is also called **smoothing**.

Let  $f(W)$  denote the number of occurrence of the word sequence  $W$  in the training text. Then the trigram  $P(w_3 | w_1 w_2)$  can be estimated as:

$$P(w_3 | w_1 w_2) = \lambda_1 \cdot \frac{f(w_1 w_2 w_3)}{f(w_1 w_2)} + \lambda_2 \cdot \frac{f(w_1 w_2)}{f(w_1)} + \lambda_3 \cdot \frac{f(w_1)}{\sum f(w_i)}$$

$$P(w_3 | w_1 w_2) = \lambda_1 \cdot \frac{f(w_1 w_2 w_3)}{f(w_1 w_2)} + \lambda_2 \cdot \frac{f(\mathbf{C}_1(w_1 w_2 w_3))}{f(\mathbf{C}_1(w_1 w_2))}$$

Question: How do we find good values for  $\lambda$ s?



# Language Models: N-Grams

- Predict Next Word based on History
- History is Approximated by Past two or three (generally  $n$ ) past words
  - Everything including word  $w_{i-n}$  is placed into an equivalence class
- Then Probability of next word is given by
  - Trigram:  $P(w_i | w_{i-1}, w_{i-2})$
  - Bigrams:  $P(w_i | w_{i-1})$
  - Unigrams:  $P(w_i)$
- Advantage:
  - Trainable on Large Text Databases
  - Prediction 'Soft' (Probabilities)
  - Can be Directly Combined with Acoustic Model
- Problem:
  - Need Large Text Database for each Domain

# Objective Estimation of Language Model Quality

A language model is better than an alternative one, if the probability  $P(\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n)$  with which it would generate a test corpus  $W$  is larger.

But

$$P(\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n) = \prod_{i=1}^n Q(\mathbf{w}_i / \Psi(\mathbf{w}_1, \dots, \mathbf{w}_{i-1}))$$

so a good quality measure is the LOGPROB

$$\hat{H}(\mathbf{W}) = -\frac{1}{n} \sum_{i=1}^n \log_2 Q(\mathbf{w}_i / \Psi(\mathbf{w}_1, \dots, \mathbf{w}_{i-1}))$$

# Objective Estimation of Language Model Quality

If words were generated by the language "mechanism" uniformly at random from a vocabulary of size  $V$ , then

$$Q(\mathbf{w}_i / \Psi(\mathbf{w}_1, \dots, \mathbf{w}_{i-1})) = \frac{1}{V}$$

and

$$2^{\mathbb{H}(\mathbf{w})} = 2^{\log V} = V$$

We can thus define the PERPLEXITY of the language model as:

$$PP(\mathbf{W}) = 2^{\mathbb{H}(\mathbf{w})}$$

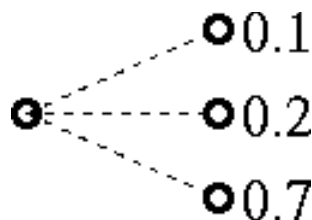
and interpret it as the "branching factor" of the language, when  $\Psi$  is available.

# The Perplexity of a Language Model

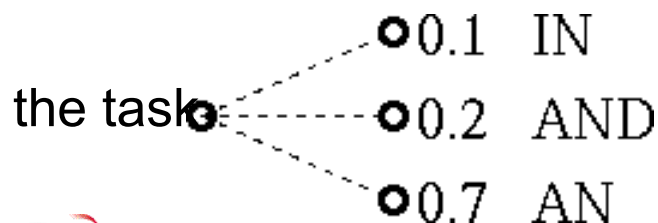
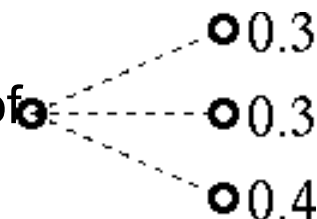
The perplexities of some tasks:

task	vocabulary	language model	perplexity
conference registration	400	bigrams	7
resource management	1000	wordpairs	60
resource management	1000	bigrams	20
Wall Street Journal	60000	bigrams	160
Wall Street Journal	60000	trigrams	130

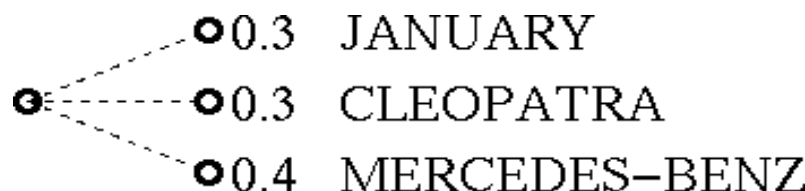
**Problem:** Even if the perplexity of



is lower than the one of



is more difficult than



# Hidden Markov Models

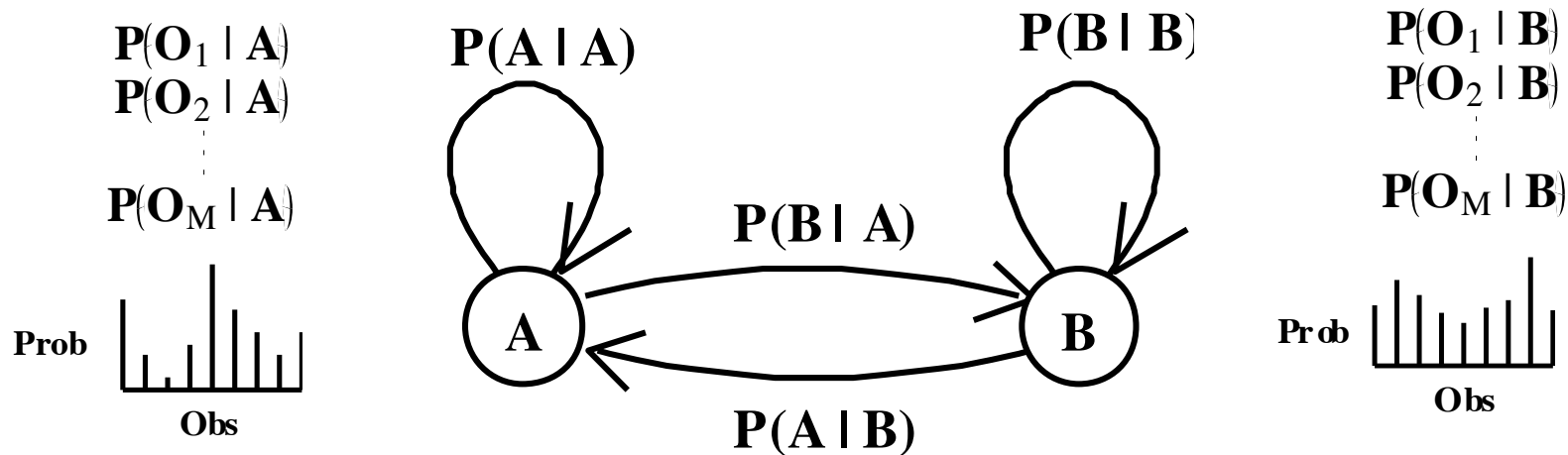
- Elements:

- States
- Transition probabilities
- Output prob distributions (at state  $j$  for symbol  $k$ )

$$S = \{S_0, S_1, \dots, S_N\}$$

$$P(q_t = S_i \mid q_{t-1} = S_j) = a_{ji}$$

$$P(y_t = O_k \mid q_t = S_j) = b_j(k)$$



# HMM Problems And Solutions

- Evaluation:
  - Problem - Compute Probability of observation sequence given a model
  - Solution - **Forward Algorithm** and **Viterbi Algorithm**
- Decoding:
  - Problem - Find state sequence which maximizes probability of observation sequence
  - Solution - **Viterbi Algorithm**
- Training:
  - Problem - Adjust model parameters to maximize probability of observed sequences
  - Solution - **Forward-Backward Algorithm**