Person-Machine Communication using speech

Renato De Mori

IEEE distinguished lecturer

LUNA IST contract no 33549

Summary

Introduction

Speech features

Acoustic Models

Language Models

Search

Adaptation

Multiple systems and features

Denoising

Confidence and results

Problems

It is difficult to conceive models of human capabilities such as:

- Signal processing,
- Phonetic and lexical decoding,
- Syntactic processing and semantic interpretation.

It is necessary to conceive models because we cannot reproduce the complexity of human processing

Conceiving good models is difficult because knowledge is limited

Applications

Command and control

Dictation, Transcription (broadcast news)

Robust systems (car, house, ...)

Spoken dialogues (call routing, question answering, customer relation services, opinion analysis, directory assistants health care)

Voice browsers (meeting browser), Information retrieval

Multimedia systems

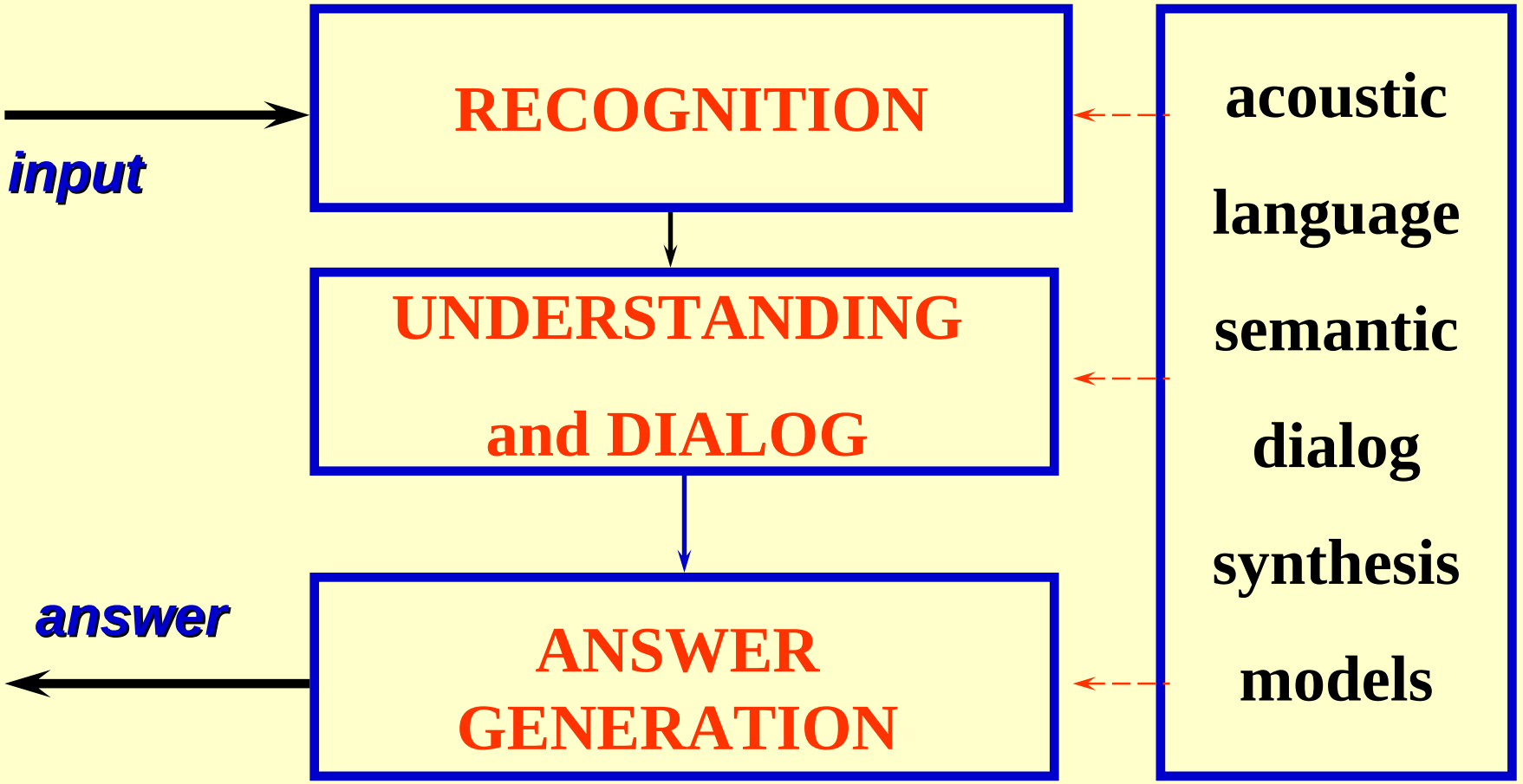
Translation

Modeling

If analytic models are incomplete because of lack of knowledge, then statistical models can be useful, because they are built from corpora and may have a complete coverage of the observations.

Statistical models are often built on top of structural models

system architecture



The recognition paradigm

Person-machine Communication (PMC) can be seen as an *exchange of information* coded in a way suitable for transmission through a physical medium.

Coding is the process of producing a representation of what has to be communicated. The content to be communicated is structured. The basic component is a *vocabulary of signs or symbols*. By concatenation of symbols, words and sentences are obtained.

Concatenations are subject to constraints described by *knowledge sources (KS)*.

Coded messages undergo further transformations that make them transmittable trough a physical channel.

The decoding process

Dictation and interpretation systems perform a *decoding process* using *KSs* to transform the message carried by a speech signal into different levels of symbolic representation. Decoding can produce word sequences or conceptual hypotheses.

The KSs used by machines in the decoding process are only *models* of the ones used by humans for producing their messages.

Sources of imprecision

The decoding process has to deal with *imprecision* due to

- distortions introduced by the transmission channel,
- the limits of the knowledge used,
- the intrinsic ambiguity of many spoken messages.

Imprecision too has to be modeled

To some extent, ambiguities can be reduced by exploiting message *redundancy*.

Model structures

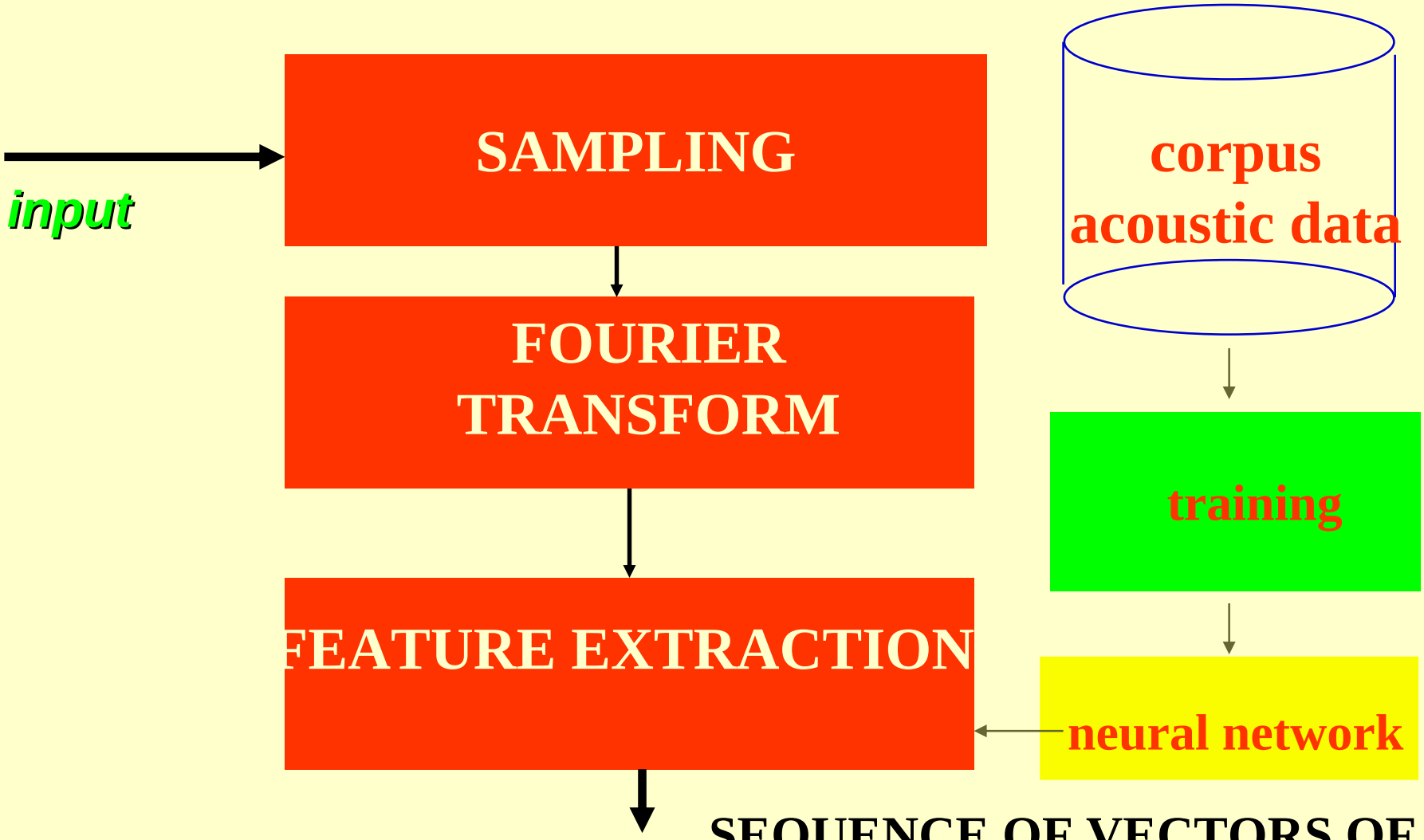
Model structures are described by formal *grammars*

Imprecision is described by augmenting grammars with *probability distributions* leading to *statistical models (soft constraints)*

Different types of models are integrated into a single statistical KS

Generation of word hypotheses is a search for the most likely match between a description of the input signal and a word sequence satisfying the constraints represented in the integrated KS

Feature extraction



Basic approach

Sampling

$$f(mT) = \int_0^{\infty} \delta(t - mT)x(t)dt$$

The z-transform

$$F(z) = \sum_{m=0}^{\infty} f(mT)z^{-m}$$

For frequency analysis:

$$z = e^{j2\pi fT}$$

Spectral representation

Classical Fourier Transform computes the spectrum of a signal weighted by a window.

$$F(k,n) = \sum_{m=0}^{N_W-1} f(nT+m)W(m)e^{-\frac{j2\pi mk}{N_W}}$$

where f(.) represent signal samples, W(n) represent window samples and F(.) represent spectral samples.

The window is defined as follows:

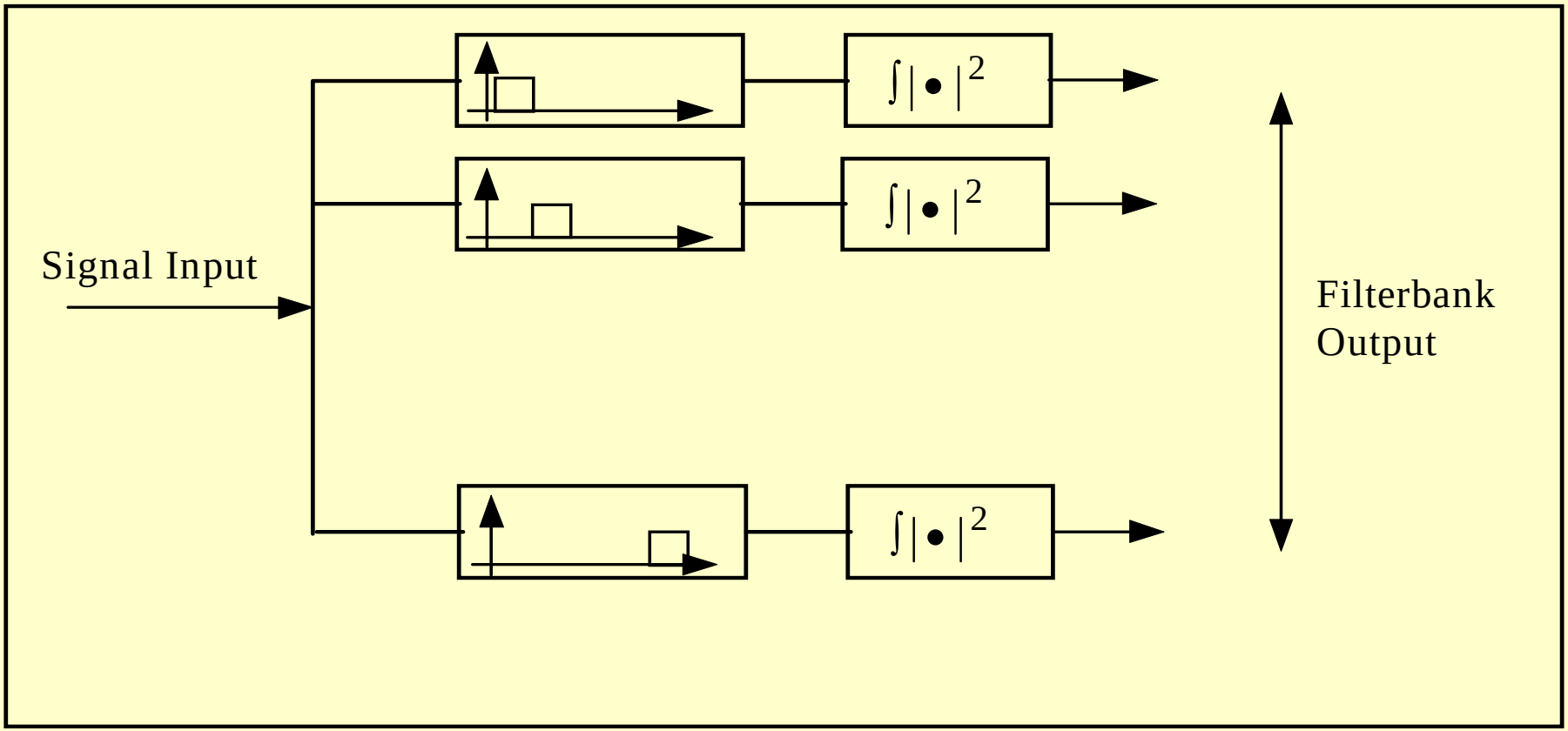
$$W(t) = 0.5\beta_W\left(1-\cos\frac{2\pi t}{N_W-1}\right)$$

The spectral energy is given by:

$$E(k,n) = |F(k,n)|^2$$

And the log spectrum is: $$S(k,n) = \log|F(k,n)|^2$$

Filterbank Based Spectral Analysis



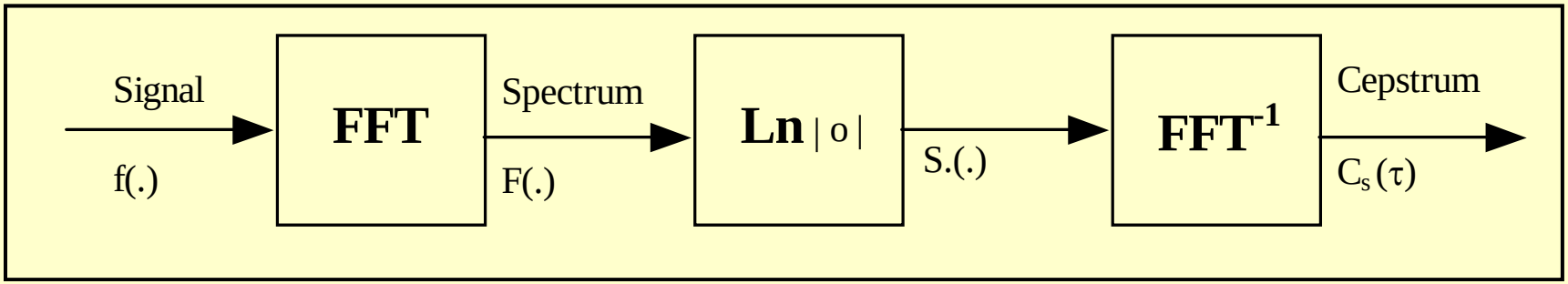
Filter bank

A filter-bank defines a time-frequency distribution in which an energy is associated with the central frequency of each filter output. Such an energy, for the j-th filter, is computed as follows:

$$E_j(n) = \sum_{k=1}^{N-1} \left| \Phi_j(k) F(k,n) \right|^2$$

where $\Phi_j(k)$ represents the frequency response of the j-th filter. Following an ear model, the magnitudes of their frequency response may have a triangular shape

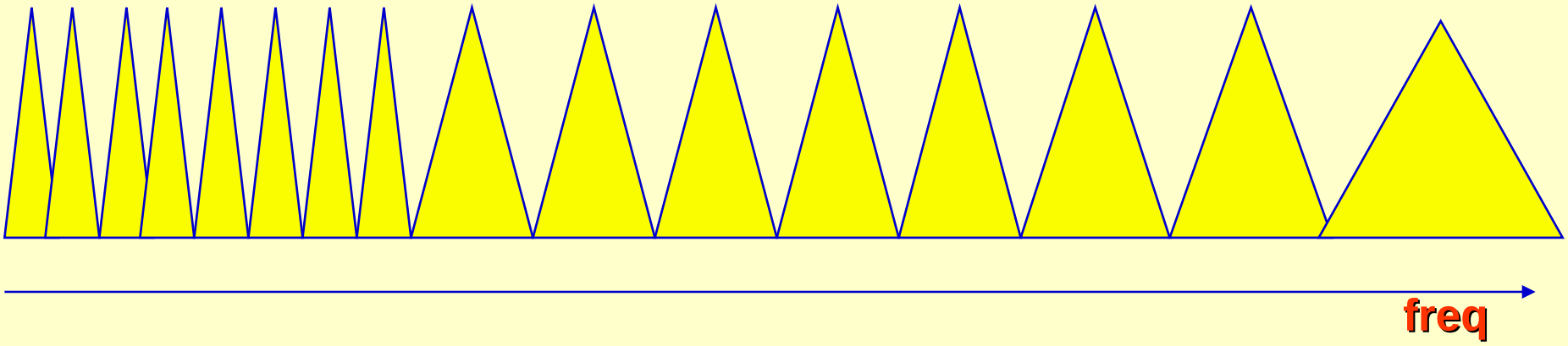
homomorphic analysis



Spectrum can be obtained by mel-scaled filters to model ear sensitivity

This is done by adding a filter bank after the FFT

Mel scaled filters



A set of filters is considered spanning the spectrum of the speech signal

They have triangular frequency response

Bandwidth increases logarithmically with frequency qualitatively reproducing ear frequency sensitivity

relation between linear frequency scale and Bark scale

$$f = \begin{cases} \dfrac{1000}{0.76}\tan\left(\dfrac{z}{13}\right) & \text{if } 0\,\text{Bark} \le z \le 10.41\,\text{Bark} \\ 1000 \bullet 10^{\frac{z-8.7}{14.7}} & \text{if } 10.41\,\text{Bark} \le z \le 17.25\,\text{Bark} \end{cases}$$

MEL frequency scaled cepstral coefficients

$$C(n,i) = \sum_{j=1}^{J} X_j(n) \cos\left[i\left(j - \frac{1}{2} \right) \frac{\pi}{M} \right]$$

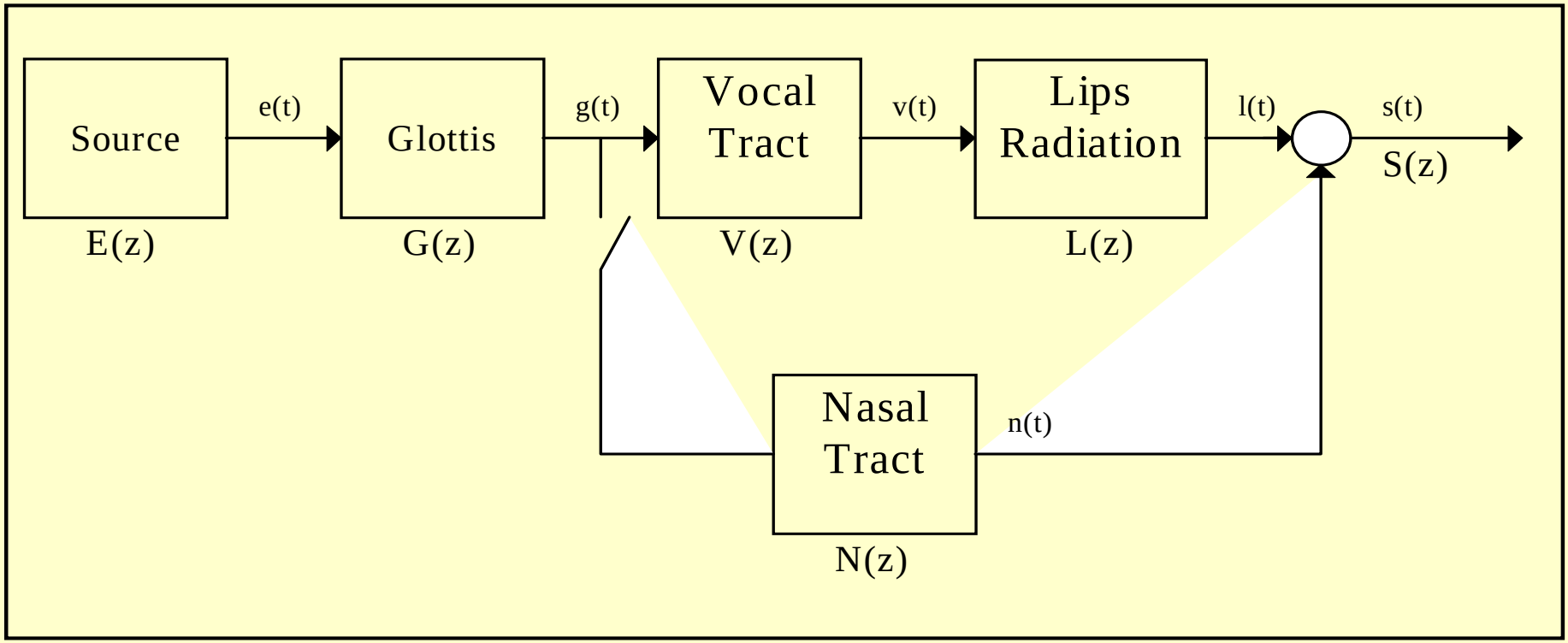
$$X_j(n) = \log E_j(n)$$

First and second time detivatives

vector A(n)

The future : feature combinations

Speech Production model



The source generates a sequence of pulses regularly spaced in time for voiced sounds and a white noise for unvoiced sounds. The glottis acts as a low-pass filter with a cutoff frequency around 100Hz for male speakers and 175Hz for female speakers.

Linear Prediction

The vocal tract can be approximated by an all-pole filter and the lips radiation by an all-zero filter or a Moving Average (MA) model. The nasal tract has a fixed model connected to the vocal tract one for producing nasalized sounds. Thus, an Auto-Regressive Moving Average (ARMA) model reliably approximates this speech production model. In general, the ARMA model is approximated by an Auto-Regressive (AR) model that is simpler and has parameters that are easy to estimate. For the sake of simplicity, the source signal $e(t)$ can be white noise.

$$f(t) = \sum_{m=1}^{p} a_m f(t-m) + \sum_{m=1}^{q} b_m g(t-m)$$

In practice, the second summation reduces to bg(t) and coefficients are estimated by MMSE

Perceptual Linear Prediction

In PLP analysis, an auditory like spectral representation is derived by warping the short-time spectrum of speech according to the Bark frequency scale of human hearing.

Procedure continues by convolving the warped spectrum with a critical band frequency masking curve, modifying the amplitude according to a typical equal loudness curve and compressing the modified spectral amplitude by a cubic-root non-linearity.

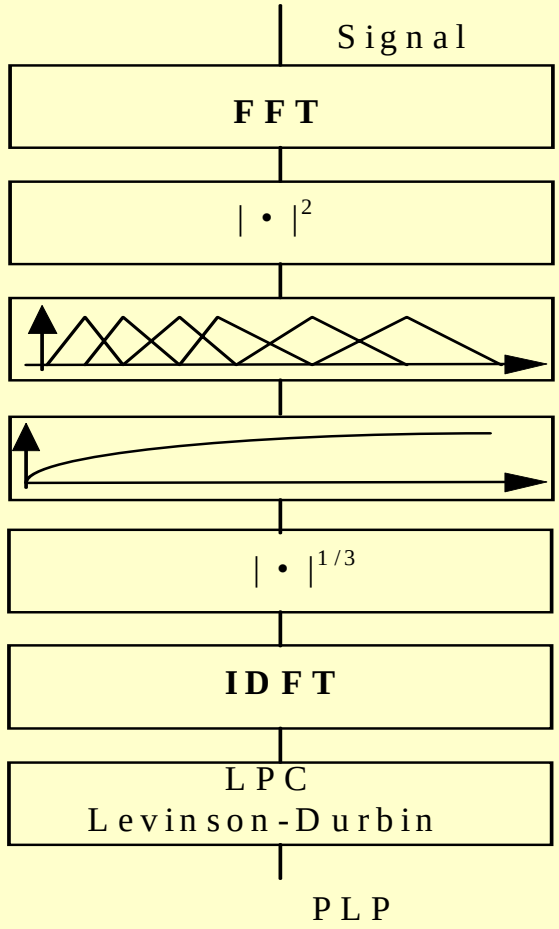
The auditory-like short-term spectrum is subsequently approximated by a low order autoregressive model. Many speaker-dependent components are reduced and robustness is introduced to slowly varying additive and convolutional errors.

Perceptual Linear Prediction

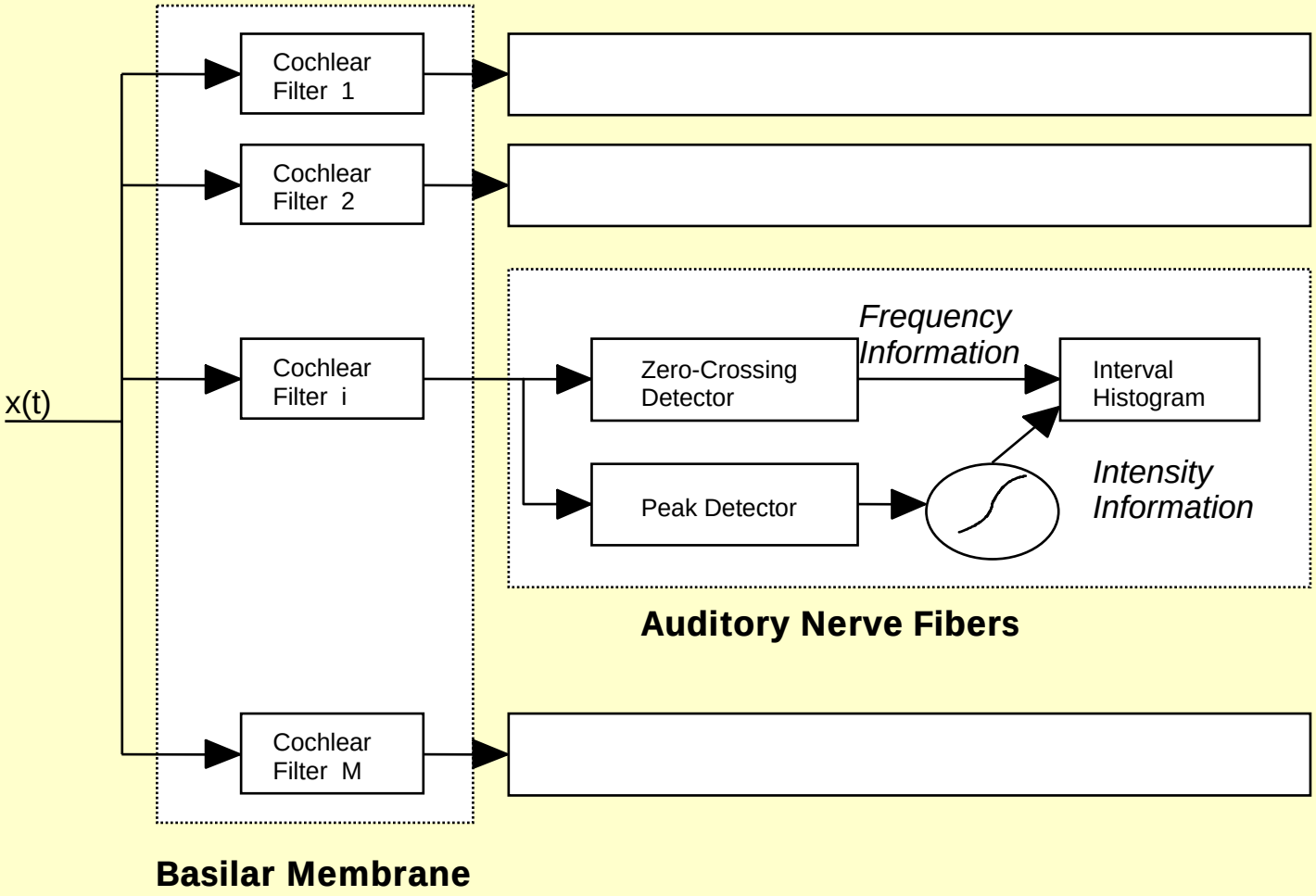
Critical band integration

Equal loudness preemphasis

Intensity-to-loudness compression



Ear models


Cochlear Filter 1
Cochlear Filter 2
Cochlear Filter i
Cochlear Filter M
x(t)
Frequency Information
Zero-Crossing Detector
Interval Histogram
Peak Detector
Intensity Information
Auditory Nerve Fibers
Basilar Membrane

Time-frequency resolutions

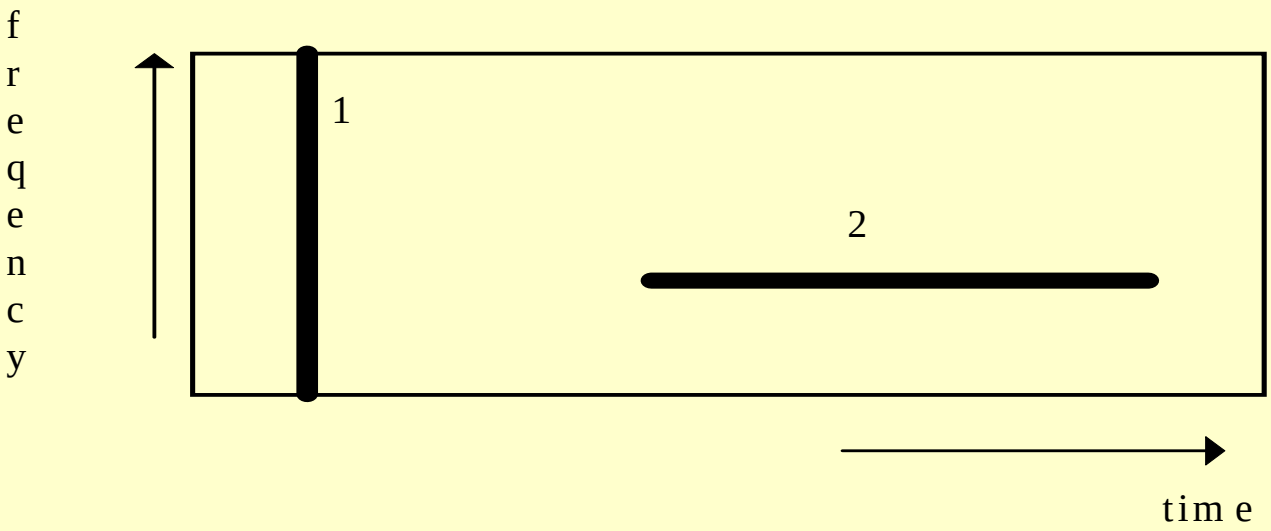


Figure 1 - Time-frequency representations

TRAPS

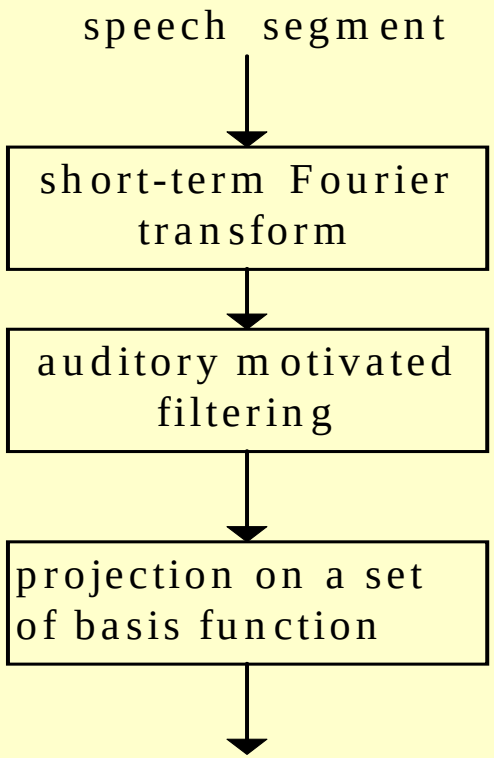


Figure 2 - acoustic features

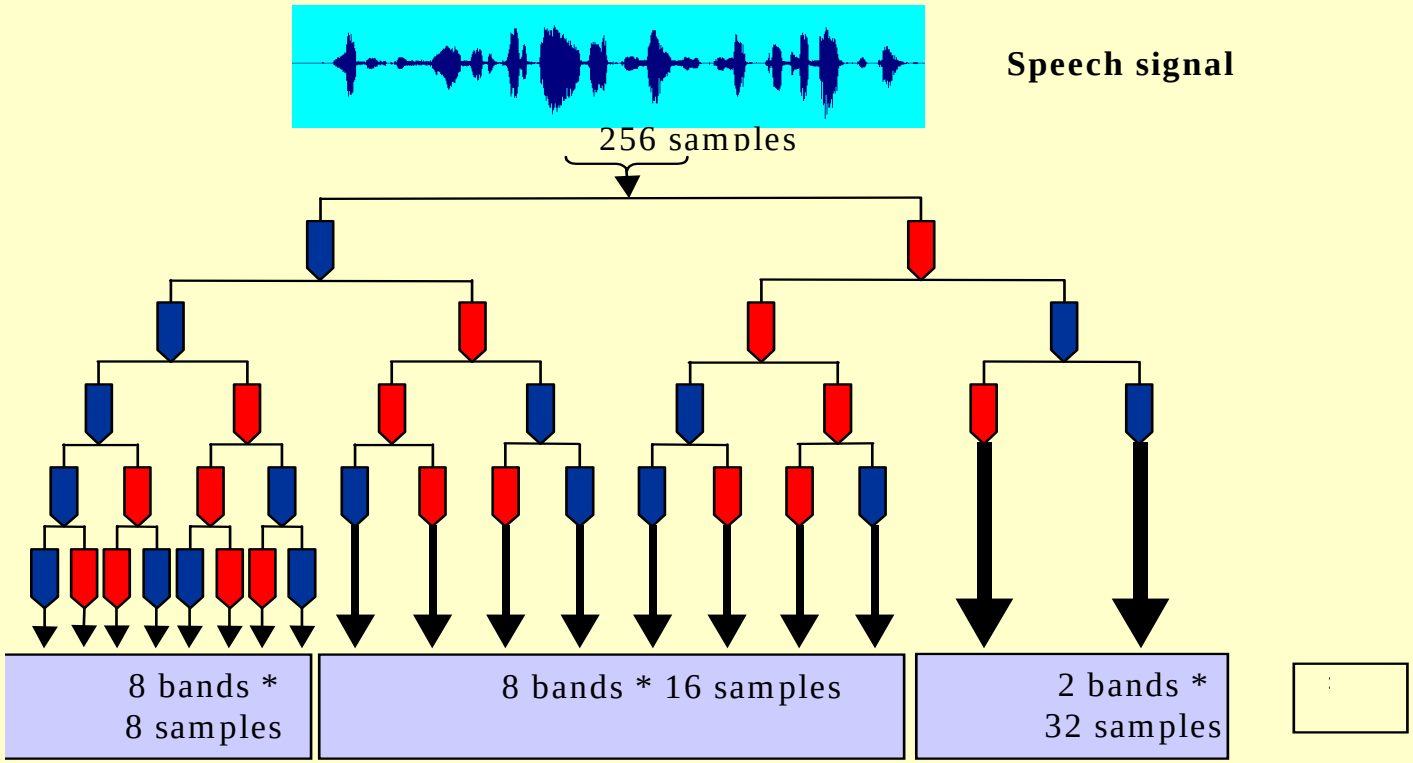
TRAPS

At OGI it is observed that TRAPS-estimated features contain 20% more information than MFCCs. Gain is shown by computing the mutual information between phones and features. It is suggested to use them in combination with MFCC for DSR. The features considered are manner of articulation:

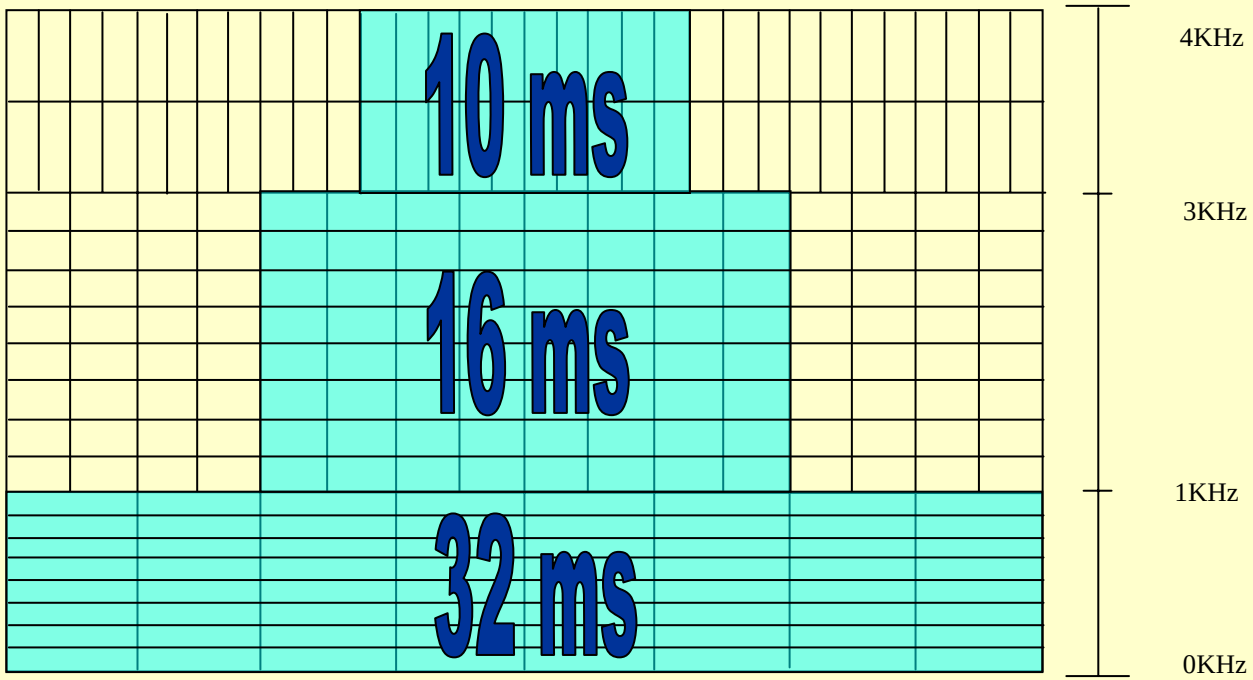
nasal, fricative, flap, stop , silence

and are computed after signal reconstruction at the server side. They are computed with a MLP having 10 inputs . The is one MLP per critical band. Training is performed on noisy TIMIT. Latency is 90 msecs and the total number of parameters is 19000.

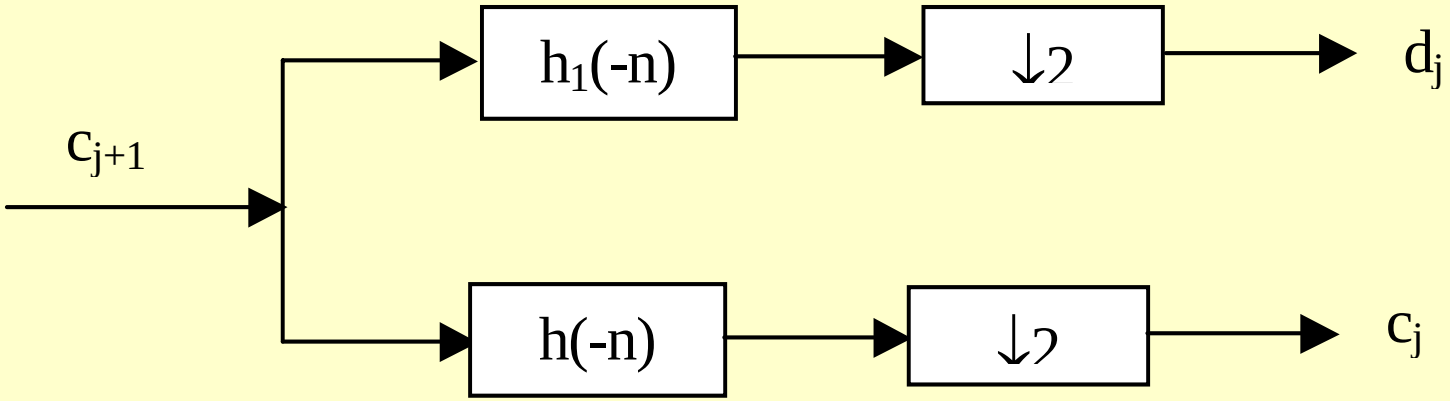
Multi-resolution analysis scheme



Time resolution



Node scheme



$$c_j(k) = \left\langle f(t), \varphi_{j,k}(t) \right\rangle = \int f(t) \cdot 2^{j/2} \cdot \varphi(2^j t - k) dt$$

$$d_j(k) = \left\langle f(t), \Psi_{j,k}(t) \right\rangle = \int f(t) \cdot 2^{j/2} \cdot \Psi(2^j t - k) dt$$

Acoustic measures

Energy

$$E = \frac{1}{N}\sum_{i=1}^{N} c_i^2$$

Norm

$$X = \frac{1}{N}\sum_{i=1}^{N} \left| c_i \right|^p$$

Average entropy

$$H = \frac{1}{N}\sum_{i=1}^{N} c_i^2 \cdot \log c_i^2$$

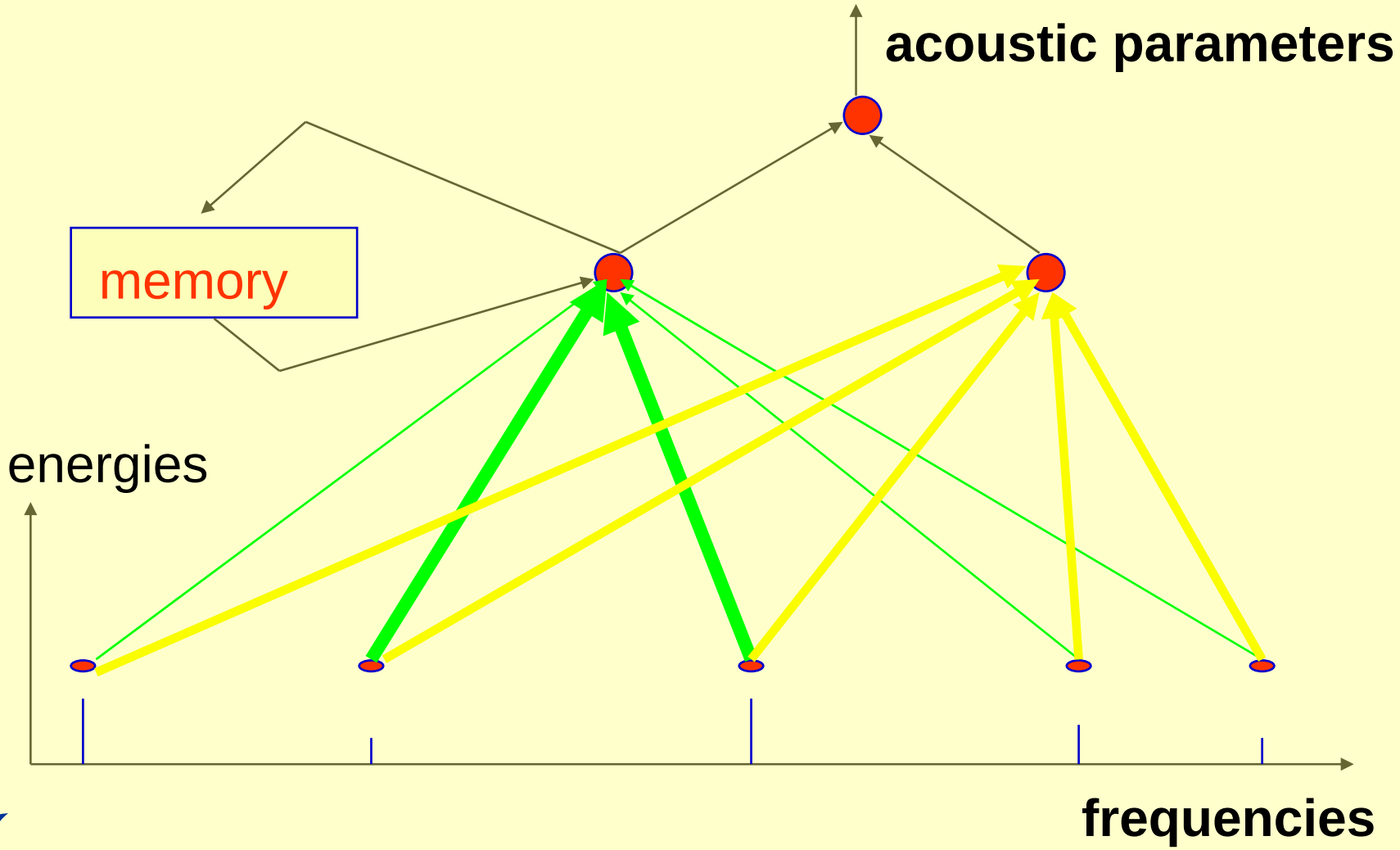
Teager operator

$$T = \frac{1}{N-2}\sum_{i=2}^{N-1} (c_i^2 - c_{i-1} \cdot c_{i+1})$$

Theoretical dimension

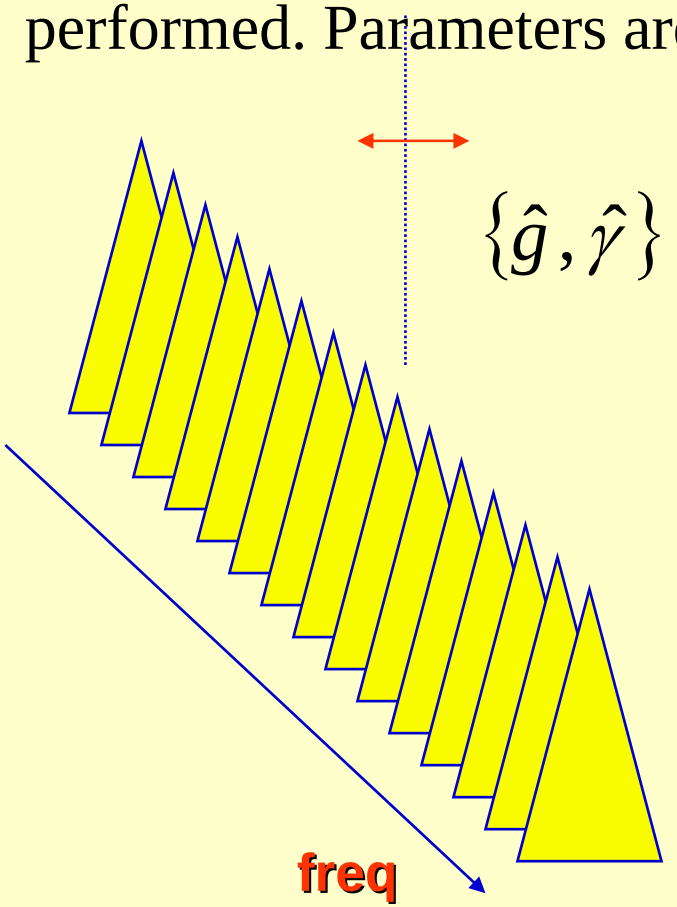
$$TD = \frac{1}{N}\left(\sum_{i=1}^{N} c_i^2 \right) \cdot e^{-\frac{\sum_{i=1}^{N} c_i^2 \cdot \log c_i^2}{\sum_{i=1}^{N} c_i^2}}$$

hybrid systems



Central frequency adaptation

A linear transformation of the filter central frequencies is performed. Parameters are estimated from data



$$\{\hat{g},\hat{\gamma}\} = \arg\max_{\{g,\gamma\}} P(A(g)/g,\gamma,\Theta_{\gamma},H)$$

g : new filters

γ : new means

H : alignment

Comments on features

Many feature sets have been proposed (review Speech Communication 2007)

The choice of features is constrained by the acoustic modelling choice

Possible improvements are obtained with

- feature transformation
- feature integration

Endpoint Detection

Consists in selecting a look-ahead method based on the fist cepstral coefficient and a filter which computes the values of an objective function :

$$F(t) = \sum_{i=-W}^{W} h(i)\, g(t-i)$$

where h is the filter impulse response and g is the energy function.

W is 7 or 13; h(i) is [-f(i), f(i)]

$$f(x) = e^{Ax}\left[K_1 \sin(Ax) + K_2 \cos/Ax)\right] + e^{-Ax}\left[K_3 \sin(Ax) + K_4 \cos/Ax)\right] + K_3 + K_6 e^{sx}$$

VAD

In (Kingsbury et al., 2002) the following approaches are considered for VAD detection :

TRAPS with critical band trajectories and median filters

TRAPS with posterior probability scaled into likelihood and used with five state HMM

five state HMM representation for speech and for non-speech are trained using

- log-energy
- degree of voicing

autocorrelation at lag i.

$$v(t) = \max_i \frac{r_i(t)}{r_0(t)}$$

Speech decoding

An effective decoder can be designed by considering the sequence of acoustic observations

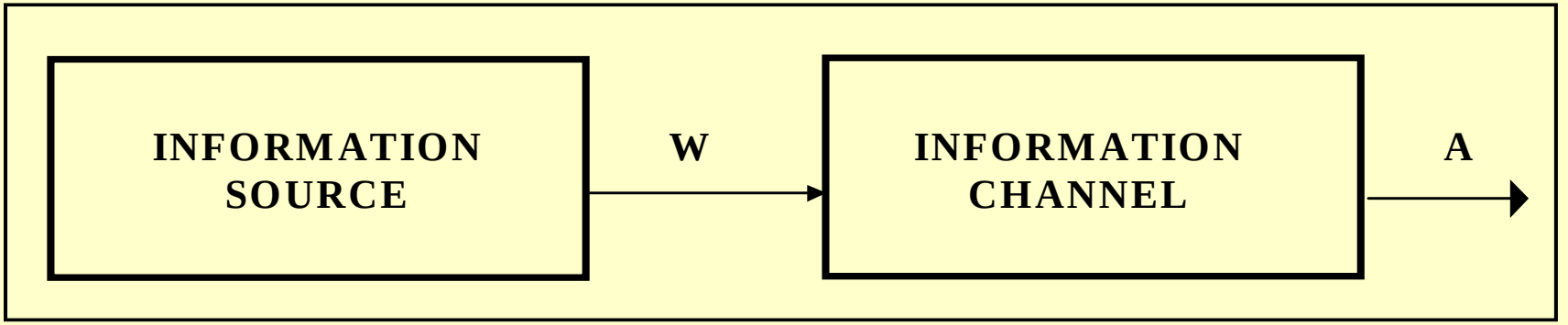
$$A = a_1 a_2 \cdots a_n \cdots a_N$$

as the output of an information channel that receives at the input a sequence of symbols representing the intention of the speaker. If these symbols are words, then they can be represented by the sequence

$$W = W_1 \ldots W_k \ldots W_K$$

reconstruction of the coding process

Considering the following coding scheme :



the objective of recognition is to reconstruct W based on the observation of A.

The source and the channel contain KSs. The source generates a variety of sequences W with a given probability distribution.

The channel, for a given W, generates a variety of A with a given probability distribution.

Decoding as search

Search for a sequence of words W such that

$$\hat{w} = \arg\max_{W} \left[P(W/A) = \frac{P(A/W)P(W)}{P(A)} \right]$$

computed by the acoustic model AM

computed by the language model LM

W : sequence of hypothesized words

A: acoustic evidence

Multi-expert decision model

Decision is based on the weighted sum of expert scores:

$$\underset{W}{\text{Max}}\left\{\log P(A / W) + \beta \log P(W)\right\}$$

fudge factor

score of the acoustic expert

score of the linguistic expert

Speech understanding

$$C' = \underset{C}{\operatorname{argmax}} \Pr(C/A) = \underset{C}{\operatorname{argmax}} \sum_{W} \Pr(CW/A) \cong$$

$$\underset{CW}{\operatorname{argmax}} \Pr(CW/A) =$$

$$= \underset{CW}{\operatorname{argmax}} \Pr(A/CW) \Pr(CW) \cong$$

$$\underset{CW}{\operatorname{argmax}} \Pr(A/W) \Pr(CW)$$

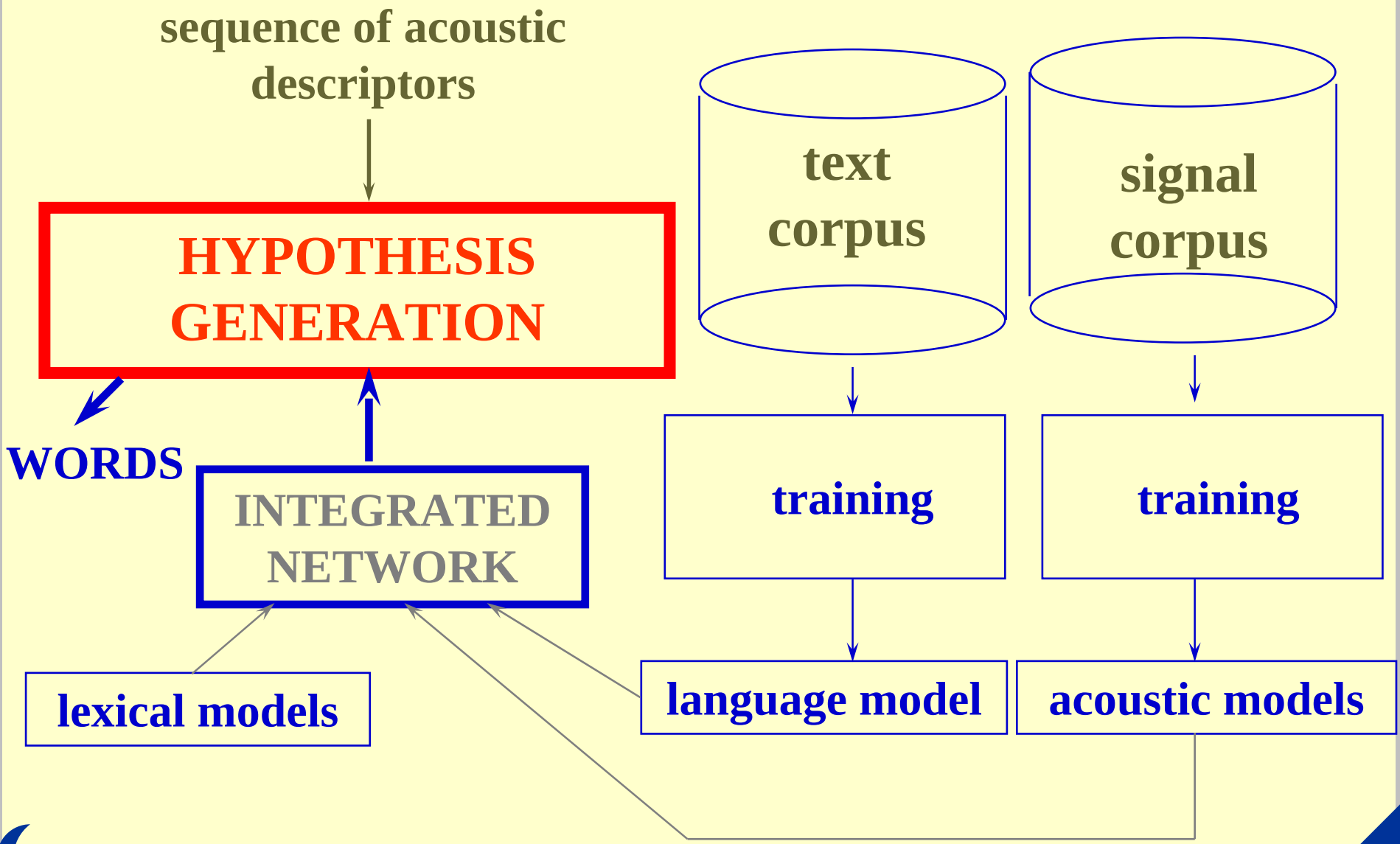
Designing KSs

Knowledge can be manually compiled or obtained by automatic *learning* from a corpus of data.

The best results so far have been obtained using component models having a simple, manually decided structure.

Statistical parameters of these models are estimated by automatic training from corpora. Complex knowledge structures are obtained by composition of basic models.

word hypothesis generation



Performance indicators

A first requirement is *coverage*. The system has to be able to recognize virtually all the sentences that can be pronounced.

Another requirement is *precision*. KSs and methods for their use should produce the lowest recognition or understanding error rates.

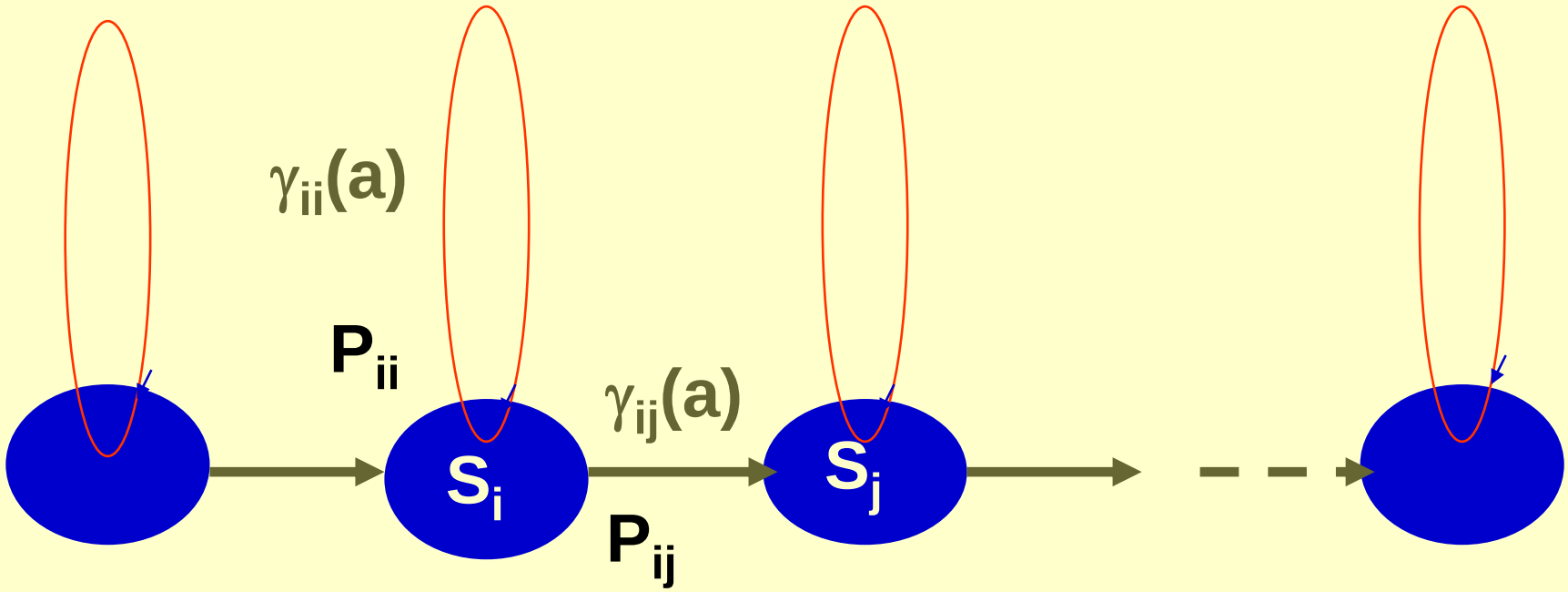
A third requirement is acceptable *computational complexity*, both in terms of *time* and *space*. Having responses close to real-time is a necessary condition. Linear time algorithms are preferred.

left to right models

$\gamma_{ij}(a)$ is the probability of observing a during transition

$i \longrightarrow j$

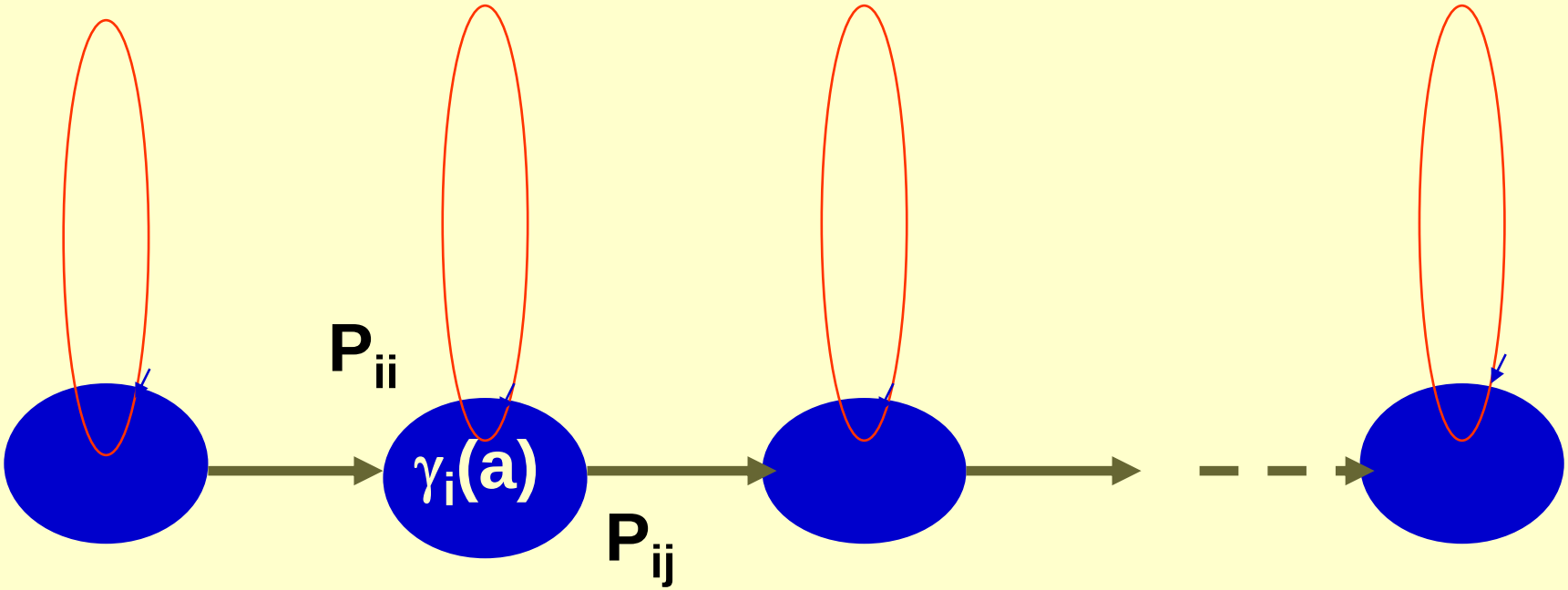
$P_{ii} + P_{ij} = 1$



left to right models alternate representation

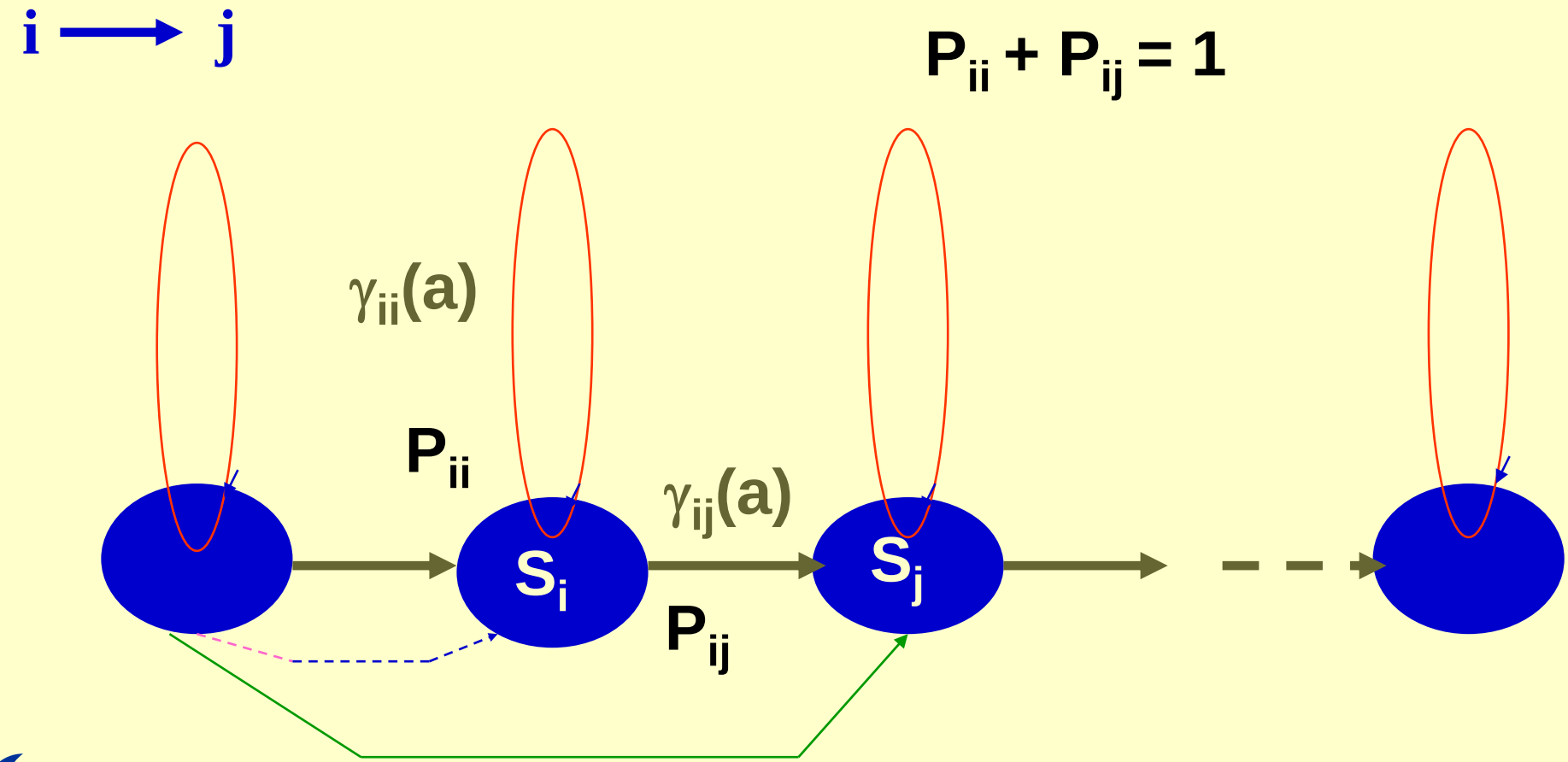
$\gamma_i(a)$ is the probability of observing a in state i

$P_{ii} + P_{ij} = 1$



left to right models with skips and emplty transitions

$\gamma_{ij}(a)$ is the probability of observing a during transition



Discrete and continuous HMMs

Discrete models

a is a symbol representing a vector of parameters or a set of Q symbols describing various aspects of the vector $\{x_q\}$

$$\gamma_{ij}(a) = b_{ij}(a) \quad \text{or} \quad \sum_{q=1}^{Q} b_{ijq}(x_q)$$

Continuous models

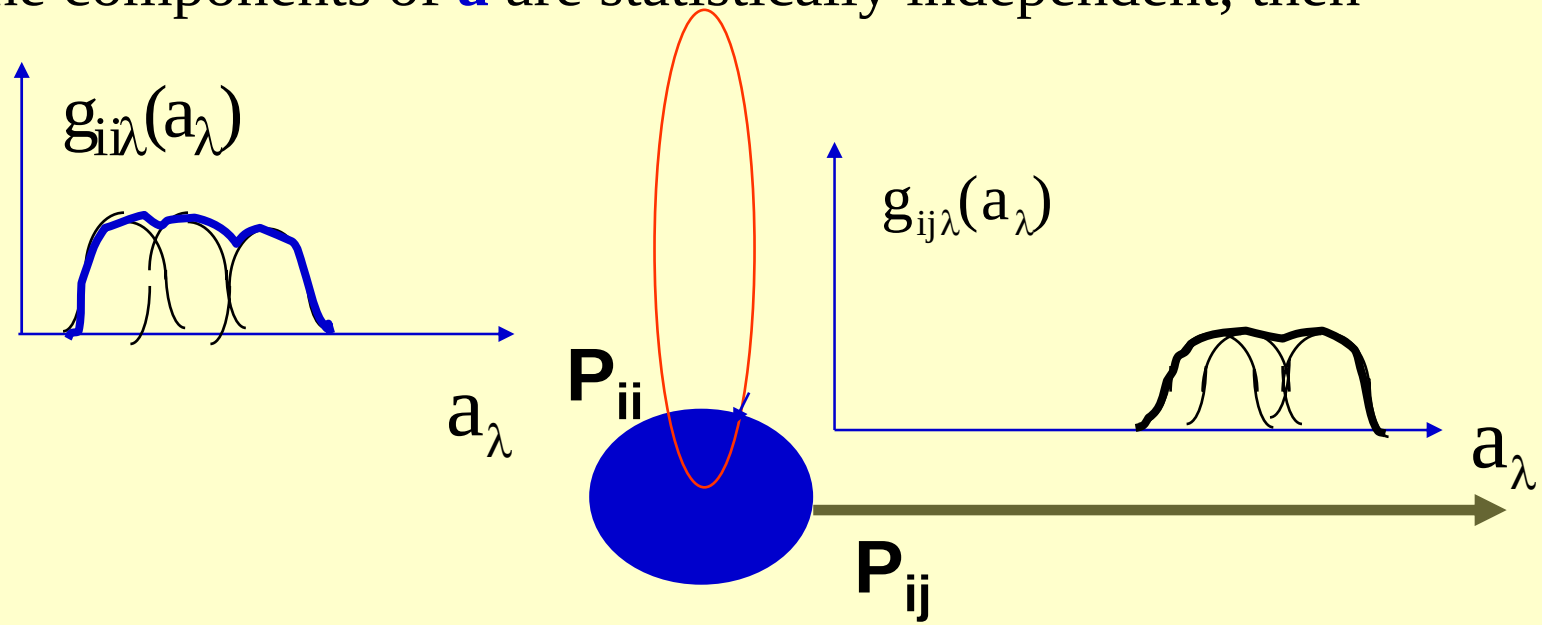
a is a vector of continuous parameters, function can be gaussian

$$\gamma_{ij}(a) = N(\mu_{ij}, \sigma_{ij}, a)$$

Mixture densities

$$\gamma_{ij}(a) = \sum_{g=1}^{G} w_{ijg} N(\mu_{ijg}, \sigma_{ijg}, a)$$

If the components of **a** are statistically independent, then



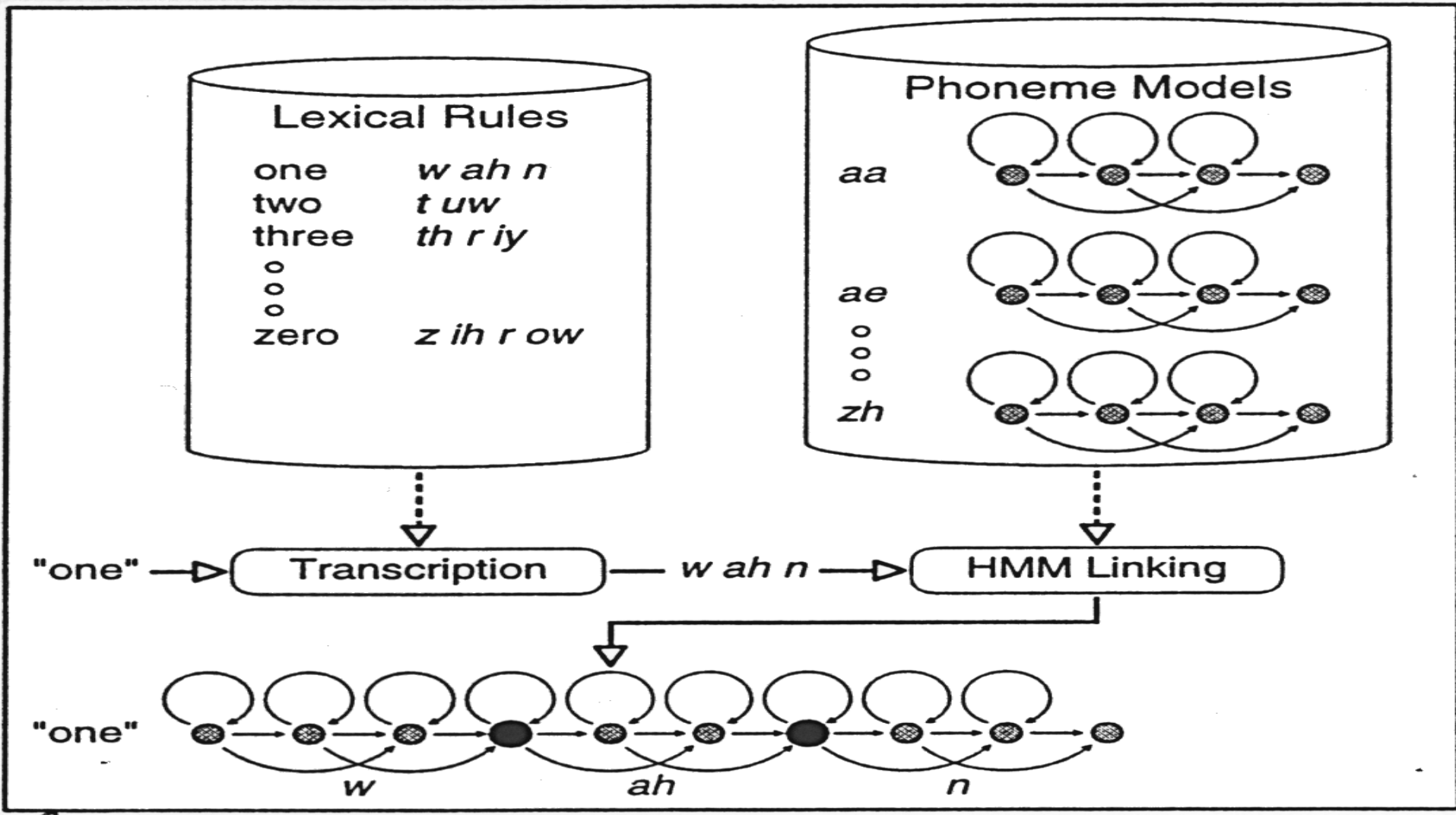
Semicontinuous HMM

Models share the same mixture of G gaussians. What makes the difference between distributions are the weights.

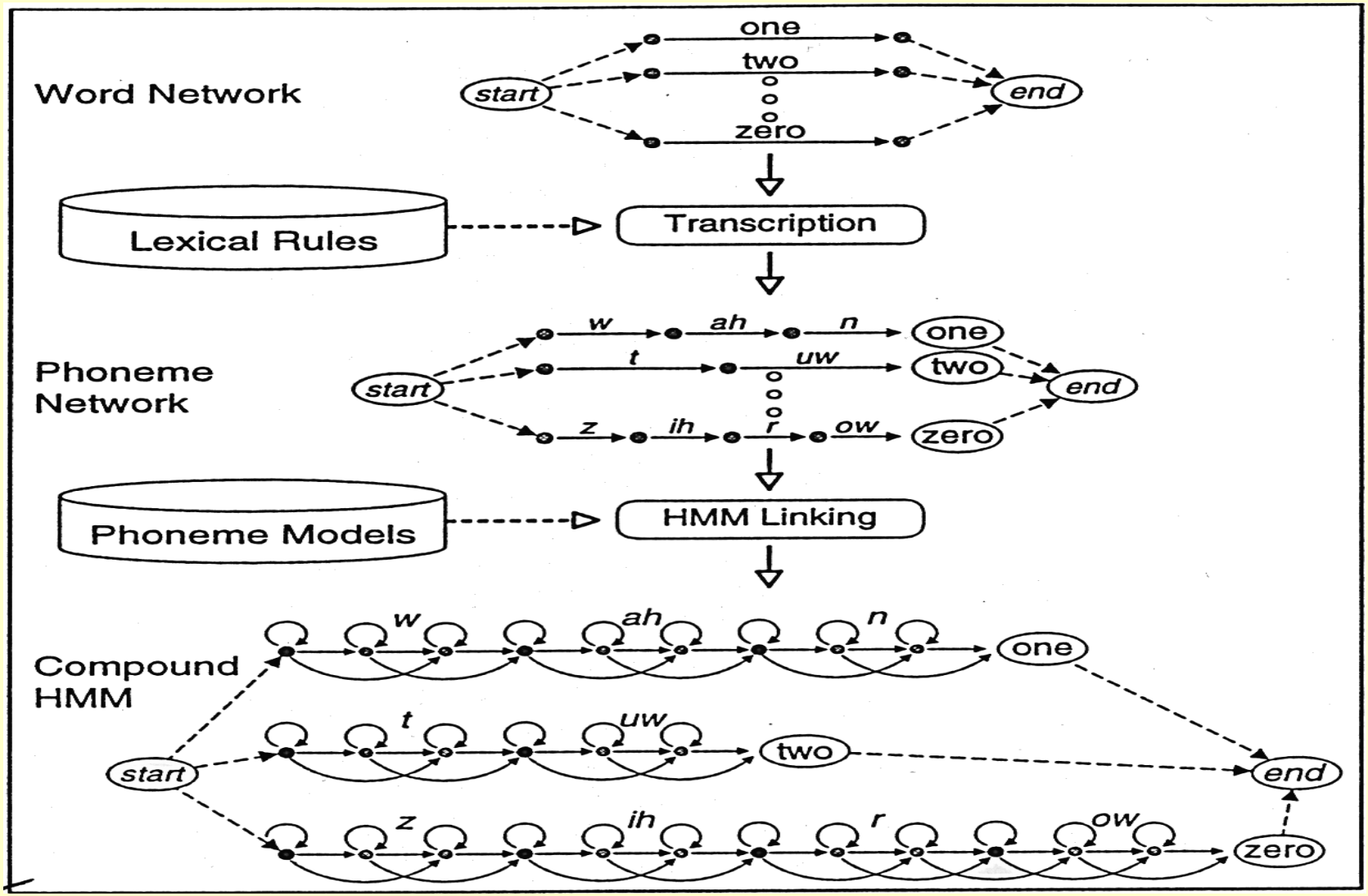
$$\gamma_{ij}(a) = \sum_{g=1}^{G} w_{ijg} N(\mu_g, \sigma_g, a)$$

$$N(\mu_g, \sigma_g, a_\lambda) = \frac{1}{2\pi\sigma_g} e^{\frac{(a\lambda - \mu_g)}{2\sigma_g}}$$

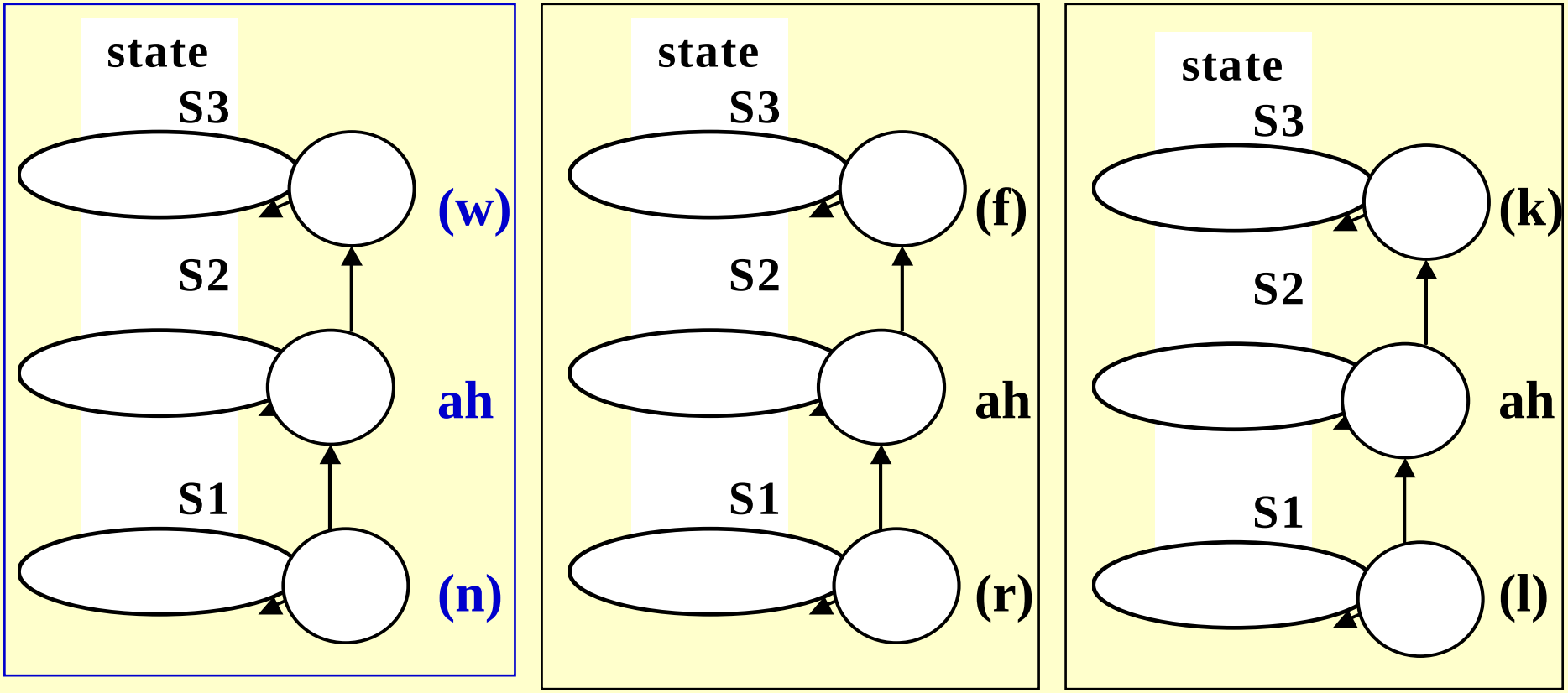
Word models


Lexical Rules
one w ah n
two t uw
three th r iy
zero z ih r ow
Phoneme Models
aa
ae
zh
"one"
Transcription
w ah n
HMM Linking
"one"
w
ah
n

Grammars


Word Network
start
one
two
zero
end
Lexical Rules
Transcription
Phoneme Network
w
ah
n
t
uw
z
ih
r
ow
Phoneme Models
HMM Linking
Compound HMM

Context dependent models : triphones



Allophones of ah

Basic problems

1 Scoring problem

Given the observation sequence A and a model, how do we efficiently compute P(A|model)?

2 Alignment problem

Given the observation sequence, how do we choose the most likely state sequence that generated it?

3 Learning problem

How do we adjust the model parameters to maximize P(A|model)?

The scoring problem

Given a fixed state sequence $I = I_1,.....,I_t,.....,I_T$ then:

$$P(A \mid I,\lambda) = b_1(a_1)\cdot b_2(a_2)\cdot....\cdot b_t(a_t)\cdot....\cdot b_T(a_T)$$

$$P(I \mid \lambda) = \pi_1 \cdot q_{I_1I_2} \cdot q_{I_2I_3} \cdot..... \cdot q_{I_{t-1}I_t} \cdot..... \cdot q_{I_{T-1}I_T}$$

$$2T \bullet K^T$$

$$P(A, I \mid \lambda) = \pi_1 \cdot b_1(a_1)\cdot q_{I_1I_2} \cdot b_2(a_2)\cdot q_{I_2I_3} \cdot b_3(a_3)\cdot...$$

$$..\cdot q_{I_{t-1}I_t} \cdot b_t(a_t)\cdot..... \cdot q_{I_{T-1}I_T} \cdot b_T(a_T) =$$

$$= P(A \mid I,\lambda)\cdot P(I \mid \lambda)$$

$$P(A \mid \lambda) = \sum_{\text{all } I} P(A, I \mid \lambda)$$

Calculation requires $2T \bullet K^T$ steps

Recursive computation

Assuming models are left-to-right, define

Forward coefficient:

$$\alpha_t(a_1^T, k) = \sum_{j \in Pred(s_k)} q_{jk} \cdot b_k(a_t) \cdot \alpha_{t-1}(a_1^T, j)$$

Backward coefficient

$$\beta_t(a_1^T, k) = \sum_{j \in Succ(s_k)} q_{kj} \cdot b_j(a_{t+1}) \cdot \beta_{t+1}(a_1^T, j)$$

$$P(A \mid \lambda) = \sum_{\text{all } k} \alpha_T(a_1^T, k) = \sum_{\text{all } k} \beta_0(a_1^T, k) = \sum_{\text{all } k} \alpha_t(a_1^T, k) \cdot \beta_t(a_1^T, k)$$

Complexity : K T

Phonetically Tied Mixture (PTM)

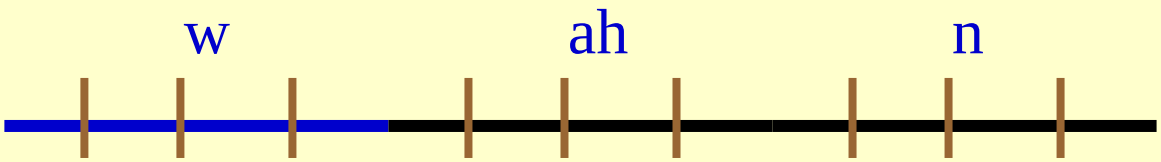
All the allophones of the same phoneme share the same set of gaussians but with different weights. For all distributions of allophones of phoneme f :

$$\gamma_{ij}^{f}(a) = \sum_{k_f=1}^{K_f} w_{ijk_f} N(\mu_{ijk_f}, \sigma_{ijk_f}, a)$$

Continuous densities outperform conventional tied mixtures (SCHMM) by about 20%

HMM training

Start with an arbitrary segmentation of the sentence



Basic recursion

Estimate model parameters based on samples assigned to each distribution

Perform recognition with only the sentence model to obtain new alignment

HMM training

Consider the function : $P(\vartheta) = \int p(\xi, \vartheta) d\xi$

and the auxiliary function:

$$Q(\vartheta, \vartheta') = \frac{1}{P(\vartheta)} \int p(\xi, \vartheta) \log p(\xi, \vartheta') d\xi$$

It is possible to prove that:

$$Q(\vartheta, \vartheta') - Q(\vartheta, \vartheta) \leq \log P(\vartheta') - \log P(\vartheta)$$

Baum-Welch algorithm

Consider the following transformation:

$$\Gamma(\vartheta) = \arg\max_{\vartheta'} Q(\vartheta, \vartheta')$$

This is a *growth* transformation for the function P :

$$P\{\Gamma(\vartheta)\} \geq P(\vartheta)$$

By successively applying the transformation, a sequence $\vartheta_n = \Gamma(\vartheta_{n-1})$ that monotonically increases the objective function P can be found.

Maximum Likelihood Estimation

Let $\vartheta=(\pi,A,B)$ be the set of the parameters of all the HMMs

Let x_1^T be the sequence of all the training data

Let $\Psi\left(i_0^T\right)$ be the set of possible sequences of T states

$$P(\vartheta) = Pr_\vartheta(x_1^T) = \sum_{\xi\in\Psi\left(i_0^T\right)} Pr_\vartheta(x_1^T,\xi) = \sum_{\xi\in\Psi\left(i_0^T\right)} p(\vartheta,\xi)$$

$$p(\vartheta,\xi) = \pi_{i_0} \prod_{t=1}^{T} a_{i_{t-1},i_t}\, b_{i_{t-1},i_t}(x_t) \qquad \text{for model } \vartheta$$

Maximum Likelihood Estimation

$$Q(\vartheta,\vartheta') = \frac{1}{P(\vartheta)} \sum_{\xi \in \Psi\left(i_0^T\right)} p(\vartheta,\xi)\left\{\log \pi'_{i_0} + \sum_{t=1}^{T} a'_{i_{t-1},i_t} + \sum_{t=1}^{T} b'_{i_{t-1},i_t}(x_t)\right\}$$

$$Q(\vartheta,\pi') = \frac{1}{P(\vartheta)} \sum_{\xi \in \Psi\left(i_0^T\right)} p(\vartheta,\xi) \log \pi'_{i_0}$$

$$Q(\vartheta,A') = \frac{1}{P(\vartheta)} \sum_{\xi \in \Psi\left(i_0^T\right)} p(\vartheta,\xi)\left\{\sum_{t=1}^{T} a'_{i_{t-1},i_t}\right\}$$

$$Q(\vartheta,B') = \frac{1}{P(\vartheta)} \sum_{\xi \in \Psi\left(i_0^T\right)} p(\vartheta,\xi)\left\{\sum_{t=1}^{T} b'_{i_{t-1},i_t}(x_t)\right\}$$

Maximum Likelihood Estimation

By deriving $Q(\vartheta, A')$ with respect to each a'_{ij} and using Lagrange multipliers, the following re-estimation formula is obtained:

$$a'_{ij} = \frac{\gamma(i,j)}{\sum_{l \in set\_of\_states} \gamma(i,l)} \qquad b'_{ij}(x) = \frac{\sum_{t=1}^{T} \gamma_t(i,j)\delta(x,x_t)}{\gamma(i,j)}$$

$$\gamma(i,j) = \sum_{t=1}^{T} \gamma_t(i,j)$$

$$\gamma_t(i,j) = \frac{\alpha_{t-1}(x_1^T, i)a_{ij}b_{ij}(x_t)\beta_t(x_1^T, i)}{Pr_{\vartheta}(x_1^T)}$$

Maximum Mutual Information Estimation

Given a statistical model of a speech unit *w* (e.g. a word or a phoneme) characterized by a set of statistical parameters specifying its statistical distributions, and a sequence X of acoustic descriptors (feature vectors or symbols), the posterior probability that model has produced the observed sequence X can be expressed as:

$$P_\vartheta(m_w / X) = \frac{P_\vartheta(X / m_w)P(w)}{\sum_{w' \in V} P_\vartheta(X / m_{w'})P(w')}$$

MMIE

Effective training algorithms exist for MLE that are not applicable to MMIE for which classical gradient descent algorithms with related convergence problems are used. Furthermore, the presence of the denominator makes exact computation practically impossible if the size of V is too large.

For what concerns the convergence speed of gradient descent, various solutions have been proposed that are reviewed in (Normandin et al., 1994). For what concerns approximations of the denominator for the case of large vocabularies, various solutions have been proposed using the N-best lists of word hypotheses (Chow, 1990), phoneme lattices (Normandin et al., 1994) or word lattices (Valtchev et al., 1994).

minimum phone error or classification error (MPE/MCE)

MPE reduces the training set estimated phone error (in a word recognition context) and has been shown to outperform MMIE

.

$$\mathcal{F}_{MPE}(\lambda) = \sum_{r=1}^{R} \frac{\sum_{\hat{w}} P_\lambda(O_r|\mathcal{M}^{\hat{w}})^\kappa P(\hat{w}) RawAccuracy(\hat{w})}{\sum_{\hat{w}} P_\lambda(O_r|\mathcal{M}^{\hat{w}})^\kappa P(\hat{w})},$$

Language modeling

STOCHASTIC LANGUAGE MODELS

The purpose of LM is to compute the following probability :

$$P(W_1^n) = P(w_1)\prod_{i=2}^{n} P(w_i \mid w_1,....,w_{i-1}) = P(w_1)\prod_{i=2}^{n} P(w_i \mid h_i)$$

HISTORY APPROXIMATION

the *history* (W 1 ,.....,Wi-1) is represented by an *equivalence CLASS* S(W 1 ,.....,Wi-1)

$$P(W) = P(W_1) \prod_{i=2}^{n} P(W_i / S(W_{i-1} \ldots\ldots W_1))$$

S can be the state of a finite state automaton

or a word, a pair of words, a class, a pair of classes.....

History approximations

BIGRAM PROBABILITIES

$$P(W) = P(W_1) \prod_{i=2}^{n} P(W_i / W_{i-1})$$

TRIGRAM PROBABILITIES

$$P(W) = P(W_1) \prod_{i=2}^{n} P(W_i / W_{i-1} W_{i-2})$$

Maximum likelihood

$$\vartheta^{ML} = \arg\max_{\vartheta \in \Theta} P(SW_1^n, \vartheta)$$

$$\frac{\partial L}{\partial \vartheta_w} = \frac{c(w)}{\vartheta_w} - \lambda = 0$$

$$\sum_{w \in V} c(w) = \lambda \bullet \sum_{w \in V} \vartheta_w = \lambda$$

$$\vartheta_w^{ML} = \frac{c(w)}{\lambda} = \frac{c(w)}{\sum_{w \in V} c(w)} = \frac{c(w)}{n}$$

Backing-off scheme for a bigram (w/x)

$$P(w/x)=\begin{cases} f'(w/x) & \text{if} \quad c(wx)>0 \\ K_x\lambda(x)P(w) & \text{if} \quad c(wx)=0 \wedge c(x)>0 \\ P(w) & \text{if} \quad c(x)=0 \end{cases}$$

f' is the discounted frequency distribution λ is the zero frequency probability, c are counts

Trigram Probabilities

$$P(w/xy) = q_1 f(w/xy) + q_2 f(w/x) + q_3 f(w)$$

the q coefficients can be determined by interpolation methods

Part Of Speech (POS) models

$$P(W) = P(W_1 / g_1) \prod_{i=2}^{n} P(W_i / g_i) P(g_i / g_{i-1} g_{i-2})$$

g can be a syntactic class like a noun or any class like a semantic one or one determined by clustering words

Wi can also be a *multiword* sequence

Entropy of a text

$$H(S) = -\left\{ \sum_{i=1}^{L} P(w_i) \log\left(P(w_i)\right) \right\}$$

$$L' = 2^{H}$$

Training set entropy

the source entropy of a set of all phrases of n word length or less is measured using an LM

$$H(S) = -\lim_{n \to \infty} \frac{1}{n} \left\{ \sum_{w_1^n} P(w_1^n) \log\left(P(w_1^n)\right) \right\}$$

Ergodic source and LOGPROB

If the source is ERGODIC and its statistical properties do not vary with time, then all the sequences with the same length have the same probability and the entropy becomes equal to:

$$H(S) = -\lim_{n \to \infty} \frac{1}{n} \log \left(P(w_1^n) \right)$$

If ENTROPY is estimated using a corpus with n words, then:

$$H(S) = -\frac{1}{n} \log \left(P(w_1^n) \right)$$

If probabilities are computed with a model then rather than entropy we have a LOGPROB:

$$LP(w_1^n) = -\frac{1}{n} \log \left(P'(w_1^n) \right) = -\frac{1}{n} \log \left(P(w_1) \prod_{i=2}^{n} P(w_i \mid w_{i-1}) \right)$$

Measuring Model Quality

Using a validation set and re-rank the N-best

Perplexity PP

measures how well a LM M predicts an unseen text T using the cross entropy of the distribution functions of M and T:

$$PP_M(T) = 2^{H(P_T;P_M)}$$

Perplexity

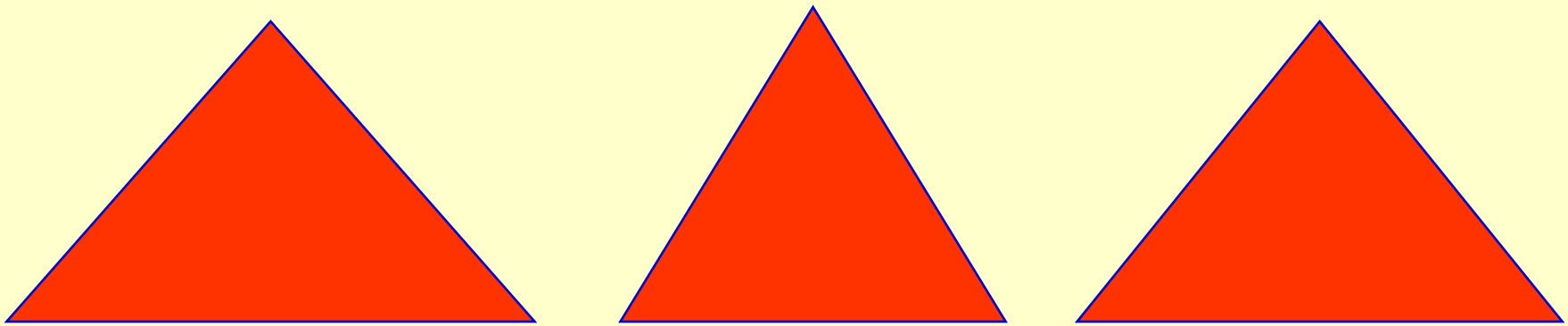
$$PP = 2^{LP(w_1^n)} = P(w_1^n)^{-1/n}$$

Perplexity can be roughly interpreted as the geometric mean of the branchout factor of the language

Head trigrams

Pr(Defence_Minister | President, appoint)

President appoint Defence_Minister



The recently elected President decided to appoint a new Defence Minister

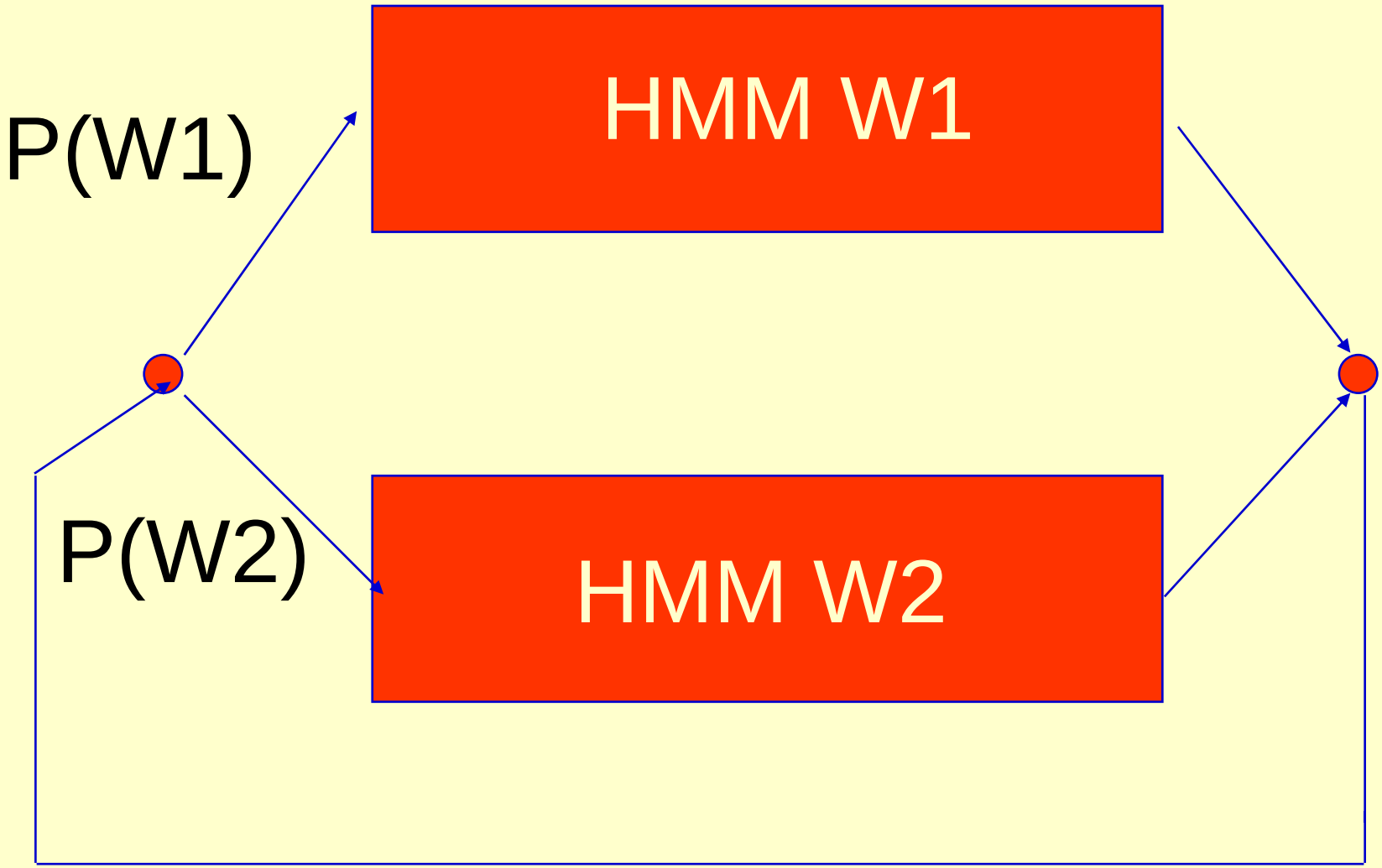
Models

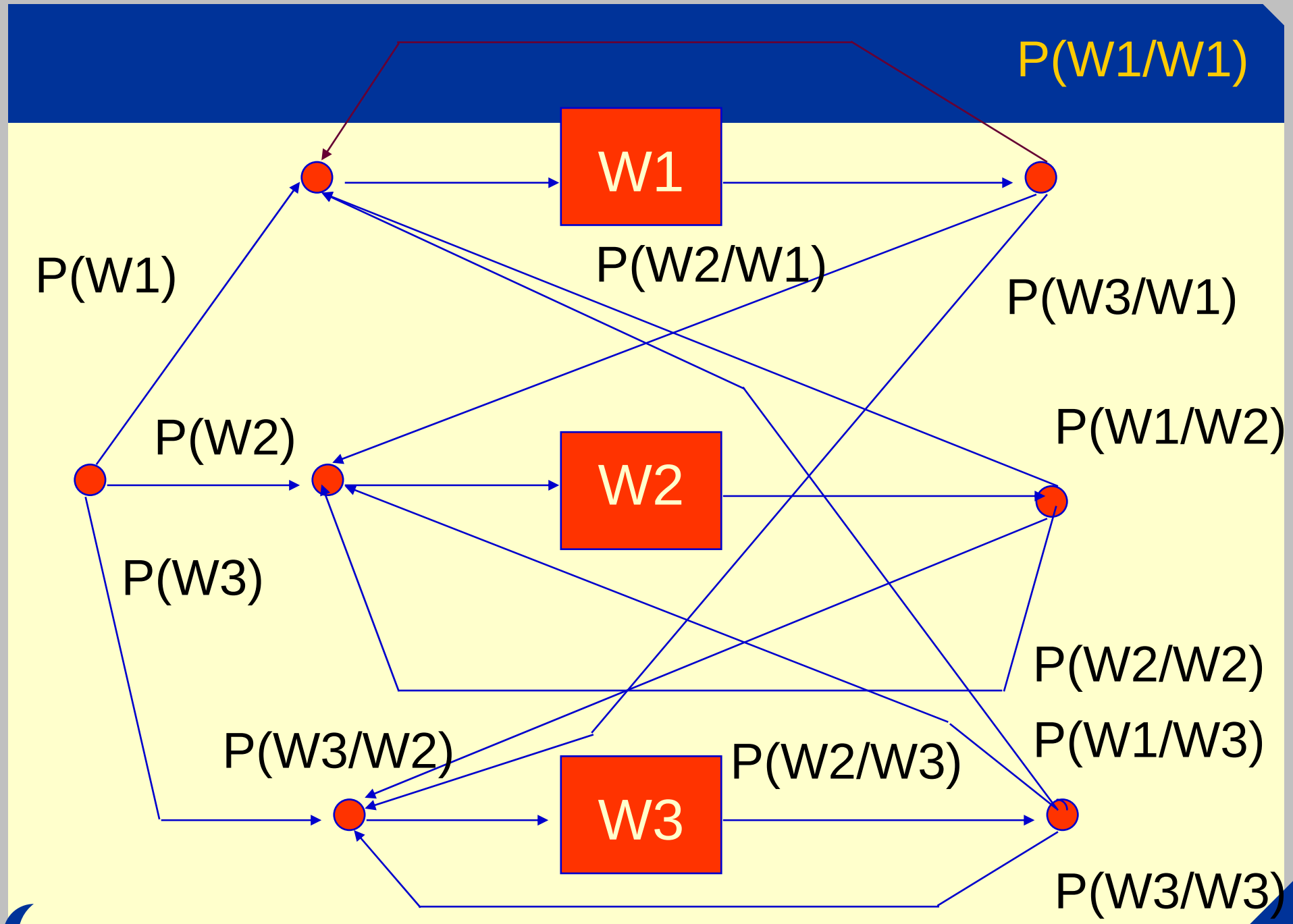
Word clustering

$$\Pr(w_k / w_1^{k-1}) = \Pr(w_k / c_k) \Pr(c_k / c_1^{k-1})$$

n-gram class models are obtained by partitioning a vocabulary of V words into C classes.

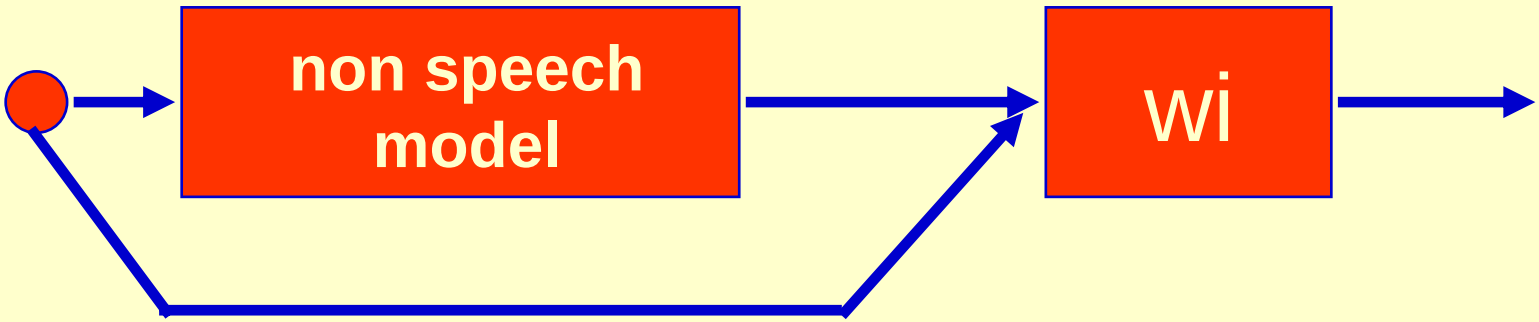
SEARCH

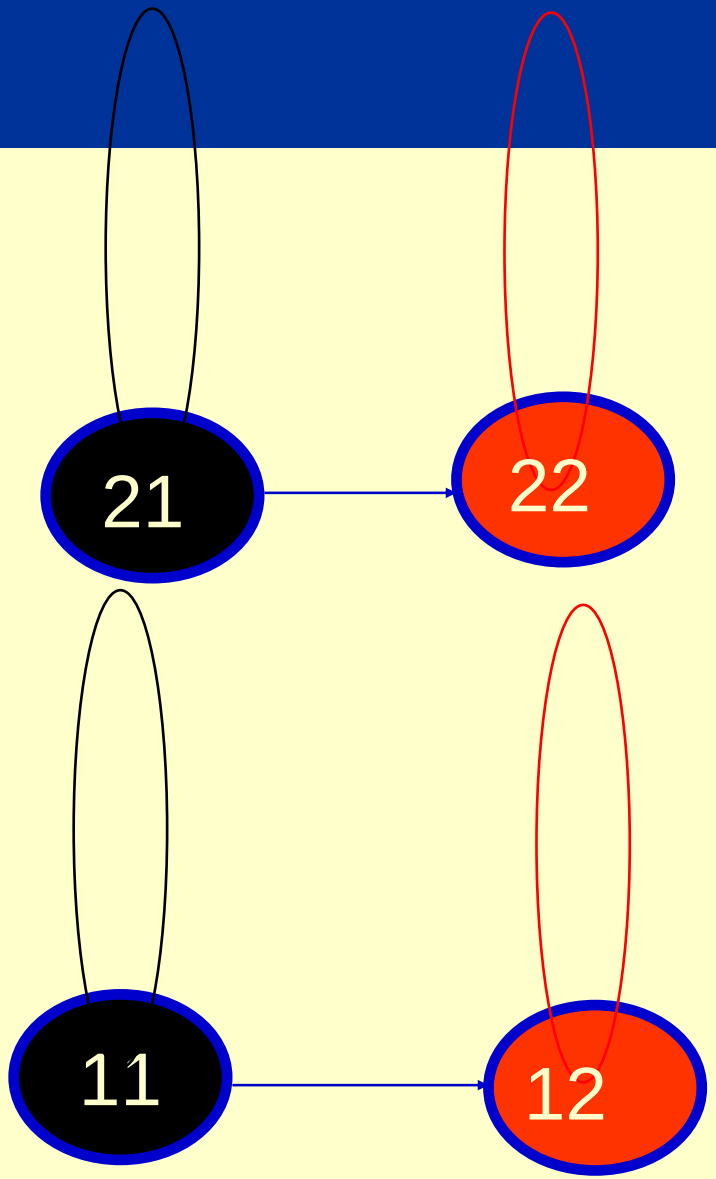



P(W1/W1)
W1
P(W1)
P(W2/W1)
P(W3/W1)
P(W2)
W2
P(W1/W2)
P(W3)
P(W2/W2)
P(W3/W2)
P(W1/W3)
P(W2/W3)
W3
P(W3/W3)

NON-SPEECH EVENTS

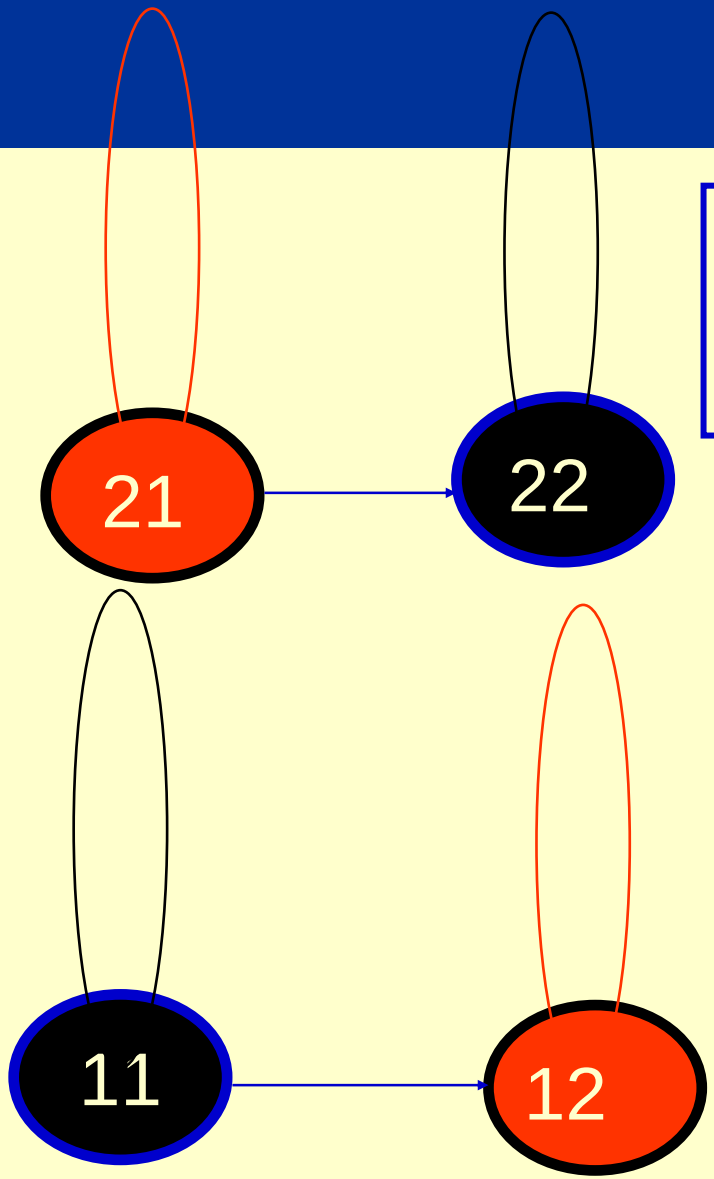
non-speech events are represented by a non speech model at the beginning of each word or before the root of a tree





state	score
11	s11(t1)
21	s21 (t1)

t1



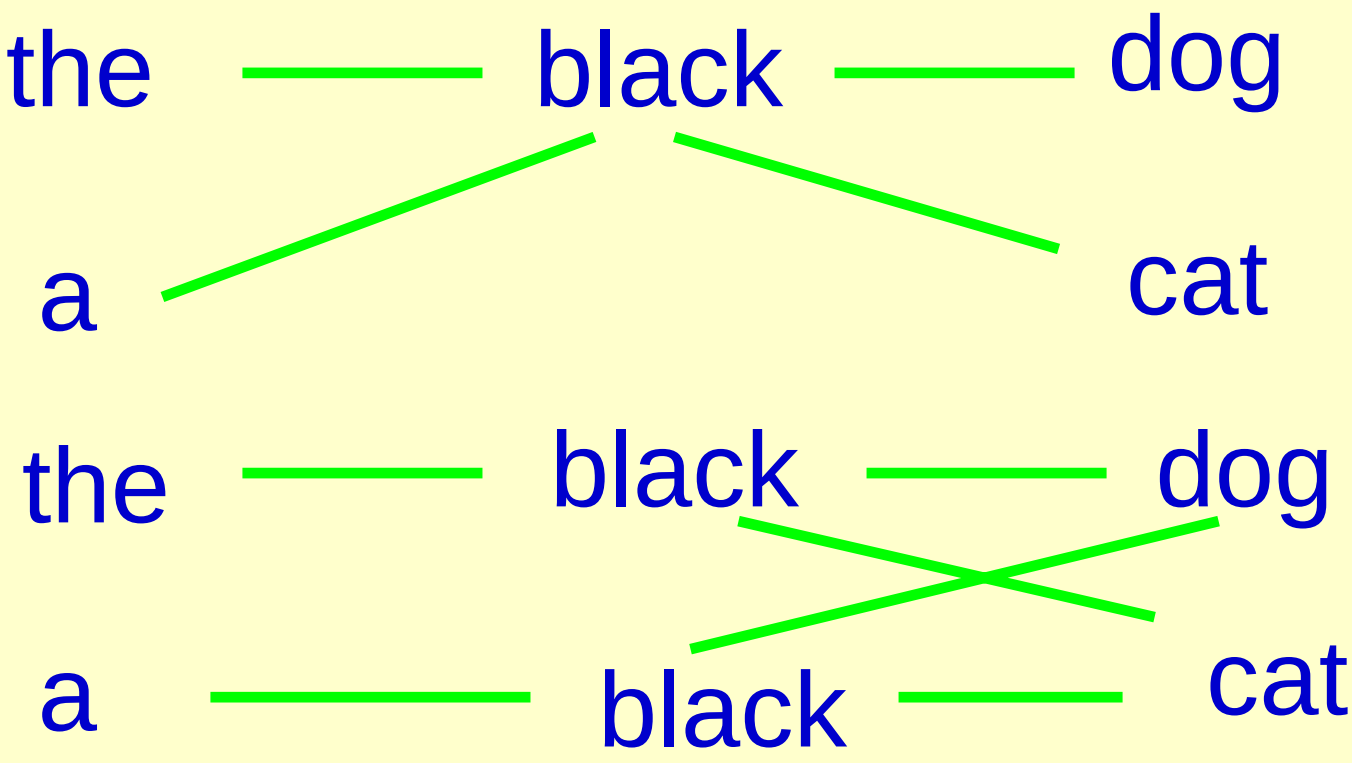
11	s11(t1)
21	s21 (t1)

t1

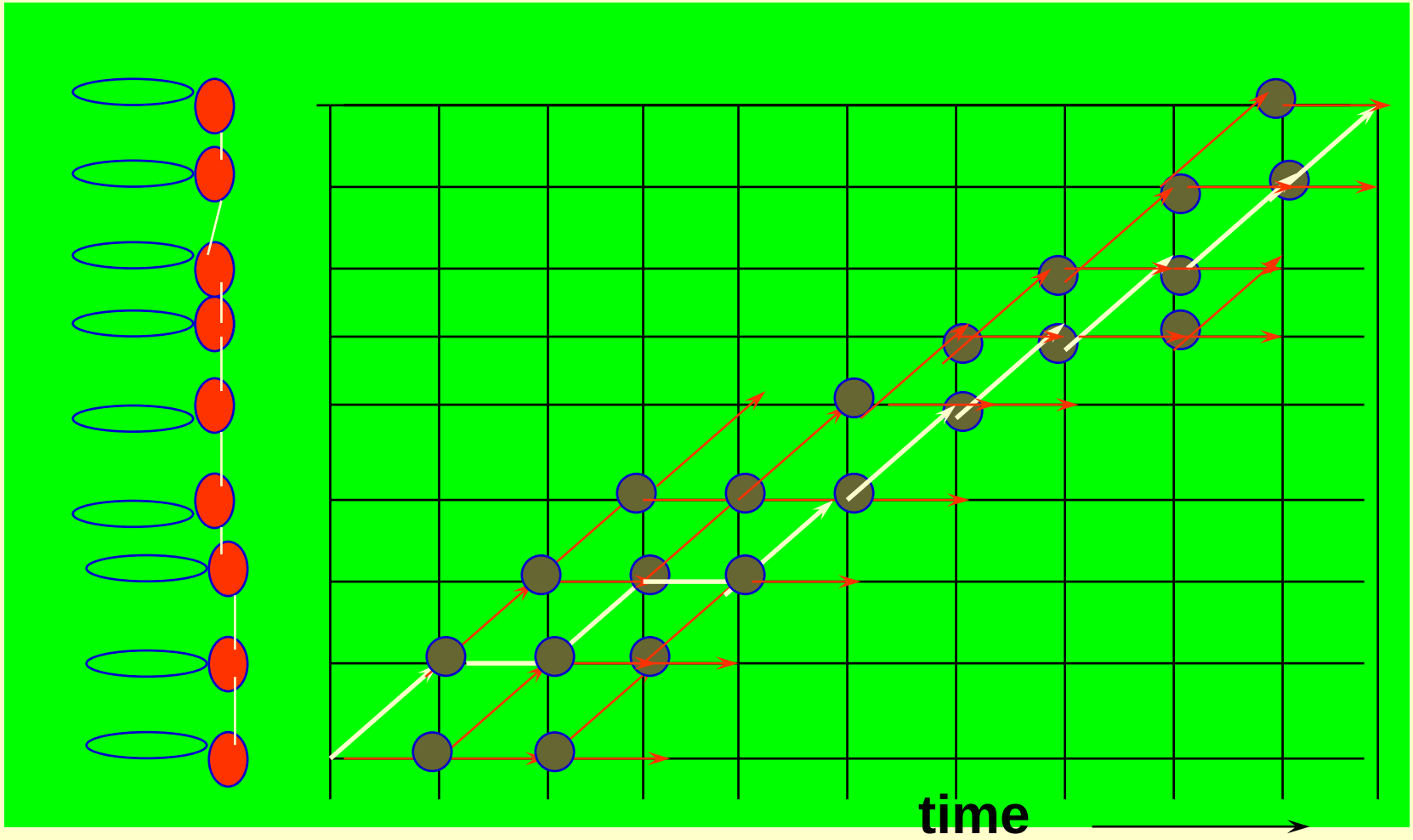
11	s11(t2)
22	s22(t2)
21	s21(t2)
12	s12(t2)

t2

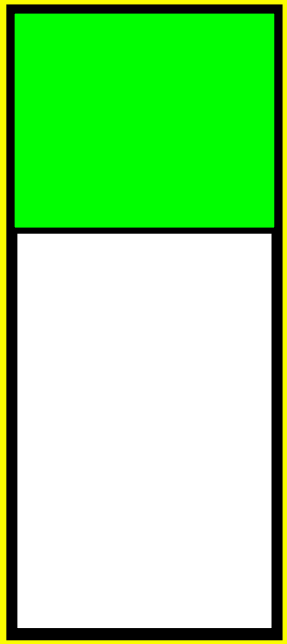
Trigrams require repetitions and more links in the search network



Beam search trellis



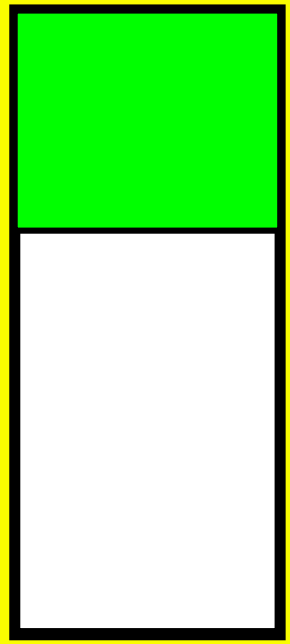
MULTIPLE QUEUE BEAM SEARCH



max

ACTIVE STATES

max -δ



max

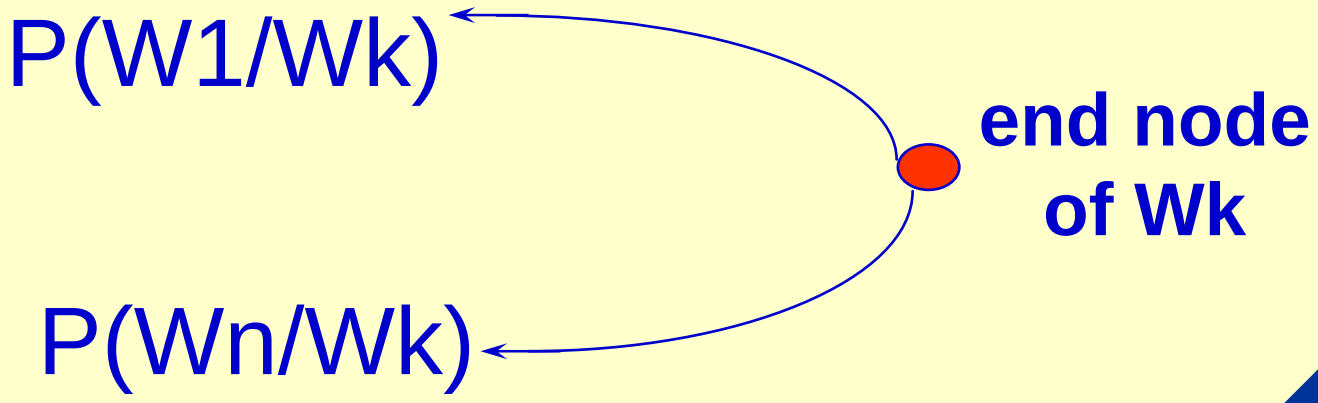
max -δ

internal state queue

final state queue

PROBLEM!

- For every word final state in the active queue there are |W| links to the beginning of each word
- |W| is the vocabulary size



BBN modification of Turing-Good method (more robust)

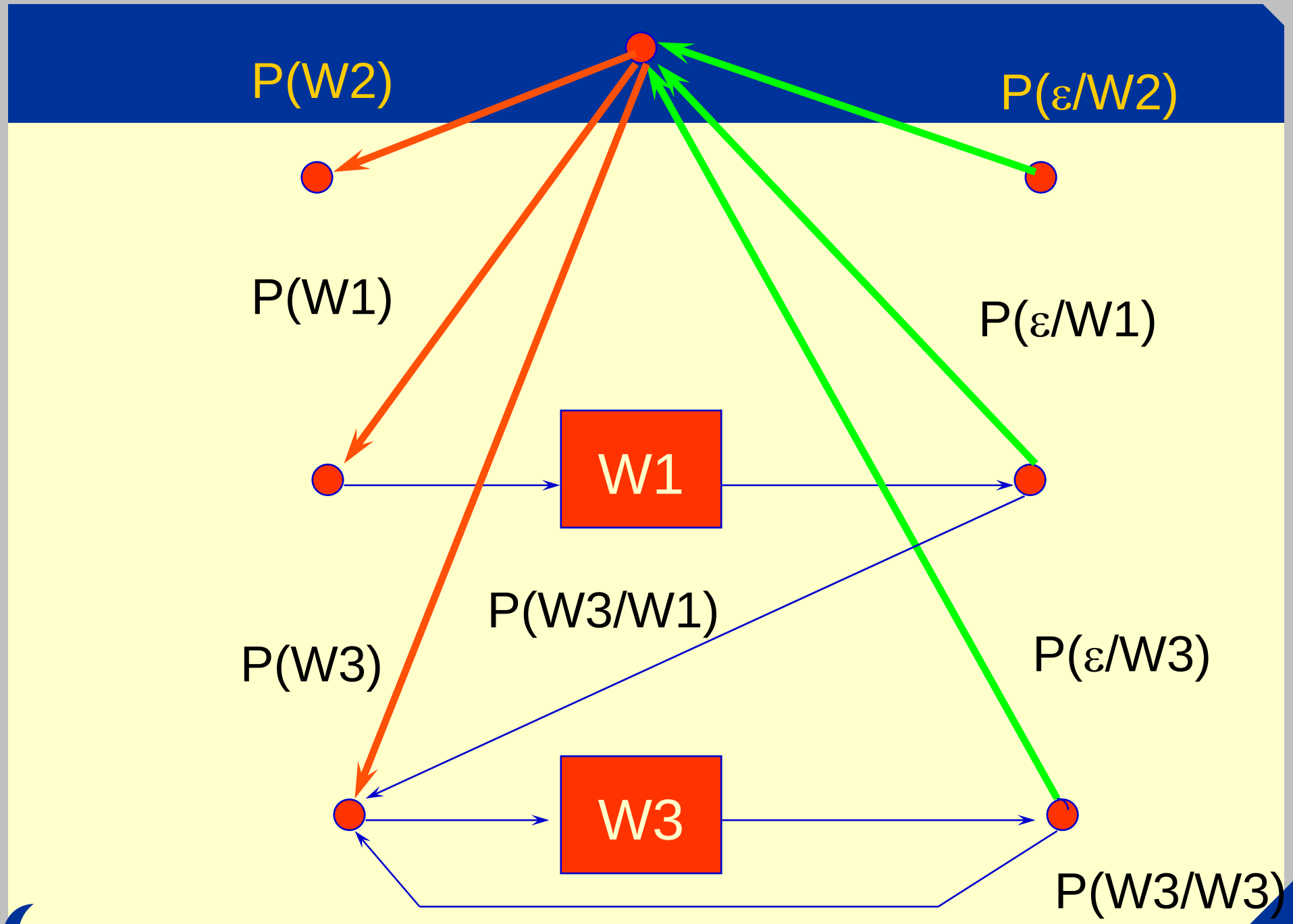
- P(w/x) : bigram prob
- P(x) : unigram prob
- P(ε/x) : prob of a previously unseen word occurring in context x
- n(x) sum of all counters c(w/x)
- r(x) number of unseen words

$$P(\varepsilon / x) = \frac{r(x)}{n(x) + r(x)}$$

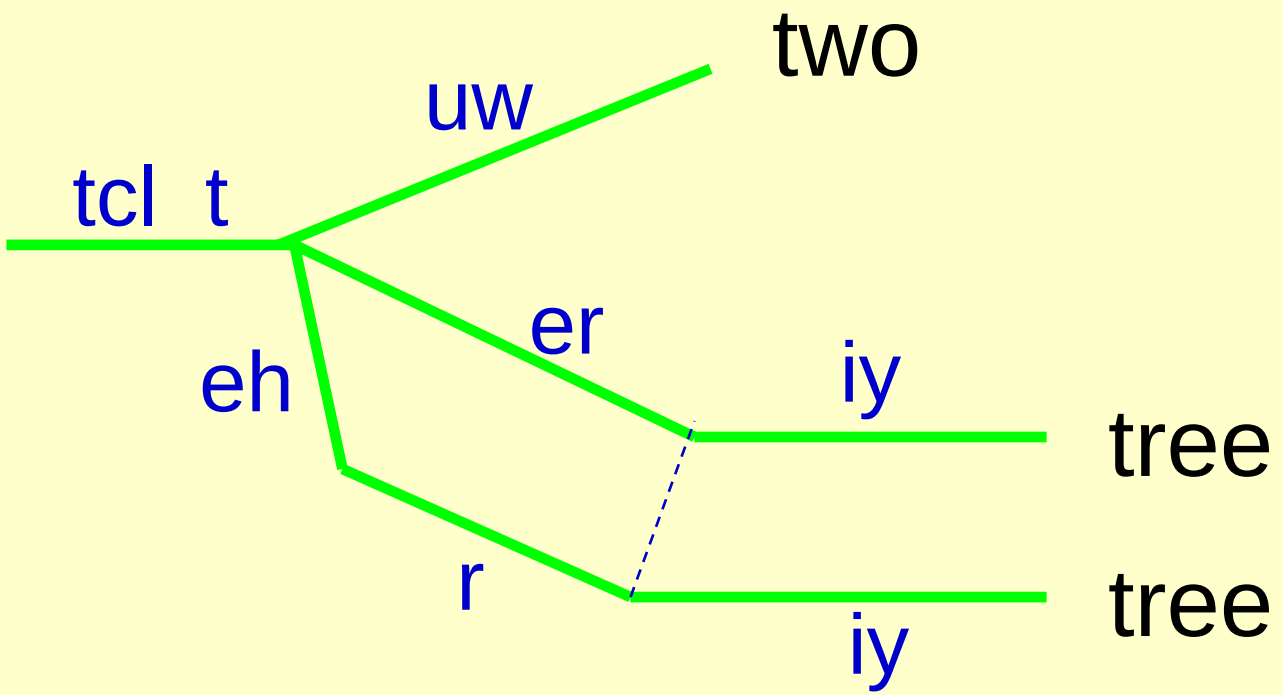
$$P(w / x) = \frac{c(w / x)}{n(x) + r(x)}$$

$$\text{unseen\_words:}$$

$$P(w / x) = P(\varepsilon / x)P(w)$$

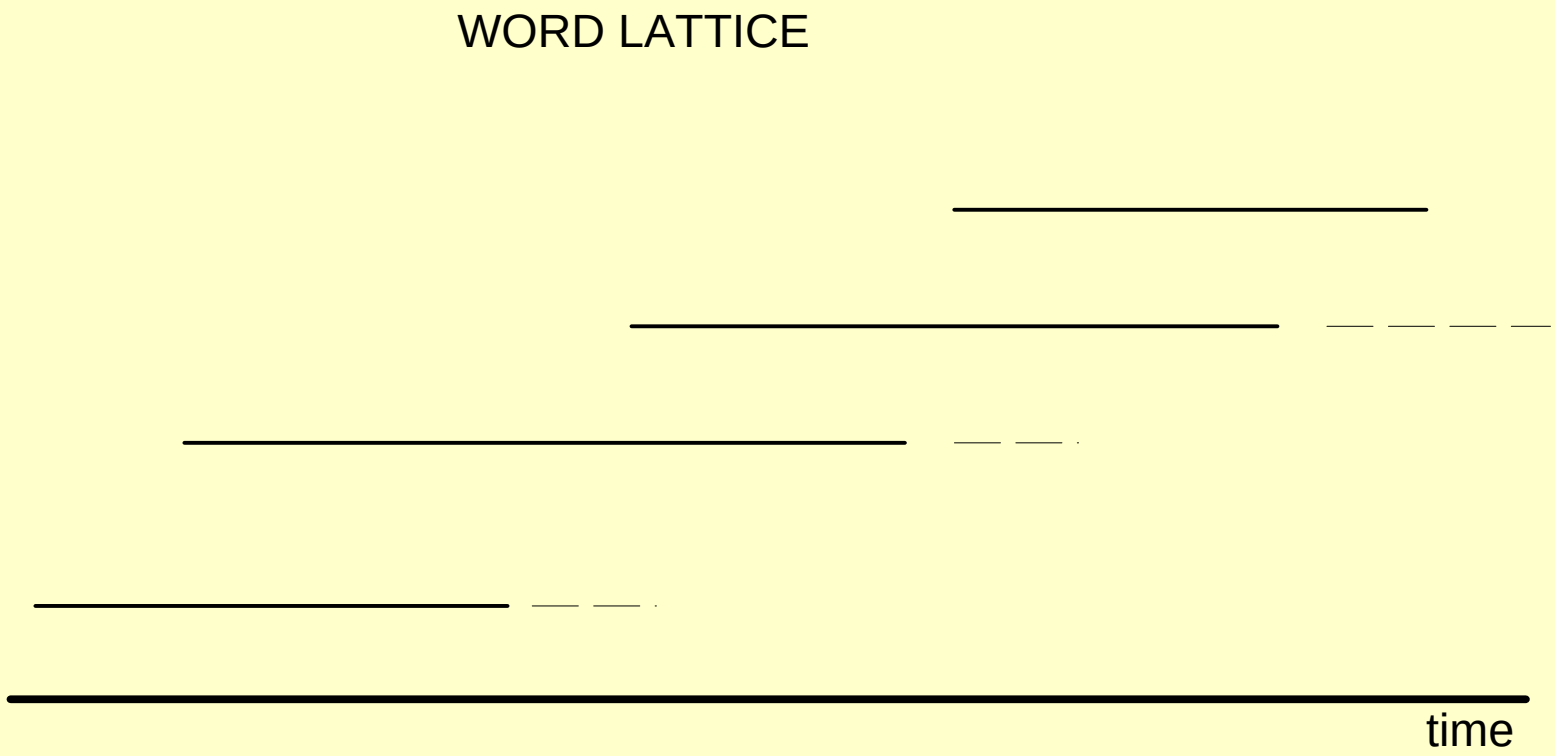

P(W2)
P(ε/W2)
P(W1)
P(ε/W1)
W1
P(W3/W1)
P(W3)
P(ε/W3)
W3
P(W3/W3)

tree lexical representation



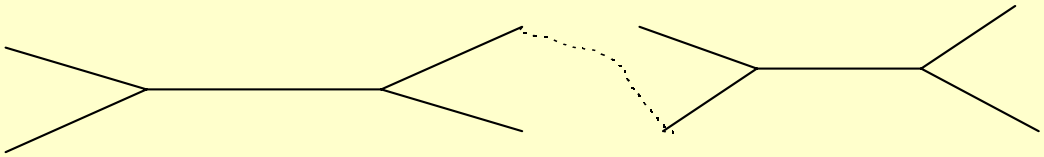
space saving, automaton compression, use of triphones results in a network increase because of differences in the right context

Word lattice



word hypotheses have fuzzy left bound (even with CMU CD models)

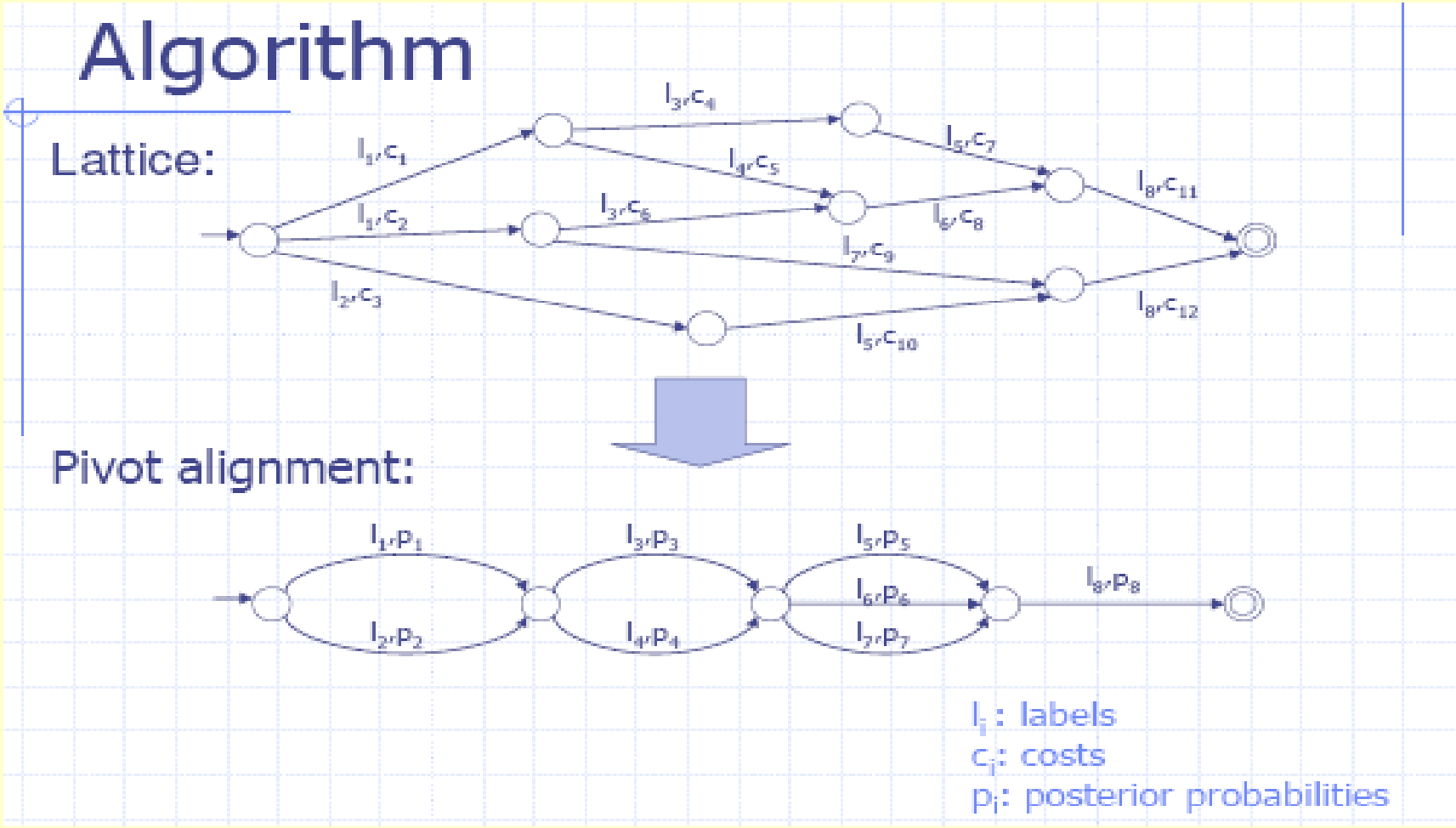
Word boundary effects



For each word, a number of possible beginnings and ends are considered.

They are represented by different connectors linked in a consistent way

Riccardi's pivot



DEMO LUNAVIZ

OK 340

HS 267

History

1941		Dudley patent
1951		Dreyfus-Graf and Smith
1952	first demo	Davis et al.,
1962		Sakai and Doshita
1968		Slustker & Vintsjuk
1969		Vicens & Reddy
1970	De Mori et al, Pools, Bridle, Sakoe &Chiba	
mid 70s		DARPA PROJECT

History

1975 **IBM system (Bahl, Jelinek)**

1975 **Dragon system (Baker)**

1975 **Beam search (Lowerre)**

1980 **IBM dictation machine**

ATT telephone applications

Artificial Neural Networks reappear

History

1990	**DARPA-NIST**	**dictation**
		ATIS
		switchboard
2000	**broadcast news**	
	speech-to-speech translation	
	spoken document retrieval, indexing	
	spoken language understanding	

Problems

Switchboard experiments have shown that WER is still high (> 30%) for conversational telephone speech

WER is high in presence of noise

WER is high for non-native speakers

Information Retreival using speech starts perfoming much worse than using text if WER goes beyond 25%.

Attempts to improve performance

SPEAKER ADAPTATION

Introduction

• Language and acoustic models are fully specified by a set of parameters.

• If the models provide a correct formulation of the reality and their parameters are known,

then :

the expected minimum recognition error rate is achieved by selecting a hypothesis with the well known Bayes decision rule

Decision rule

A decision rule of the type:

$$\hat{W} = \arg \max_{W} P(A / W) \cdot P(W)$$

Minimizes the expected error count on W

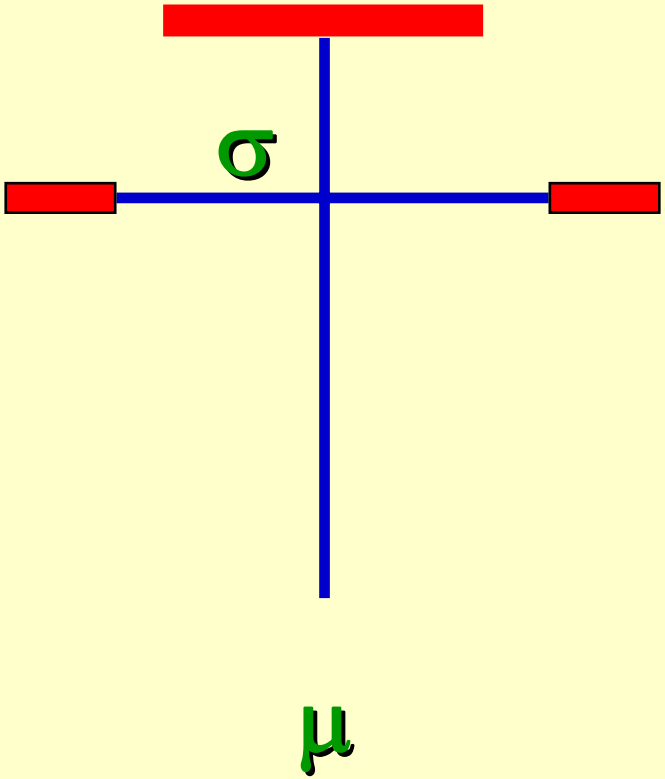
But, in practice, we can only compute:

$$\hat{W} = \arg \max_{W} P_{\Theta}(A / W) \cdot P_{\Gamma}(W)$$

Questions about the rule

- **parametric forms (statistical models) for probabilities are assumed**
- **parameters are estimated from data depending on**
  - **the type and size of the training set,**
  - **training conditions regarding**
    - **microphone,**
    - **channel, environment,**
    - **task-dependent phonetic and linguistic facts**

Model parameter uncertainty



Need for adaptation

- **Mismatch between training and testing conditions**

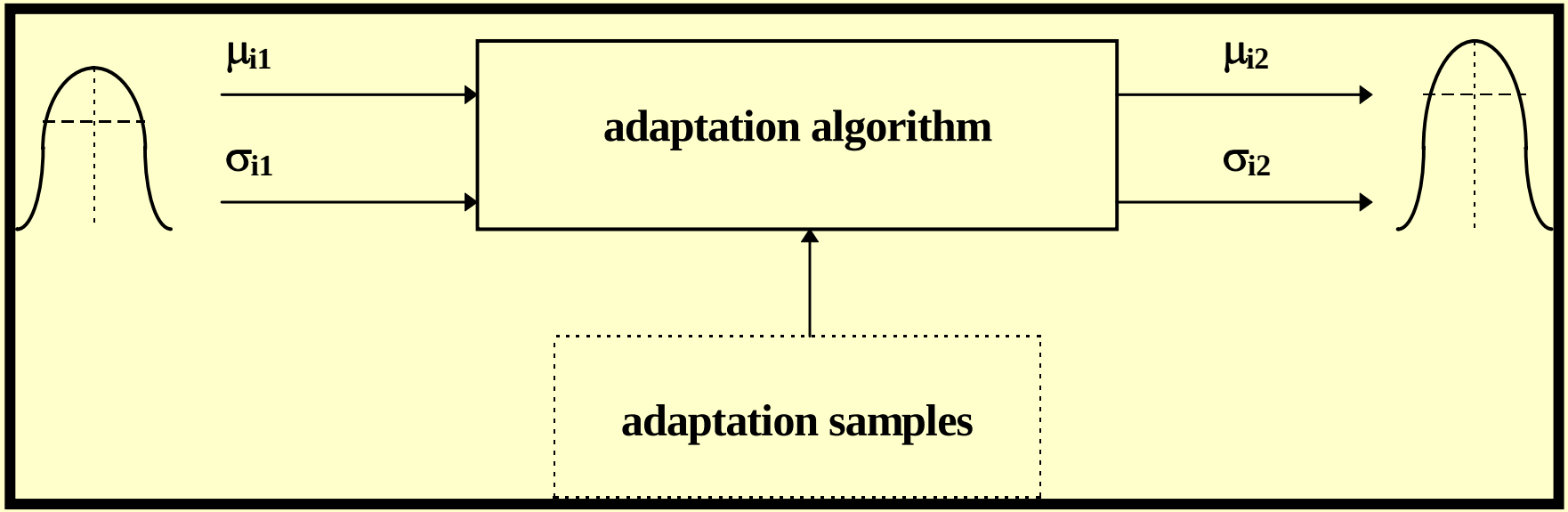
- **mismatch between speakers used for training and testing**

- **Speaker Independent (SI) models do not represent the true distributions of any speaker. These models suffer serious degradation when tested on speaker/environments not represented in training (BBN ARPA 95)**

***It is desirable to continuously adapt to an evolving environment*:**

- **model parameters**
- **feature extraction or normalization**
- **both**

Adaptation process



Noisy channels and condition mismatch

Training conditions

$$y1(t) = h1(t)*s(t)+ n1(t)$$

Testing conditions

$$y2(t) = h2(t)*s(t)+ n2(t)$$

Types of compensation

nonparametric compensation **where the features and models are compensated without any assumptions about the type of distortion,**

parametric compensation **where the structure of the compensation is assumed to have a functional form and the parameters of such a structure are estimated**

stereo data based compensation **where compensation is performed assuming clean and noisy samples of the same data are available**

Adaptation modalities

batch adaptation **is performed with samples of the entire adaptation set**

incremental adaptation **is performed after recognition of each sentence or group of sentences**

Self **or** ***instantaneous adaptation*** **is performed on each sentence before it is recognized (especially useful when there is a very brief interaction between the speaker and the system)**

Furthermore:

in ***supervised adaptation*** **the input to the algorithm includes the sentence transcription.** ***Unsupervised adaptation*** **adapts automatically based on recognizer hypotheses**

Adaptation methods

Non-parametric approaches

minmax method (Merhav and Lee, IEEETSAP 1993) **consists in adjusting model parameters in a restricted neighborhood so that the worst case probability of misclassification is minimized**

HMM inversion method (Moon and Wang, ICASSP95, p.145) **in which the Baum-Welch reestimation formula is used to obtain a better estimation of the features. Without any restriction, this procedure converges to the model means. The two procedures can be used iteratively.**

Adaptation methods

***Parametric methods* use a small set of parameters to describe the compensation structure which can be inferred from the test data.**

(good overview in Woodland IEEE ASRU 1999)

***cepstral mean normalization* and codebook dependent *cepstral normalization*,**

***hierarchical clustering*,**

***affine transformations* *of features* estimated, for example with *Maximum Likelihood Linear Regression (MLLR)*,**

Maximum A Posterior (MAP) probability estimation

and various combinations of them.

Adaptation methods

***Stereo compensation* methods include:**

***signal to noise ratio cepstral normalization*,**

***probabilistic optimal filters*, applied in the feature space, require parallel recording of clean and noisy speech. (Neumeyer et al., 1994, Rahim and Juang, 1996),**

***neural networks*.**

Most methods compensate for non-linear distortions using linear approximations.

MAP adaptation

Just considering acoustic models, the model parameters Θ are random variables themselves with a given probability distribution g(Θ). Training can then be seen as the choice of Θ such that:

$$\hat{\Theta} = \arg \max_{\Theta} g(\Theta) \cdot P(A \mid \Theta)$$

where A is an acoustic description of the training set. This is called Maximum A Posteriori (MAP) estimation. (Lee et al., IEEETSP, 1991)

MAP and MLE

$$\Theta_{ML} = \arg\max_{\Theta} P(A/\Theta)$$

$$\Theta_{MAP} = \arg\max_{\Theta} \{P(A/\Theta)\cdot g(\Theta/\Phi)\}$$

$g(\Theta/\Phi)$ **is a prior distribution of the model parameters possibly with respect to a smaller set** Φ **of *hyperparameters***

$g(\Theta/\Phi)$ **is usually a *conjugate* distribution such that** $g(\Theta/A)$ **belongs to the same family as** $g(\Theta)$**. Under this hypothesis the Expectation Maximization (EM) algorithm is applicable to MAP (Gauvin and Lee, IEEETSAP 1994).**

Problems with MAP

The classical formulation requires an amount and a distribution of data that is rarely available in practice. Approximations of the true posterior probability of Θ given A are considered

Rather that *plugging* into the Bayes decision rule the model parameters estimated with a training procedure, *Bayesian Predictive Classification* (BPC) assumes that the parameter set for the best decision lie in the neighborhoods of the parameters found with training and proposes to adjust the decision rule accordingly.

Affine transformations

A ***speaker independent count*** ***y*** can be computed for each gaussian. Let ***Y*** be the vector of such counts. Given a set of ***speaker dependent counts*** ***r***, it is possible to estimate a transformation matrix ***A(r)*** and a vector ***B(r)*** such that, the vector of the combined counts ***X*** can be computed as follows:

$$X = A(r)Y + B(r)$$

where matrix ***A(r)*** and vector ***B(r)*** can be determined by iterative MLE estimation, for classes of phonemes or , if there are not enough data, they can be the same for all phonemes. Class hierarchies are also considered with thresholds set in such a way that for each pair to be estimated there are sufficient data.

Affine transformations

There are two main forms of model-based transformation namely

***unconstrained* (Leggeter and Woodland, 1995)**

***constrained* (Digalakis et al., 1995).**

Many of these adaptation schemes consist in performing a linear transformation computed using a *Maximum Likelihood Linear Regression (MLLR)* approach (Leggeter and Woodland, 1995)

Eigenvoices

Unsupervised adaptation

Speaker-independent observation densities have the form:

$$P_{SI}(y_t|s_t) = \sum_{i=1}^{N_w} p(w_i|s_t)\,N(y_t; m_{ig}; \Sigma_{ig})$$

where t indicates time, s indicates a state,

y an observation, w the weight of a gaussian in the mixture,

m the mean and Σ the covariance matrix,

g indicates a codebook of gaussians.

Constrained adaptation

The speaker-adapted observation density is constrained by the fact that means and covariance matrix are linearly transformed:

$$P_{SA}(x_t|s_t) = \sum_{i=1}^{N_W} p(w_i|s_t) N(x_t; A_g m_{ig} + b_g; A_g \Sigma_{ig} A_g^T)$$

which corresponds to having the speaker independent observations transformed as follows:

$$x_t = A_g y_t + b_g$$

Speaker normalization

Cepstral mean subtraction is a popular and simple feature-base normalization method.

Nevertheless, is generally agreed that one of the major sources of inter-speaker variability is the vocal tract shape.

One consists in estimating vocal tract parameters (especially related to vocal tract length) and use them for normalization.

LANGUAGE MODEL ADAPTATION

methods for adapting LMs to a new domain

train a new LM if sufficient data are available

- pooling data of general model and new domain
- linear interpolation (see section on linear interpolation)
- back-off with general model
- retrieve documents and build new LM on-line
- MAP
- Minimum Discrimination Information
- log-linear interpolation

fudge factor adaptation

Domain sub-languages

Even a simple bigram LM may require large amounts of training text samples.

In case of domain change, adaptation techniques allow to reduce the amount of training material by exploiting samples of possible close domains.

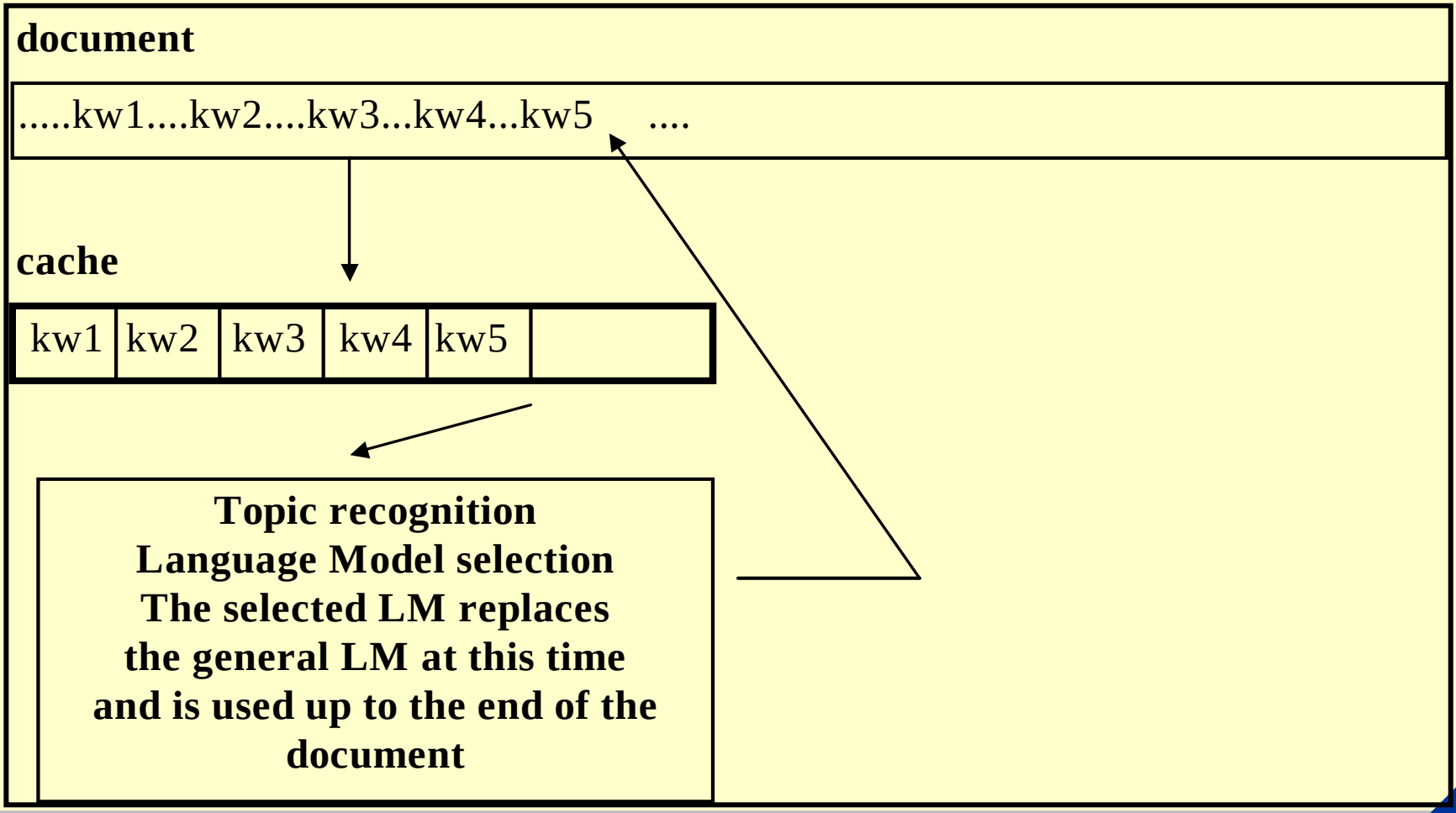
Within a domain sub-language, variations in the language can be due to intra-user differences or topic shifts. Both phenomena can significantly affect the a-priori probability of the word sequences that can be uttered.

Topic shift

Topic shifts also affect the probability of words. For instance, inside an x-ray report words like hearth and lungs are much more likely to occur after the word chest than leg.

User changes and topic shifts can be coped with adaptation that either try to capture long-distance dependencies or to adjust the n-gram statistics.

The switching Language Model



Adaptation

During usage of the system new texts are produced which reflect the user's language. These texts can be used in a **supervised** adaptation

Adaptation can be performed **incrementally**, that is the LM is adapted after every entered sentence, or in batch mode, i.e... after a significant after a suitable chunk of texts has become available.

Types of adaptation

In general, topic adaptation requires incremental adaptation, as short term language changes have to be modeled.

Domain adaptation intrinsically needs more data, hence batch adaptation is preferable.

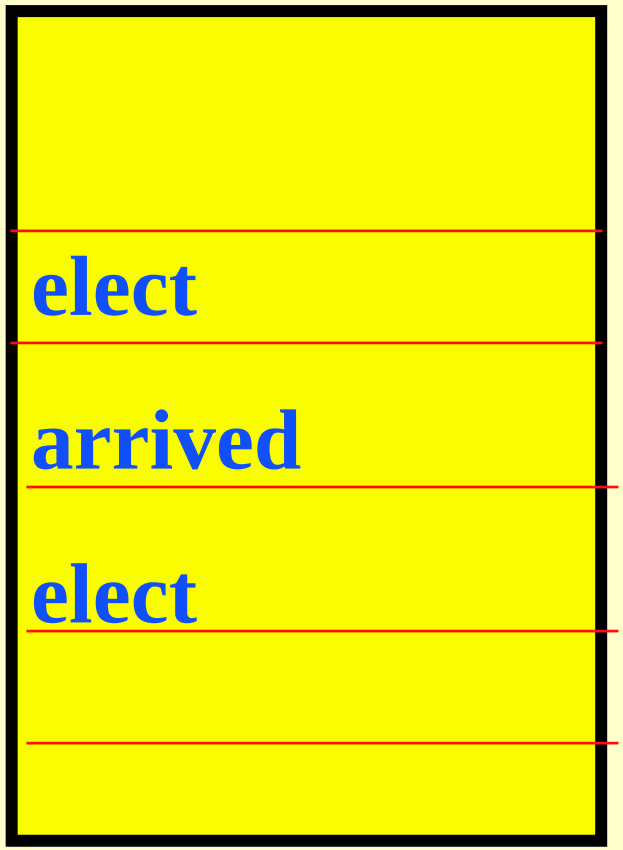
User adaptation can be performed in both modes if for instance an a priori sample of the user (see radiological reporting application) or incrementally.

Cache models

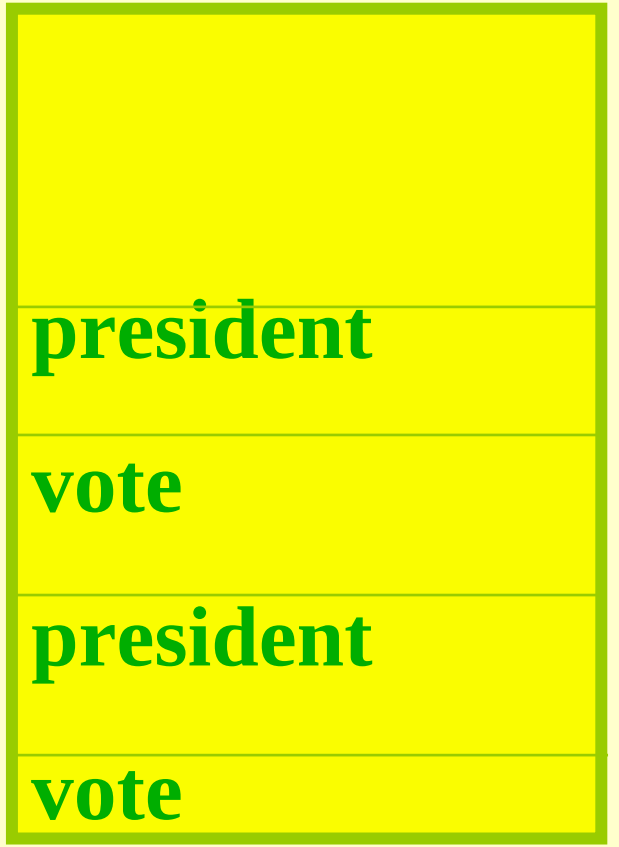
The first adaptable statistical LM was based on the simple hypothesis that a word used in the recent past is much more likely to be used soon than its overall frequency in an n-gram LM would suggest.

Modeling long-distance dependencies has been the first adaptation paradigm.

class cache model



verbs



nouns

POS cache (Kuhn and De Mori 88-90)

$$P(w_j / h_j) =$$

$$P(g(w_j) / g(w_{j-1}) g(w_{j-2})) \cdot$$

$$\left\{ \begin{array}{l} \lambda P_{static}(w_j / g(w_j)) + \\ (1-\lambda) P_{cache(g(w_j))}(w_j) \end{array} \right\}$$

Bigram and trigram cache

$$P(w_j / h_j) =$$

$$\lambda P_{static}(w_j / w_{j-1} w_{j-2}) +$$

$$(1 - \lambda) \cdot$$

$$\left\{ \begin{array}{l} \alpha_1 f_N(w_j) + \\ \alpha_2 f_N(w_j / w_{j-1}) + \\ \alpha_3 f_N(w_j / w_{j-1} w_{j-2}) \end{array} \right\}$$

Interpolated estimation

$$P(w_i / (w_1^{i-1}) = \sum_j \lambda_j (w_1^{i-1}) P_j (w_i / w_1^{i-1})$$

Adaptation consists in making one or more probability distribution or the weights varying with time (Kneser and Steinbiss, 1993, Ney ASI97, Iyer and Ostendorf, 1999).

Maximum likelihood

$$\vartheta^{ML} = \arg\max_{\vartheta \in \Theta} P(SW_1^n, \vartheta)$$

$$\frac{\partial L}{\partial \vartheta_w} = \frac{c(w)}{\vartheta_w} - \lambda = 0$$

$$\sum_{w \in V} c(w) = \lambda \bullet \sum_{w \in V} \vartheta_w = \lambda$$

$$\vartheta_w^{ML} = \frac{c(w)}{\lambda} = \frac{c(w)}{\sum_{w \in V} c(w)} = \boxed{\frac{c(w)}{n}}$$

MAP Adaptation

One model (Federico, 1996)

$$\vartheta_w^{MAP} = \frac{c(w)+c'(w)}{n+n'} = \frac{n}{n+n'}\frac{c(w)}{n} + \frac{n'}{n+n'}\frac{c'(w)}{n'}$$

Triggers boosting

$$f_{AA \rightarrow BB}(h, w) = \begin{cases} 1 & if \left\{ (\exists x, x \in AA, x \in h) \wedge w \in BB \right\} \\ 0 & otherwise \end{cases}$$

Trigger pairs are obtained by computing the mutual information between a word and words in the same sentence or in the dialog history :

$$I(X,Y) = \frac{P(X,Y)}{P(X)P(Y)}$$

Constrained adaptation

Constraint-based models introduce soft partitions in the word space.

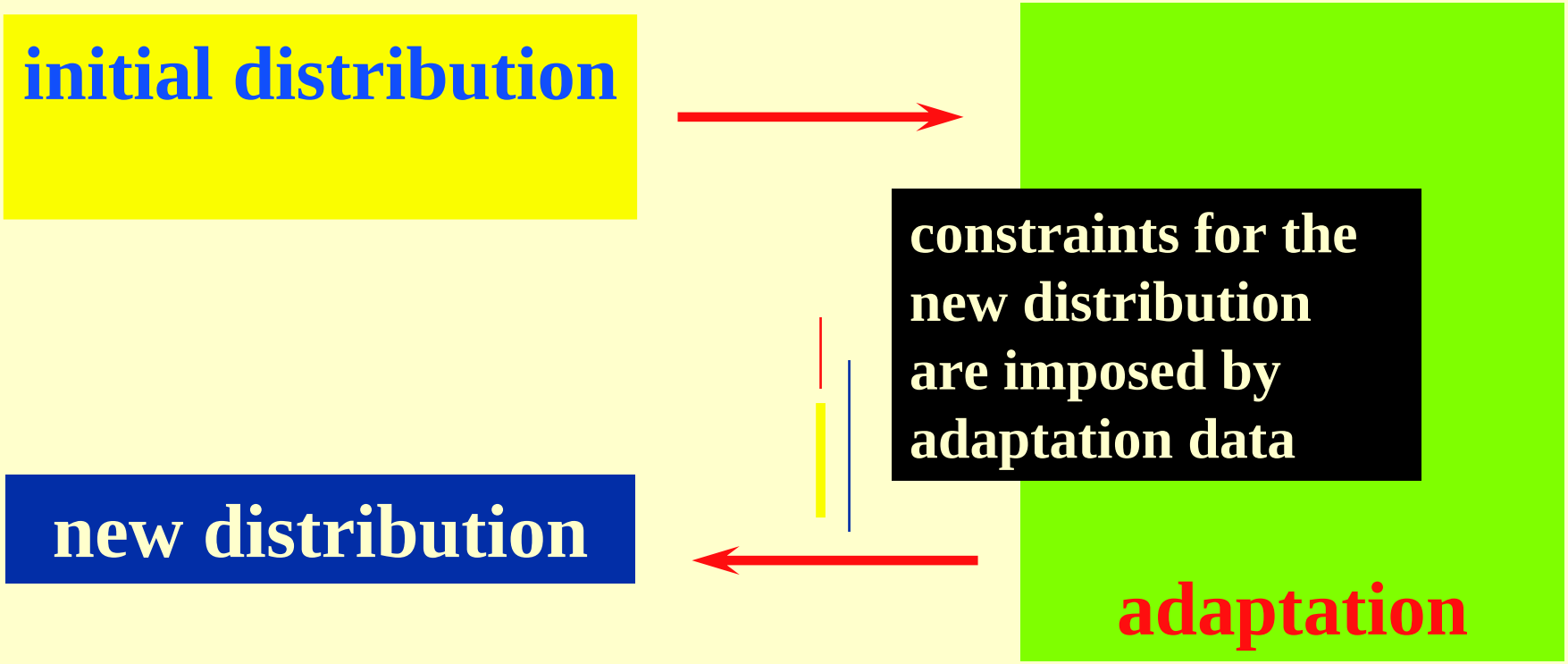
If the constraints are consistent, there exists a unique solution in the exponential family which satisfies them.

Among all solutions including those of other families, the exponential one is the closest to the prior one in the Kullback-Leibler sense or Minimum Divergence or Minimum Discrimination Information. $$D[P(x), Q(x)] = \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)}$$

If the prior is flat, then this becomes the Maximum Entropy solution.

The parameter space is concave (suitable for iterative solutions)

Constrained adaptation



General solution

$$P(w/h) = \frac{Q(w/h)}{Z(h)} e^{\sum_i \lambda_i f_i(w,h)}$$

Constraints are of the type:

$$f_S(h,w) = \begin{cases} 1 & \text{if} \quad (h,w) \in S \\ 0 & \text{otherwise} \end{cases}$$

$$\sum_{h,w} \left[P(h,w) f_S(h,w) \right] = C_S$$

$$\sum_h \tilde{P}(h) \sum_w \left[P(w/h) f_S(h,w) \right] = C_S$$

Just-in-time language modeling

This definition comes from (Berger and Miller, 1998) who propose to perform LM parameter estimation and adaptation at the same time. In processing a single utterance, a system uses its non-stop words to perform a query to a collection of words or to the WWW. Based on the retrieved documents, a new LM is derived and used to adapt a static one.

The concept is inspired by topic coherence models are proposed in (Sekine and Grishman, 1995). Here keywords of previously recognized sentences are used for retrieving pertinent documents from which an LM is derived and updated dynamically.

Log-linear interpolation

In (Klakow, 1998), the following model is considered as composed by J topic models:

$$P(w|h) = \frac{1}{Z(h)} \prod_{j=1}^{J} P_j(w|h)^{\lambda_j}$$

The simplex method can be used for finding the exponents

It can be shown that this is equivalent to impose that the model to be found has minimum distance w.r.t. the uniform distribution and satisfies the constraints that the new model has predetermined KL distances from each of the models. This approach has less free parameters to estimate w.r.t. ME.

Fast marginal adaptation

$$P(w \mid h) = \frac{1}{Z(h)} \left\{ \frac{P_a(w)}{P(w)} \right\}^{\beta} Q(w \mid h)$$

(Kneser et al., 1997)

a makes reference to the adaptation data

Combination of clustering and constraint-based re-estimation of probabilities for adaptation is proposed in (Kneser and Peters, 1997) where also evidence is provided that exponential models are superior to linear interpolation in combining multiple information sources.

BEYOND CLASSICAL N-GRAMS

Latent semantic analysis

The use of word probabilities depenent on Latent Semantic Analysis is proposed in (Bellagarda, 1997).

The starting point is a {|W|x|D|} matrix A of word (in a vocabulary W) probabilities in each document of a collection D.

This matrix is decomposed using Singular Value Decomposition (SVD) into the product of smaller matrices:

$$A = U \; x \; S \; x \; V^{T}$$

where S(RxR) matrices is square with R=125, U (|W|xR}, V (Rx|D|).

Adaptation in reduced space

Consider a set of vectors in reduced space with sufficient adaptation counts counts:

$$\mathbf{c}_i^r = \mathbf{U}^T \mathbf{c}_i \quad i \in \Phi_a$$

And the vectors of the corresponding words before adaptation:

$$\mathbf{m}_i^r = \mathbf{U}^T \mathbf{m}_i \quad i \in \Phi_a$$

Classes of sematically similar words can have an affine transformation of the form:

$$\mathbf{A}^* = \arg\min_{\mathbf{A}} \left\{ \sum_{i \in \Phi_a} \left(\mathbf{A}\mathbf{m}_i^r - \mathbf{c}_i^r \right) \sum_{i \in \Phi_a} \left(\mathbf{A}\mathbf{m}_i^r - \mathbf{c}_i^r \right)^T \right\}$$

Nondeterministic stochastic automata

Variable N-gram Stochastic Automata (VNSA) with empty transitions compute the following state dependent probability :

$$P(w_1,.....,w_j,.....w_N \mid s_k) = \prod_{j=1}^{N} P(w_j \mid w_1,.....,w_{j-1};s_k)$$

Essentially partial models recognizing chunks can be placed in series or parallel thus allowing combining class based models, back-off models and variable n-gram models.

Chunks are obtained by segmenting the corpus in such a way that minimum entropy is found.

A transition correspond to a word, thus adaptation is a linear combination of transition probabilities of a general and the adaptation data. (Riccardi and Gorin, 2000).

Recent tendencies and problems

Integration of multiple LMs and multiple features

Use context for dynamically adding OOVs

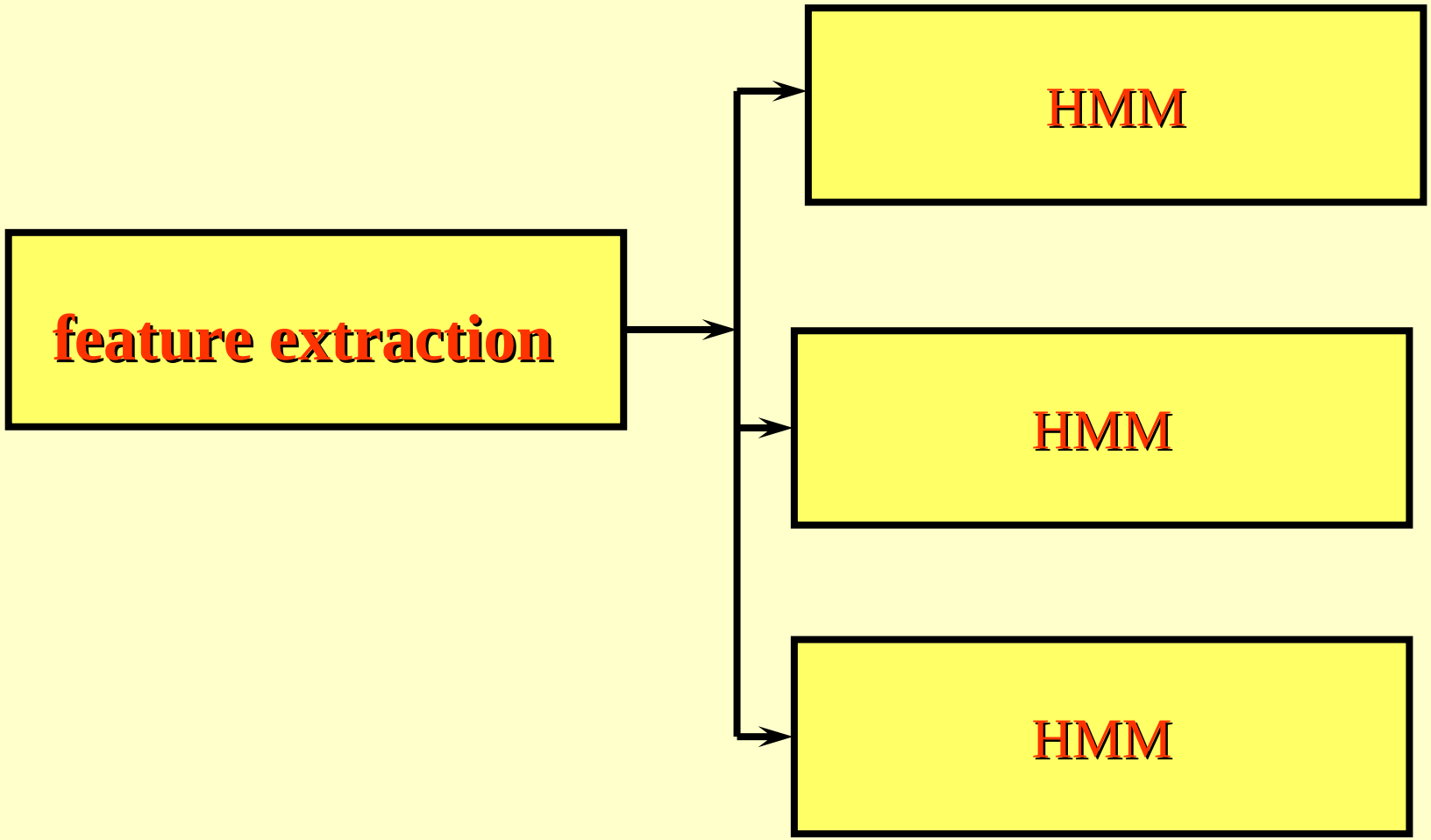
Discriminative LM training

Rescoring

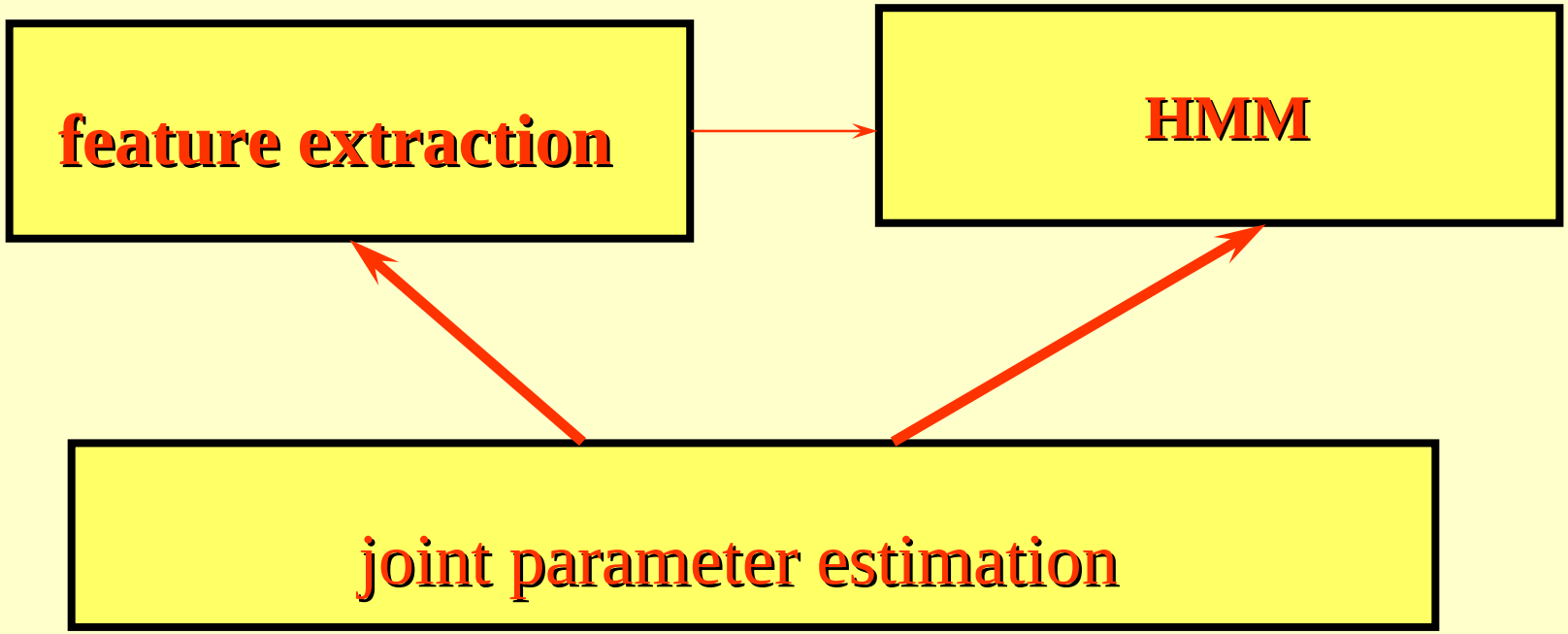
combination

- Speaker clustering and eigenvoices
- Speaker adaptive training
- Non-linear tranformations based on neural networks
- data augmentation
- adaptation to speaker speed

Mixture HMM



Joint estimation



Multiple recognizers

IBM proposes an architecture (Kingsbury, 2002) with multiple transformations and multiple classifiers (Fine et al., 2002).

Different features are used:

Full band non-compressed root cepstral coefficients (RCC)

Full band PLP 16kHzTelephone band PLP 8 kHz

Training multiple models

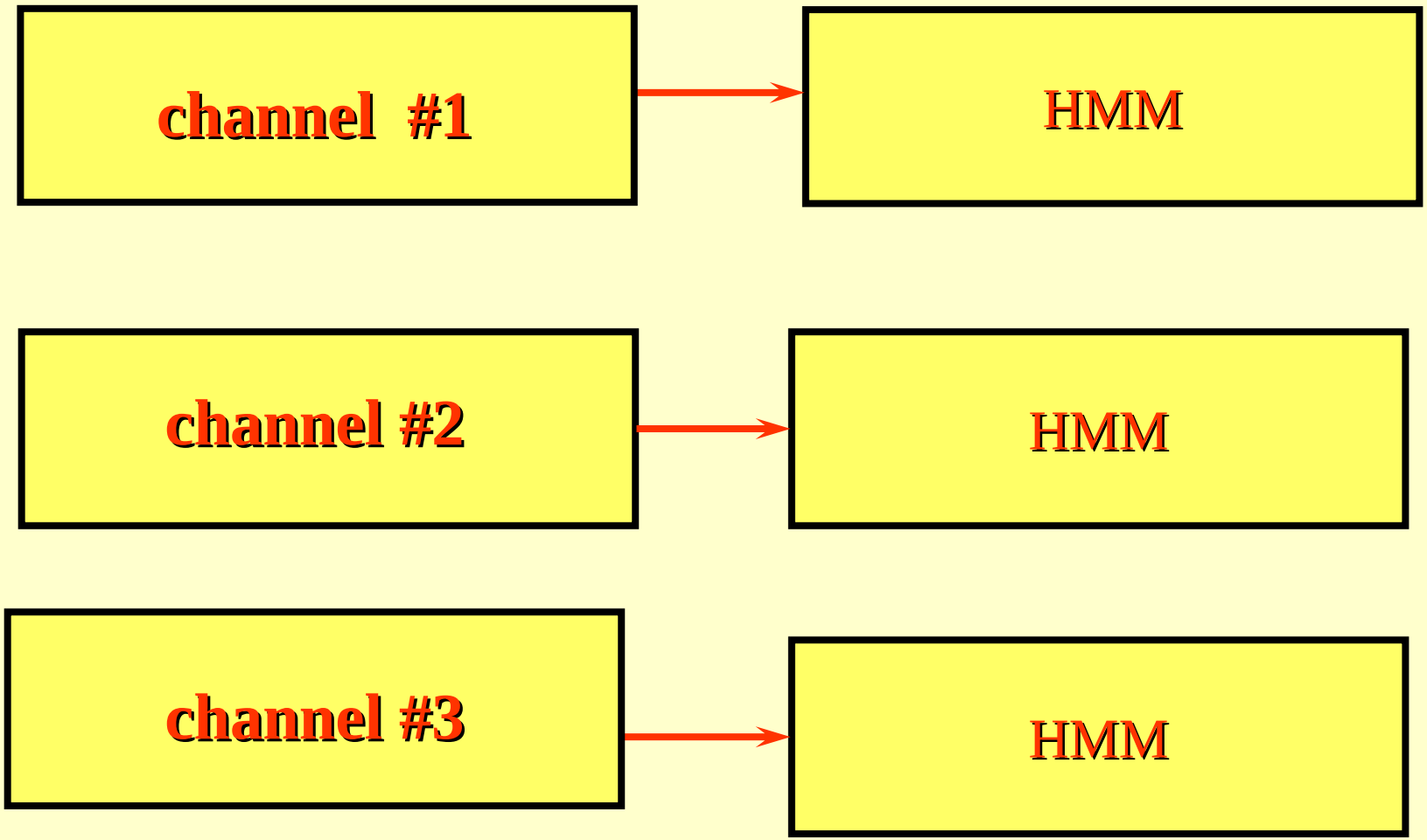
The transformations for training acoustic models (Transformation based learning) are:

VTLN . For each training speaker a VTL warp factor is selected among 21 possibilities +-20% linear warping. The canonical model after VTLN is the VTLN model.

LDA+MLLT (LDA followed by Maximum Likelihood Linear transformation) on each of the three set of features is applied to the VLT warped spectra.

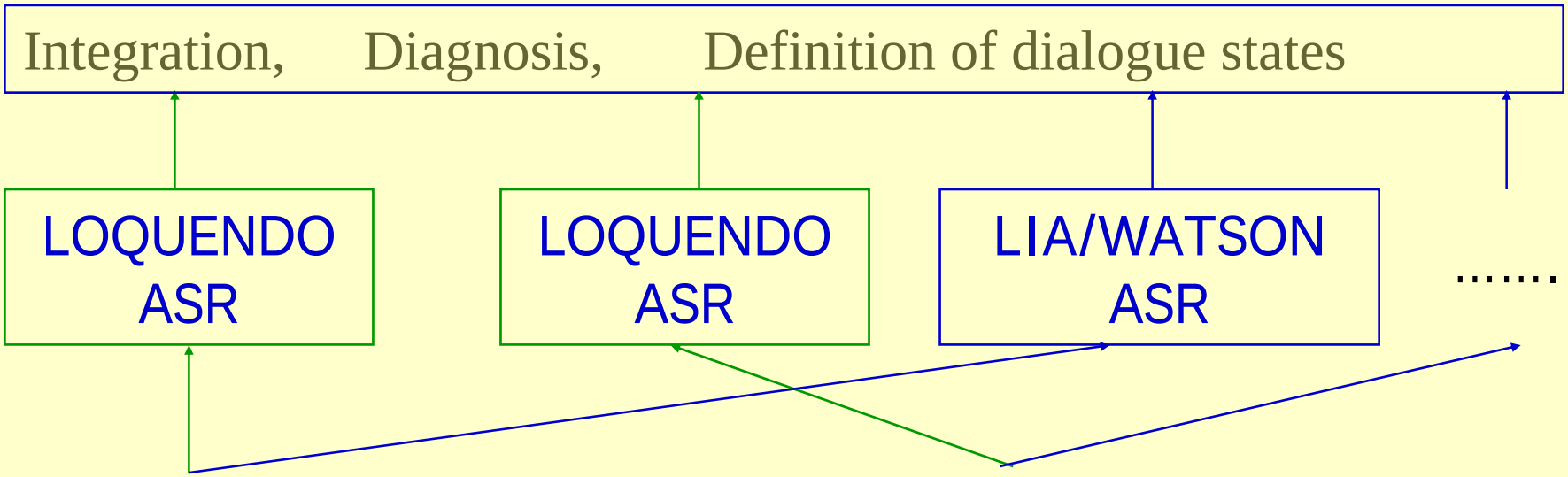
A single affine transformation is computed after VLTN LDA+MLLT for each speaker such that the likelihood of the transformed features is maximizer w.r.t. the canonical (VLTN) model.The canonical model is then re-estimated

Multi-channel systems

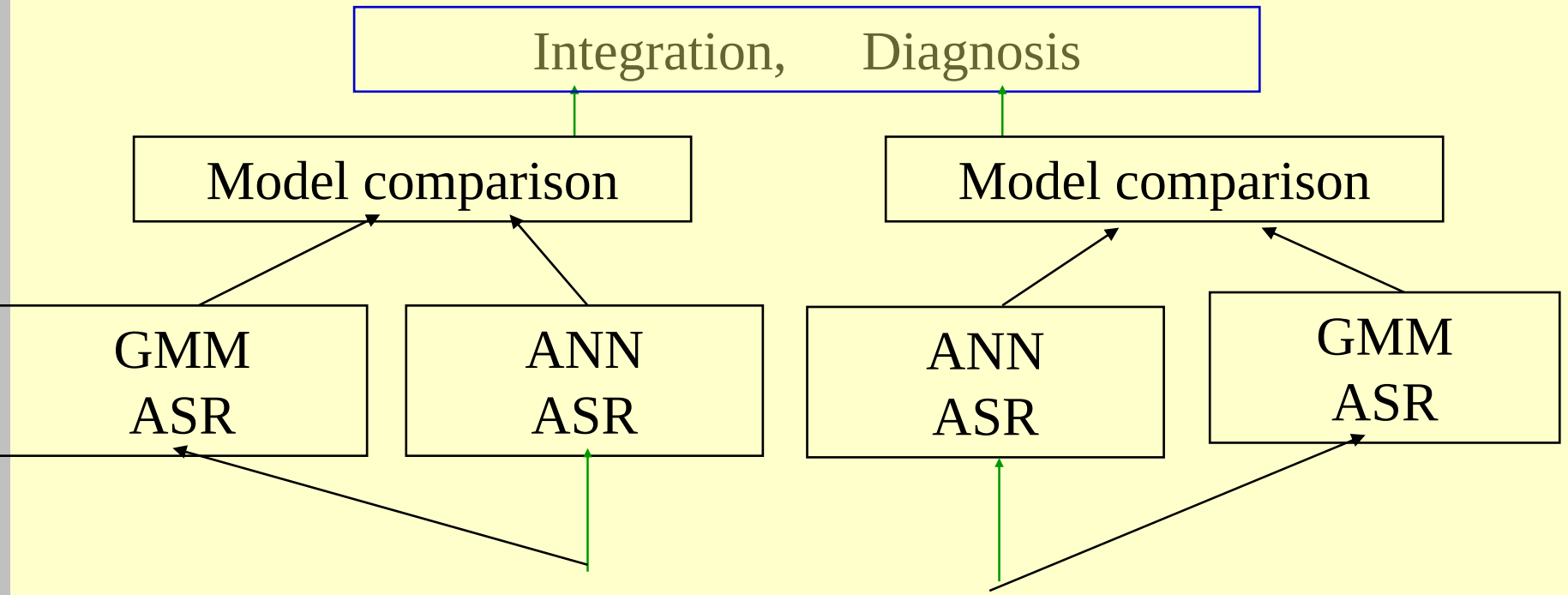


Use of different recognizers

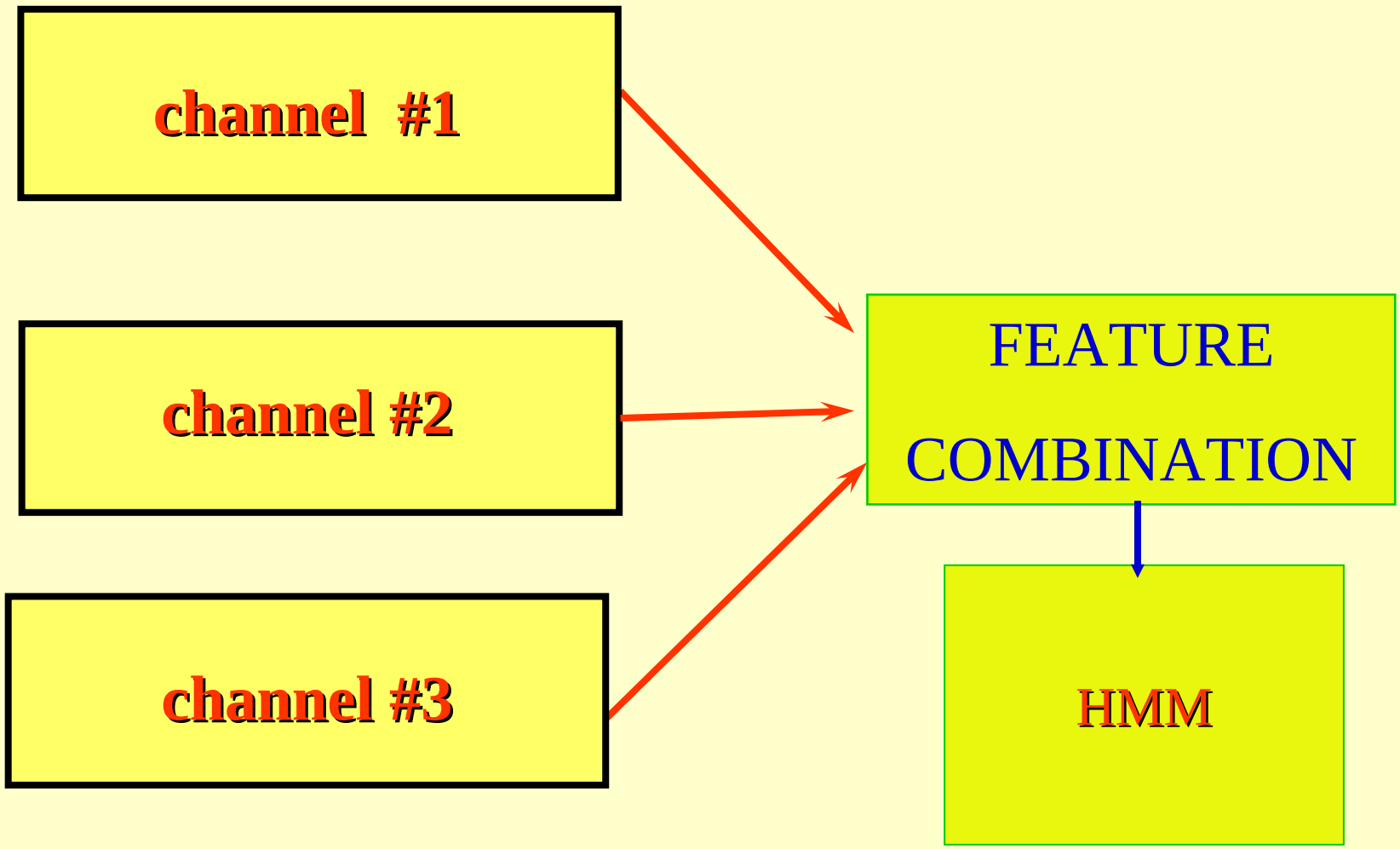
Performance analysis with multiple recognizers (Gemello, Mana, De Mori)



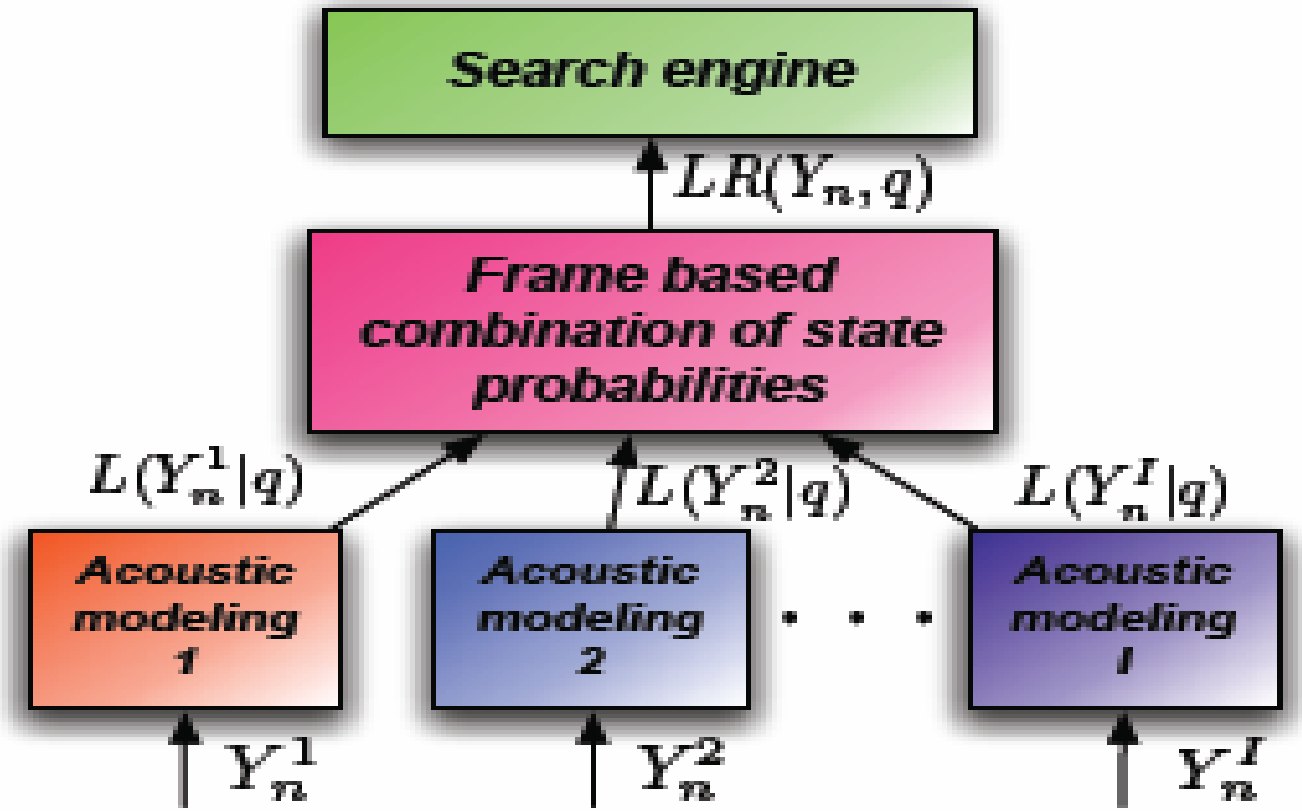
Use of different recognizers



Multiple features



Multiple features twin model



Icassp 2008

Log-prob

Just sum log probs of likelihood ratios for each frame and state to obtain observation log prob

$$\mathrm{LP}\left\{P(Y_n|q)\right\} = \sum_{i=1}^{I} \log\left\{\frac{L\left(Y_n^i|q\right)}{\sum_{g\in Q} L\left(Y_n^i|g\right)}\right\}$$

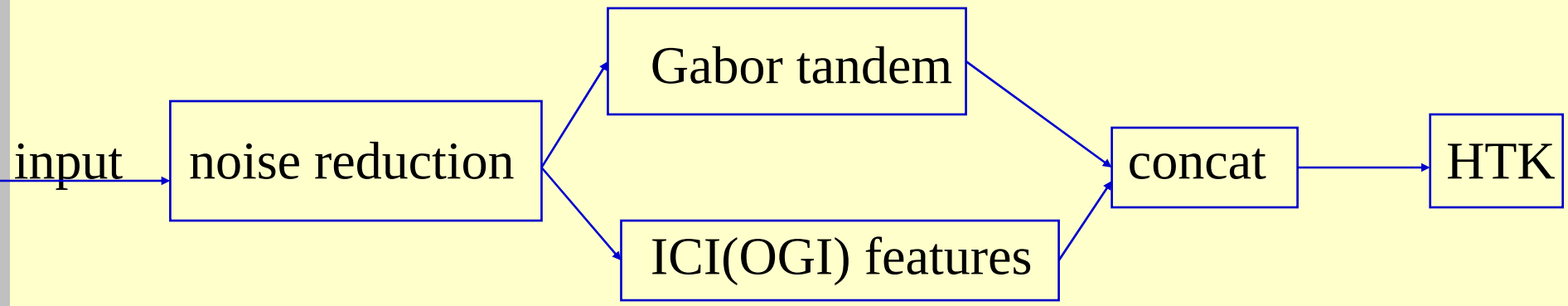
Results telephone speech

MEDIA test (1377 sentences and 10434 words) log-linear combination

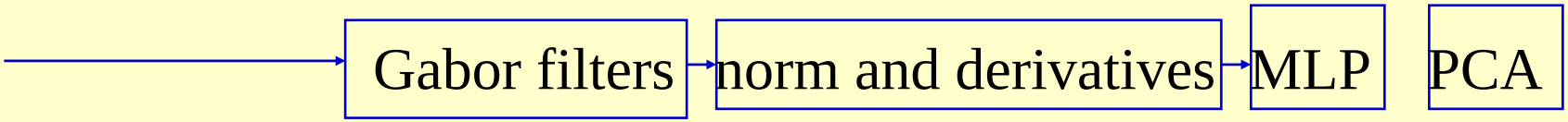
Feature set	WER (%)	Conf int (%)
MRA	33.1	0.90
RPLP	33.1	0.90
PLP	32	0.89
LLC MRA+ RPLP	28.7	0.87
LLC RPLP + PLP	27.4	0.86
LLC MRA+PLP	27.2	0.85
LLC MRA+RPLP+PLP	26.4	0.85
oracle	24.2	0.82

FEATURE COMBINATION

Icsi/OGI



Gabor tandem



DIMENSIONALITY REDUCTION

In order to reduce dimensionality and increase robustness, PCA and LDA have been performed on the whole set of tree featuresat each time frame with a 10 ms frame rate.

PCA has been performed by transforming the 63 features in order to have zero mean and unit variance. Then the covariance matrix C has been obtained. High correlations have been observed between nodes and their fathers in accordance with the theory.

FEATURE TRANSFORMATION

Linear transformations are reviewed in (Visweswariah et al , 2002), where it is proposed to express a matrix for feature or mean transformation as a combination of a certain number of bases:

$$A = A_0 + \sum_{i=1}^{D} d_i A_i$$

The bases can be obtained from training data by finding the first eigenvectors of a collection of vectors representing the elements of speaker matrices (one per speaker), while the coefficients are found with MLE from test data.

FEATURE TRANSFORMATION

A Maximum Likelihood Linear Transformation (MLLT) is a modeling technique which places some constraints on the gaussian models (IBM 2001). Multiple Linear transform (MLT) is proposed which allows each gaussian to have its own diagonalizing transform.

This transformation in feature space is called FMLLR .

Feature space normalization based on cumulative distributions is proposed in (Sioan and Huerta, 2002) in which a sample in the test is forced to be equal to the sample of the same rank in the train set. This transformation has to be followed by a decorrelating transformation such as MLLT. It gives same performance as the use of FMLLR.

FEATURE TRANSFORMATION

In (Deng et al., 2002) it is shown how MAP and BPC inspire a formulation and use of feature uncertainty.

$$\hat{W} = \arg\max_{W} P_{\Theta}(A/W) \cdot P_{\Gamma}(W)$$

$$\hat{W} = \arg\max_{W} \left[\int p_{\Theta}(A \mid W) p(\Theta \mid W) d\Theta \right] P(W)$$

$$\hat{W} = \arg\max_{W} \left[\int p(A \mid W) p(A \mid \vartheta) d\vartheta \right] P(W)$$

A further assumption is made, that, for each frame, vector **a(t)** has a probability distribution $p(\mathbf{a(t)}|\theta)$ is gaussian with mean $\mu(t)$ and variance $\Sigma(t)$. Re-estimation formulae are provided for these parameters, once noise has been estimated. The use of fixed HMMs is now made by using $\mu(t)$ as observation and by adding th the variance of each model gaussian $\Sigma(t)$.

Denoising

Noisy channels and condition mismatch

Training conditions

$$y1(t) = h1(t)*s(t)+ n1(t)$$

Testing conditions

$$y2(t) = h2(t)*s(t)+ n2(t)$$

Only additive noise

$$Y(t,f) = S(t,f) + Z(t,f) = Y_R(t,f) + jY_i(t,f) =$$
$$S_R(t,f) + jS_i(t,f) + Z_R(t,f) + jZ_i(t,f)$$
$$\left|Y(t,f)\right|^2 = \left|S_R(t,f) + Z_R(t,f)\right|^2 + \left|S_i(t,f) + Z_i(t,f)\right|^2$$

Gaussian model

$$p(S_R) = \frac{1}{\sqrt{\pi\sigma_s}} e^{-\frac{S_R^2}{\sigma_s^2}}$$

$$p(S_I) = \frac{1}{\sqrt{\pi\sigma_s}} e^{-\frac{S_I^2}{\sigma_s^2}}$$

Denoising

Basic approaches:

•Find useful transformations on features so that the noisy speech becomes closer to the clean speech,

•Use robust features (ear model)

In the fist approach, speech enhancement attempts to derive clean speech from noisy speech. There are essentially three approaches.

Denoising

- spectral subtraction (Boll 1979)
- Wiener filtering
- signal resporation by spectral mapping adaptive filtering techniques (e.g. Kalman),
- all pole modeling of degraded speech combining Wiener filtering with LP techniques
- microphone arrays
- Cepstral mean subtraction

Enhancement

Let **y(n,t)** be the noisy signal, **Y** be its transform, **Z** be an estimation of the noise transform, **S** the estimation of the signal spectrum, **s** its inverse transform. Enhancement consists in finding the gain **G(Y,Z)** such that:

S=G(Y,Z)Y

is as close as possible to the original signal.

For every frequency band:

$$\left|\hat{S}_k(n)\right|^2 = G_k(n)\left|Y_k(n)\right|^2$$

Wiener filters

A Wiener filter estimates the signal by filtering the noisy signal.

Assuming signal and noise are uncorrelated, then the filter is designed with the purpose of minimizing the mean-square difference between the denoised signal and the noisy signal in each band or for each frequency sample.

The criterion to be minimized can be expressed with the sum of the signal distortion and the noise residual, the fist one been proportional to the signal variance and the second one being proportional to the noise variance.

Wiener filter

When both noise and speech FFT coefficients have gaussian distribution, the optimal speech estimator is Wiener:

$$\hat{S} = E(S|Y) = \frac{\sigma_s^2}{\sigma_s^2 + \sigma_z^2} Y = \frac{\xi}{1+\xi} Y$$

$$\xi = \frac{\sigma_s^2}{\sigma_z^2} \quad a - priori \quad SNR$$

σ_s^2 average of $|S|^2$; σ_z^2 average of $|Z|^2$

Wiener filters

The Wiener filter has a frequency response for the n-th time sample and the k-th frequency sample given by:

$$G_k(n) = \frac{\sigma_{S_k}(n)}{\sigma_{S_k}(n) + \sigma_{Z_k}(n)}$$

A generalized Wiener filter contains modifications of the above definition.

$$G_k(n) = \max\left\{ \frac{|X_k(n)|^2 - \alpha(n)|Z_k(n)|^2}{|X_k(n)|^2}, \beta(n) \right\}$$

where $\eta(n)$ is an overestimation factor and $\beta(n)$ is usually a constant (Moticek et al., 2002)

Linear spectral subtraction

$$S(k) = \begin{cases} Y(k) - \alpha Z(k) & \text{if} \quad Y(k) > \alpha Z(k) \\ \beta Z(k) & \text{otherwise} \end{cases}$$

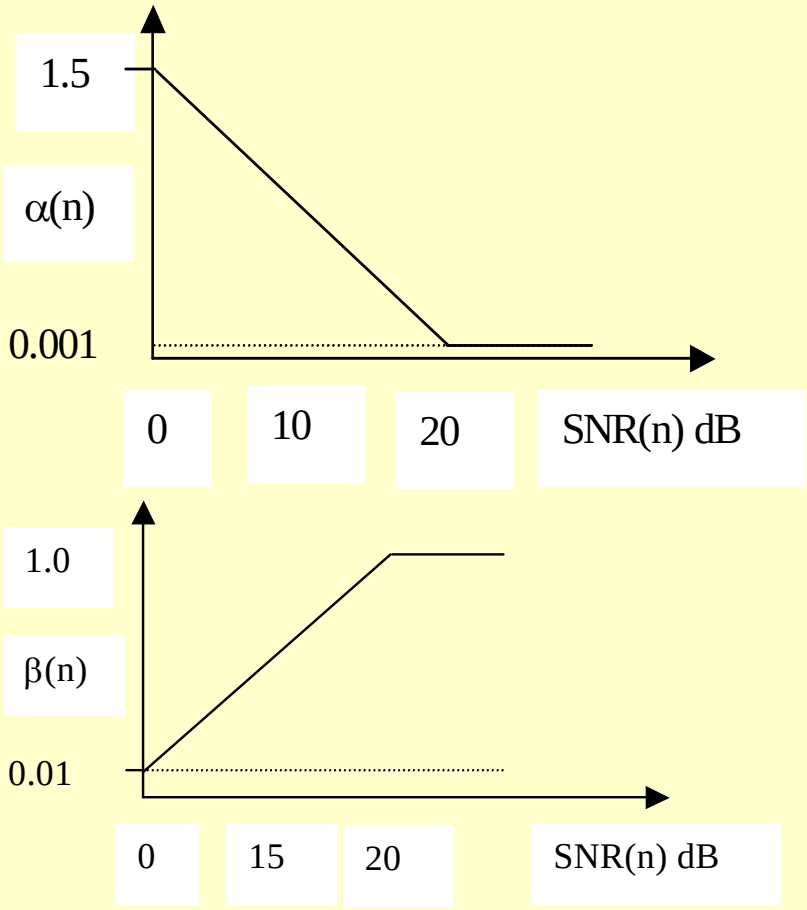
Flooring

Overestimation factor

They can be time and SNR dependent

$$SNR(n) = 10 \log_{10} \left(\frac{\sum_k |s_k(n)|^2}{\sum_k |\hat{Z}_k(n)|^2} \right)$$

Non-linear Spectral Subtraction



Cepstral noise removal

Efficient on-line noise removal of noise from speech cepstra is proposed with the SPLICE algorithm (Microsoft). Without any assumption about how noisy cepstra are produced, non-linear and possibly non.stationary distortions can be considered. The basic idea is to learn a joint probability of clean *x* and noisy speech *y* from simultaneous recording of clean and distorted speech..

One way to model the joint probability P(s,y) is:

$$P(s,y)=P(s|y)P(y)$$

P(s|y) will have parameters which are not a linear function of *y*.

Cepstral noise removal

In SPLICE, an auxiliary discrete random variable g is introduced which partitions the acoustic space into regions in which the relation between x and y is linear, so :

A noisy signal Y can be expressed as function of clean signal and noise as follows:

$y=s+r(y)$

Where r is a function which can be piece-wise leading to:

$y=s+r_g$

Noise estimation

$$N'_k(n) = \begin{cases} 0.4\ N_k(n) \text{ if } NX_{rel,k}(n) < \vartheta \\ 1.2\ N'_k(n) \text{ otherwise} \end{cases}$$

$$NX_{rel,k}(n) = \frac{NX_k(n) - NX_{min,k}(n)}{NX_{max,k}(n) - NX_{min,k}(n)}$$

$$NX_k(n) = \frac{N_k(n)}{P_{xk}(n)}$$

$$P_{x,k}(n) = (1-\alpha)X_k(n-1) + \alpha\left|X_k(n)\right|$$

$$N_k(n) = \min\left[N_k(n-D), P_{x,k}(n)\right]$$

Results (Aurora3)

Test Conditions	WER CH0 (%)	WER CH1 (%)	Overall WER (%)	
JRASTAPLP	1.4 (0.2)	41.3 (0.8)	21.4 (0.5)	IT
JRASTAPLP+SS	1.0 (0.2)	23.5 (0.7)	12.2 (0.4)	IT
MRA+PCA	0.9 (0.2)	39.4 (0.8)	20.1 (0.5)	IT
MRA+PCA+SS	0.8 (0.2)	19.1 (0.7)	9.9 (0.4)	IT
MRA+PCA+ST	0.9 (0.2)	23.5 (0.7)	12.2 (0.4)	IT
MRA+PCA+SS	0.9 (0.2)	10.6 (0.5)	5.8 (0.3)	SP
MRA+PCA+SS	2.5 (0.3)	9.8 (0.6)	6.3 (0.4)	GER

Introducing frame reliability

In (Bernard and Alwan, 2002), it is proposed to weight observation probabilities with an exponent which depends on the reliability of the received frame.

This reliability is a function of the ratio of the likelihood of the first and the second candidate in the Nbest list. This is consistent with maximum likelihood decoding.

Confidence indicators

Posterior probability $P(W|A)$

Comparison with anti-model

Consensus among different systems to compute the probability that hypothesis is correct given confidence features

……

Likelihood ratios

Likelihood Ratio Tests

- Likelihood ratio (LR) test (Lleida and Rose, 1996)

$$LR(A, \lambda^c, \lambda^a) = \frac{P(A \mid \lambda^c)}{P(A \mid \lambda^a)} \underset{H_1}{\overset{H_0}{\gtrless}} \tau$$

  - A: a sequence of feature vectors
  - λ^c: target model
  - λ^a: alternative model
- Word level confidence scores are obtained by combining LR scores.
- Requires training.

Problems

Performance varies with speaker and environment (difficult to compensate)

Re-training or incremental training is better if possible

Voice separation and denoising are difficult problems

Applications require careful tuning

Consider applications in which a limited amount of errors can be tolerated

Evaluation

Word Error Rate (WER)

$$\mathrm{WER} = \frac{\#\,\mathrm{Ins} + \#\,\mathrm{Del} + \#\,\mathrm{Subs}}{\#\,\mathrm{Ref.\,Words}}$$

REF: i'd like to review my services that i have
HYP: i'd like to have a review the services i have

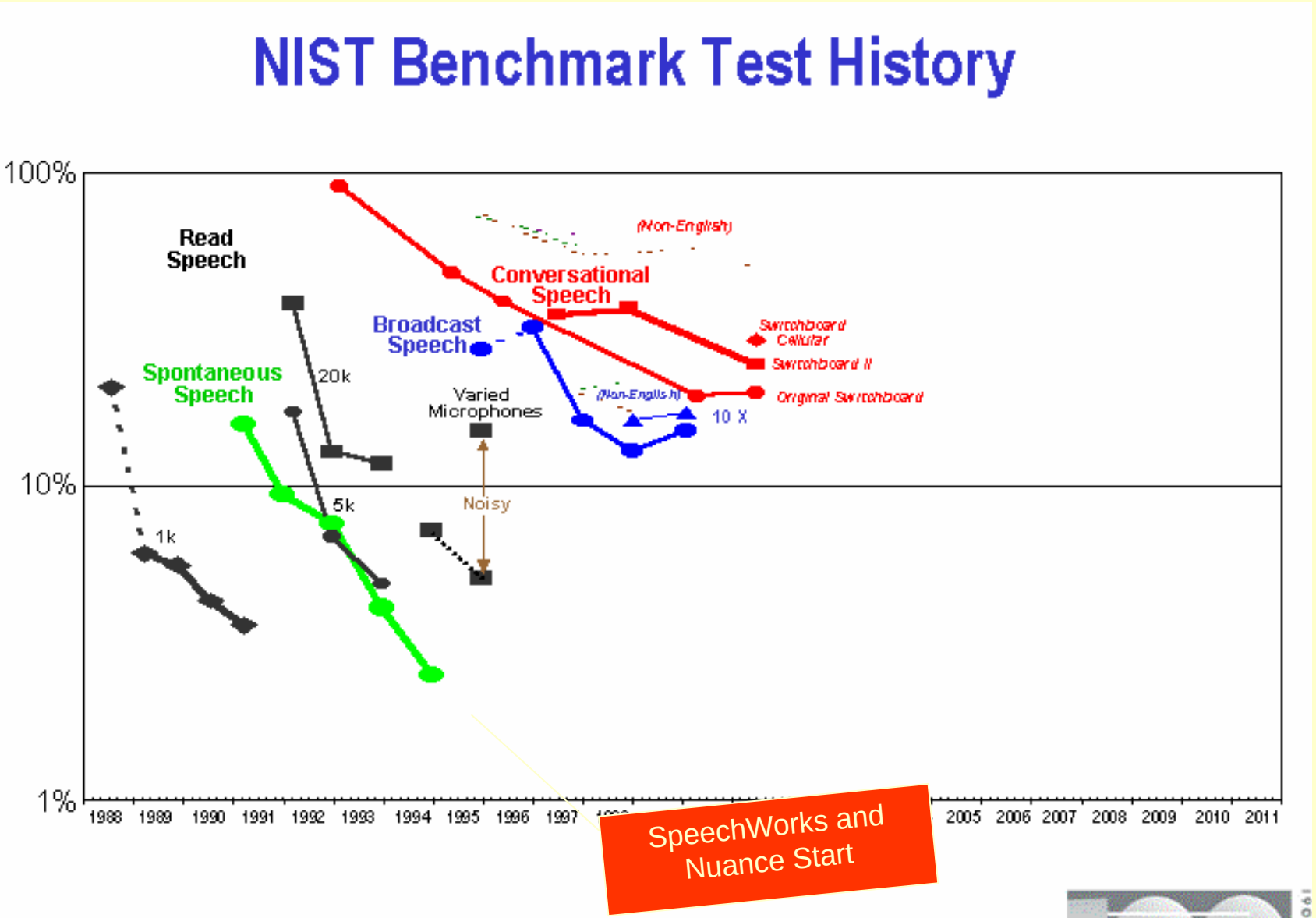
REF:	i'd	like	to	****	*	review	MY	services	THAT	i	have
HYP:	i'd	like	to	HAVE	A	review	THE	services	****	i	have
				Insertions			Substitution		Deletion			

Word Accuracy (WA)

$$\mathrm{WA} = 1 - \mathrm{WER}$$

71

History of US funded speech recognition research



Dictation results

Speaker independent continuous speech

Number of words	word error rate
65000	10%
20000	7%
1500	2.2%
10	0.23%

OOV and lexicon size

Language	corpus	number of distinct words
English	WSJ	165000
French	Le Monde	280000
Italian	Il Sole 24 ore	200000
German	FrankfurterRundshau	650000

Telephone speech

Speaker independent continuous speech

ESST	**(4000 words)**	**12-23% WER**
SWB	**(15000 words)**	**26-36% WER**
Other	**(15000 words)**	**36-55%WER**

Understanding

Air Travel Information System ATIS

46 cities, 57 aerports, 23457 flights, 1700 words

year	1990	1991	1992	1993	1994
CER	33.8%	28.2%	11.1%	13.2%	8.6%

Broadcast news

1997	27% WER
2005 Multiple steps multiple systems	10% WER