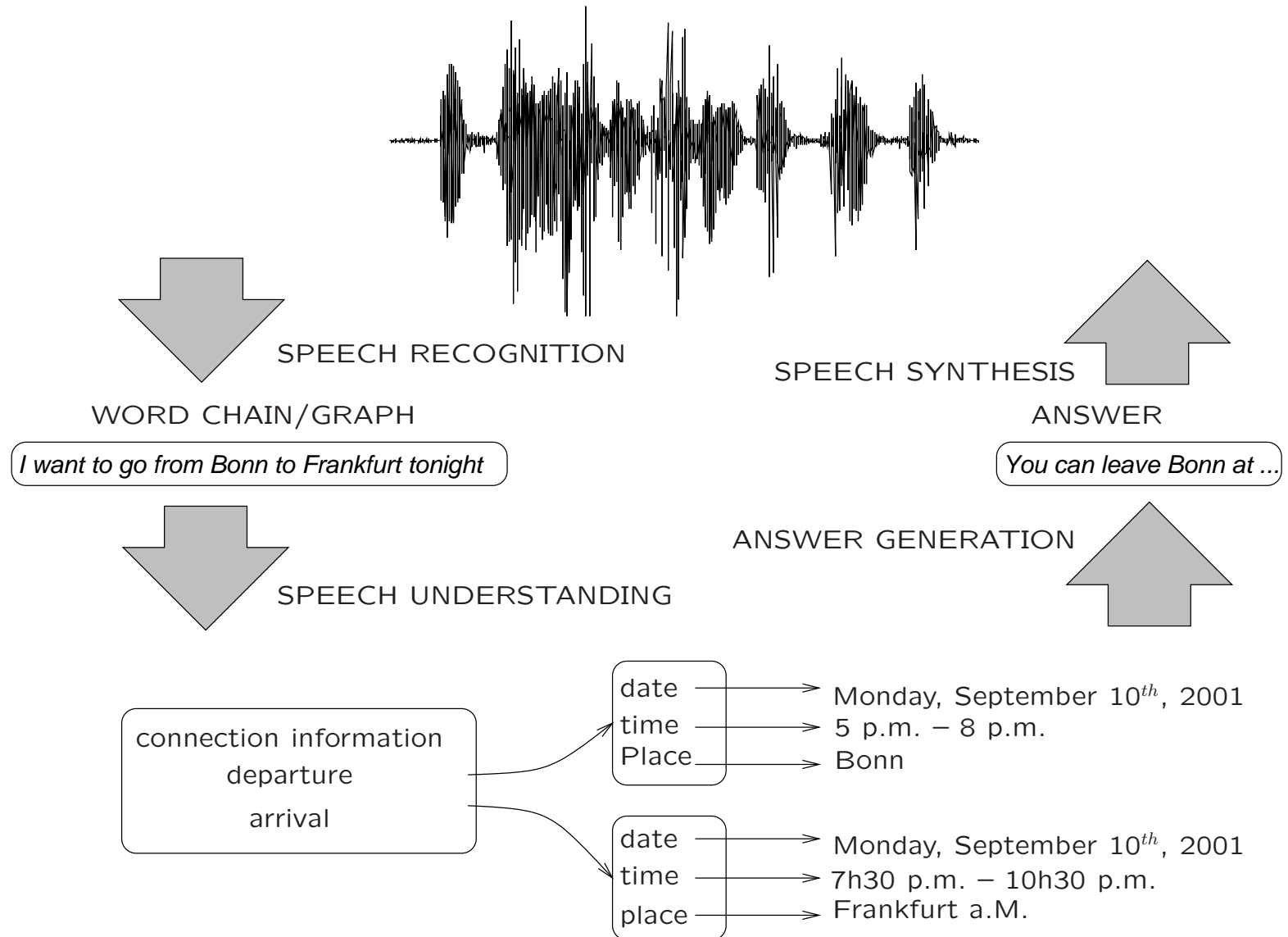
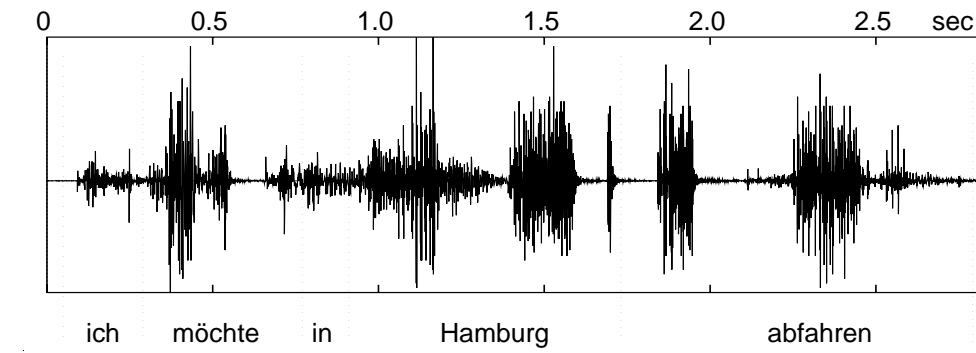


# DIALOGUE

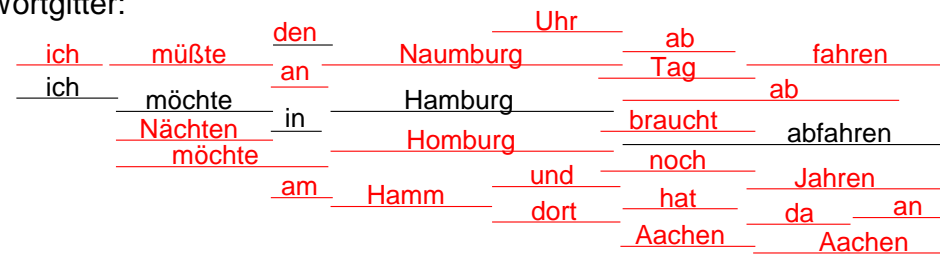


# PART I: SPEECH RECOGNITION — FROM SIGNAL TO WORD CHAIN/LATTICE/GRAPH

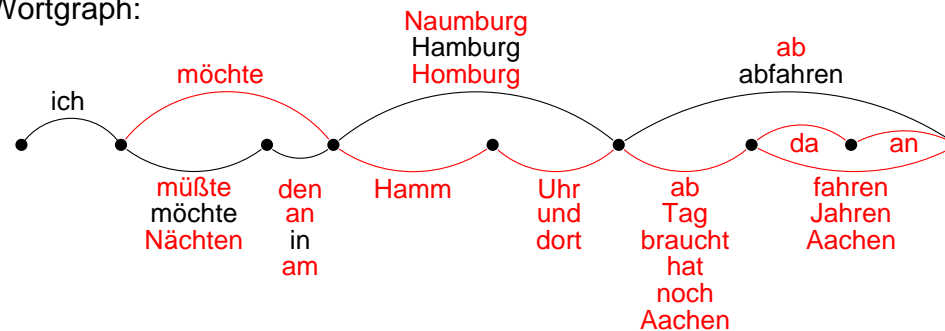
Sprachsignal:



Wortgitter:

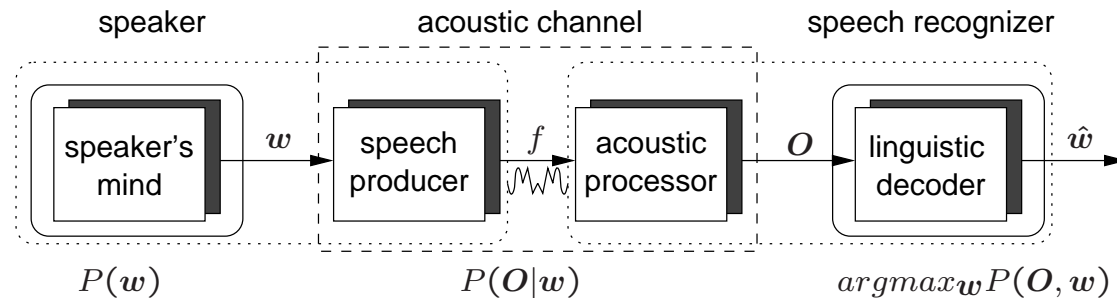


Wortgraph:



# STATISTICAL SPEECH RECOGNITION

- the acoustic channel model:



- goal: find  $\hat{w}$  that best matches  $w$

⇒ *Bayes classifier*: compute a sequence of words

$$\hat{w} = \operatorname{argmax}_w P(w|O),$$

with *maximum a posteriori probability*,  
given the acoustic evidence  $O$

# STATISTICAL SPEECH RECOGNITION

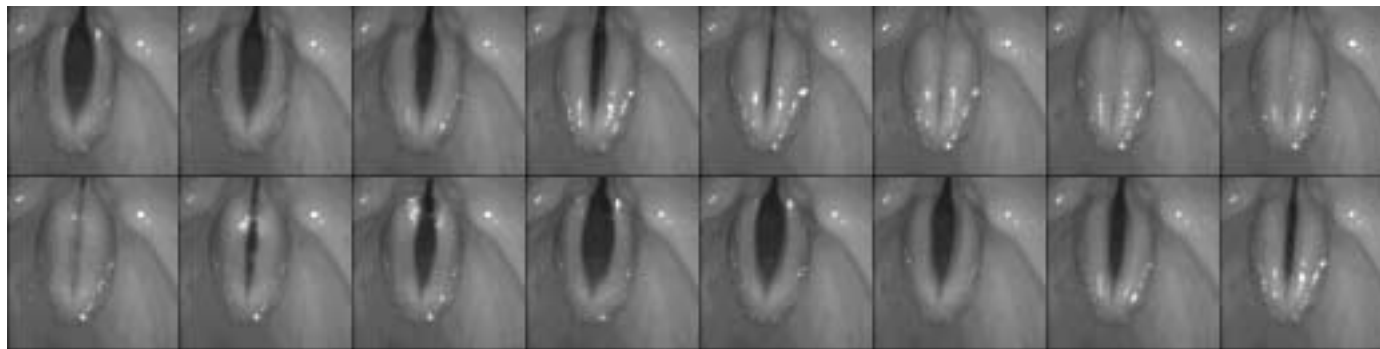
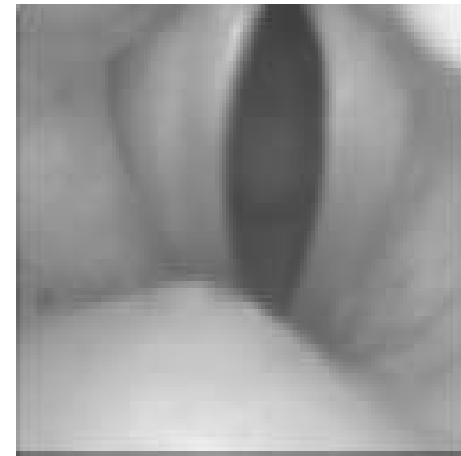
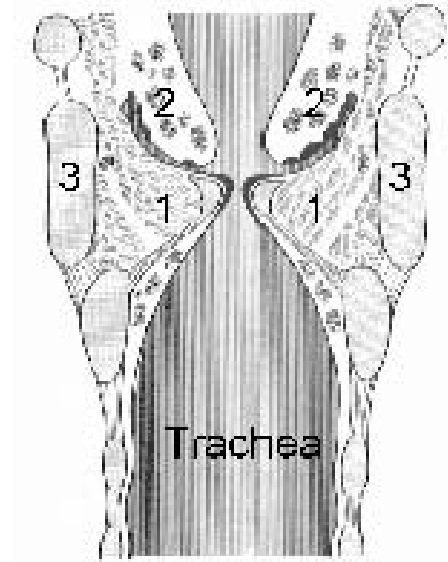
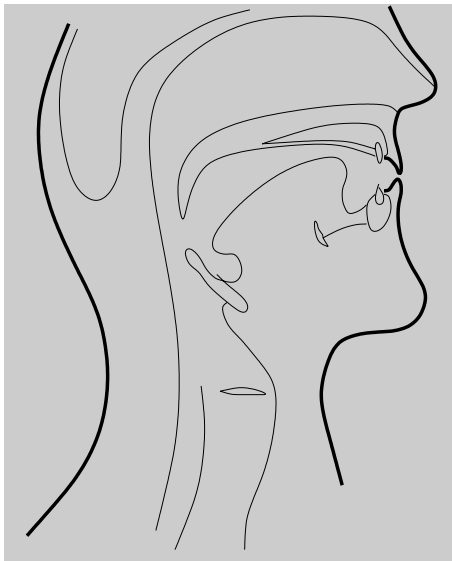
$$\begin{aligned}\hat{w} &= \operatorname{argmax}_w P(w|O) \\ &= \operatorname{argmax}_w \{P(O|w) \cdot P(w)/P(O)\} \\ &= \operatorname{argmax}_w \{P(O|w) \cdot P(w)\}\end{aligned}$$

⇒ *acoustic model* : computation of  $P(O|w)$

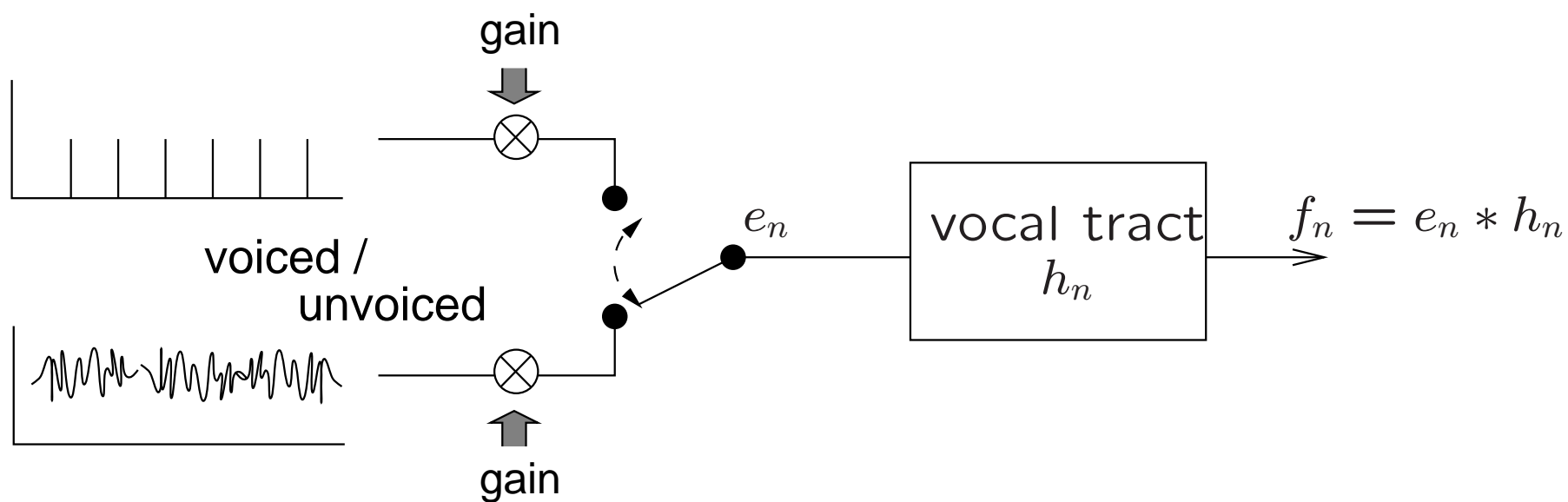
⇒ *language model* : estimation of  $P(w)$

⇒ *efficient search* : find  $\hat{w}$

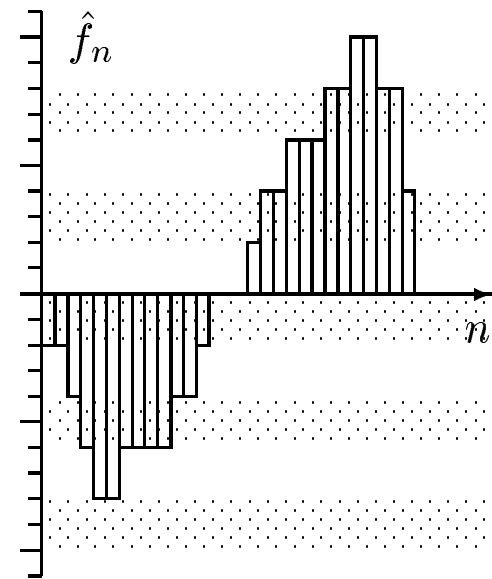
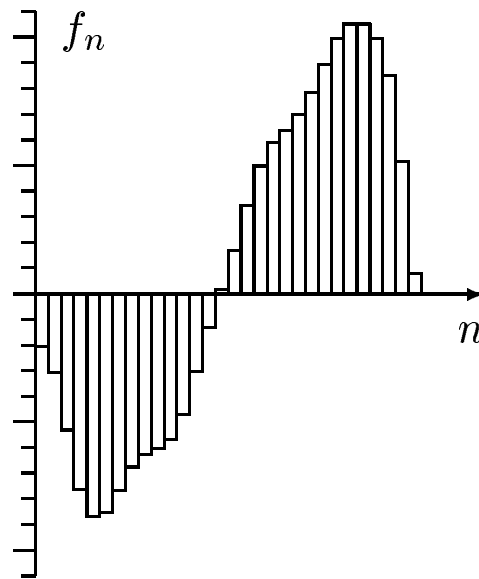
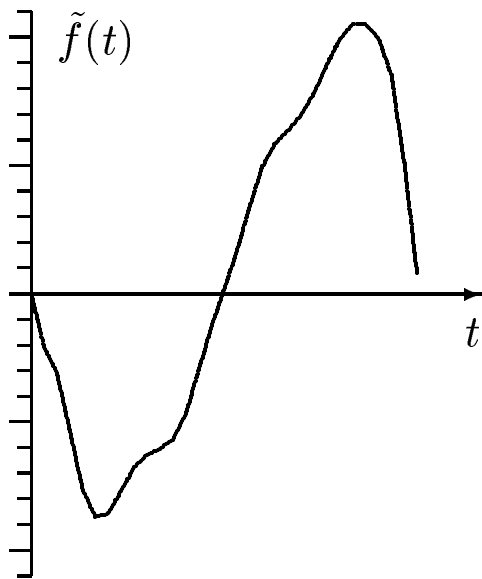
# IN THE BEGINNING THERE WAS THE SOURCE



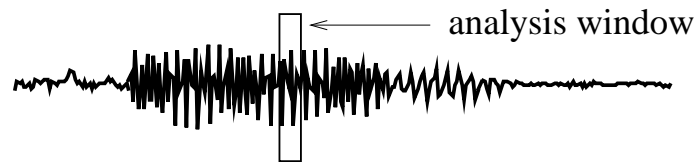
# SPEECH PRODUCTION — SOURCE-FILTER MODEL



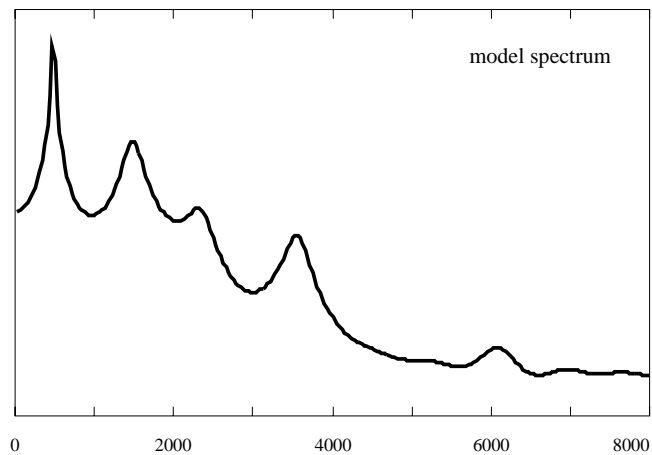
# DIGITIZATION / QUANTIZATION



# ACOUSTIC FEATURES



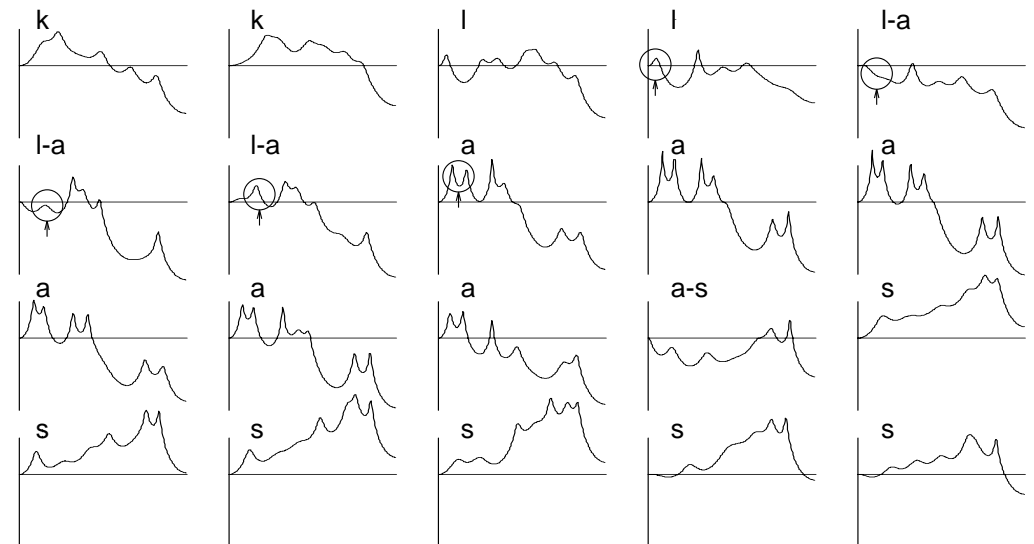
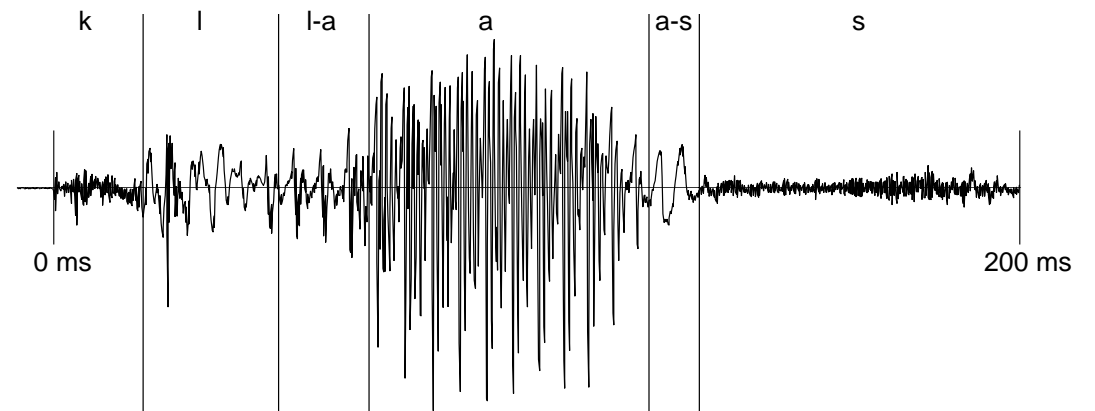
FFT



cepstrum

$$\begin{bmatrix} -0.986 \\ 1.000 \\ \vdots \\ -0.333 \end{bmatrix}$$

Was kostet eine Rückfahrkarte zweiter] **Klass** [e nach Hamburg?

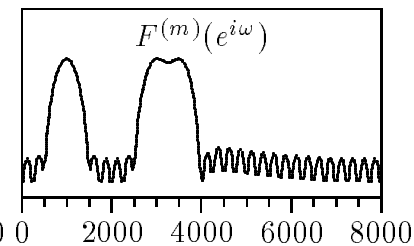
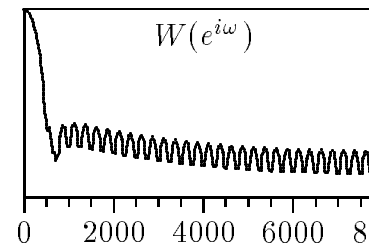
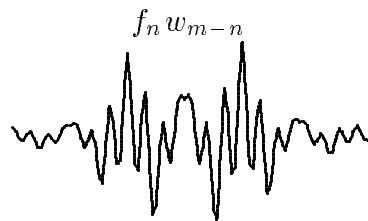
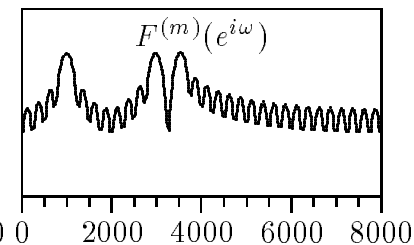
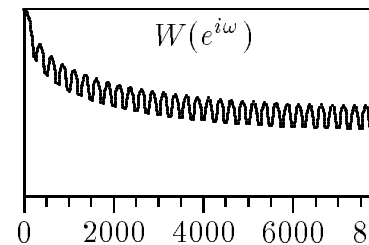
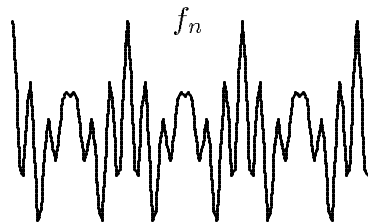


# WINDOWING

$$f_n^{(m)} = f_n \cdot w_{m-n}$$

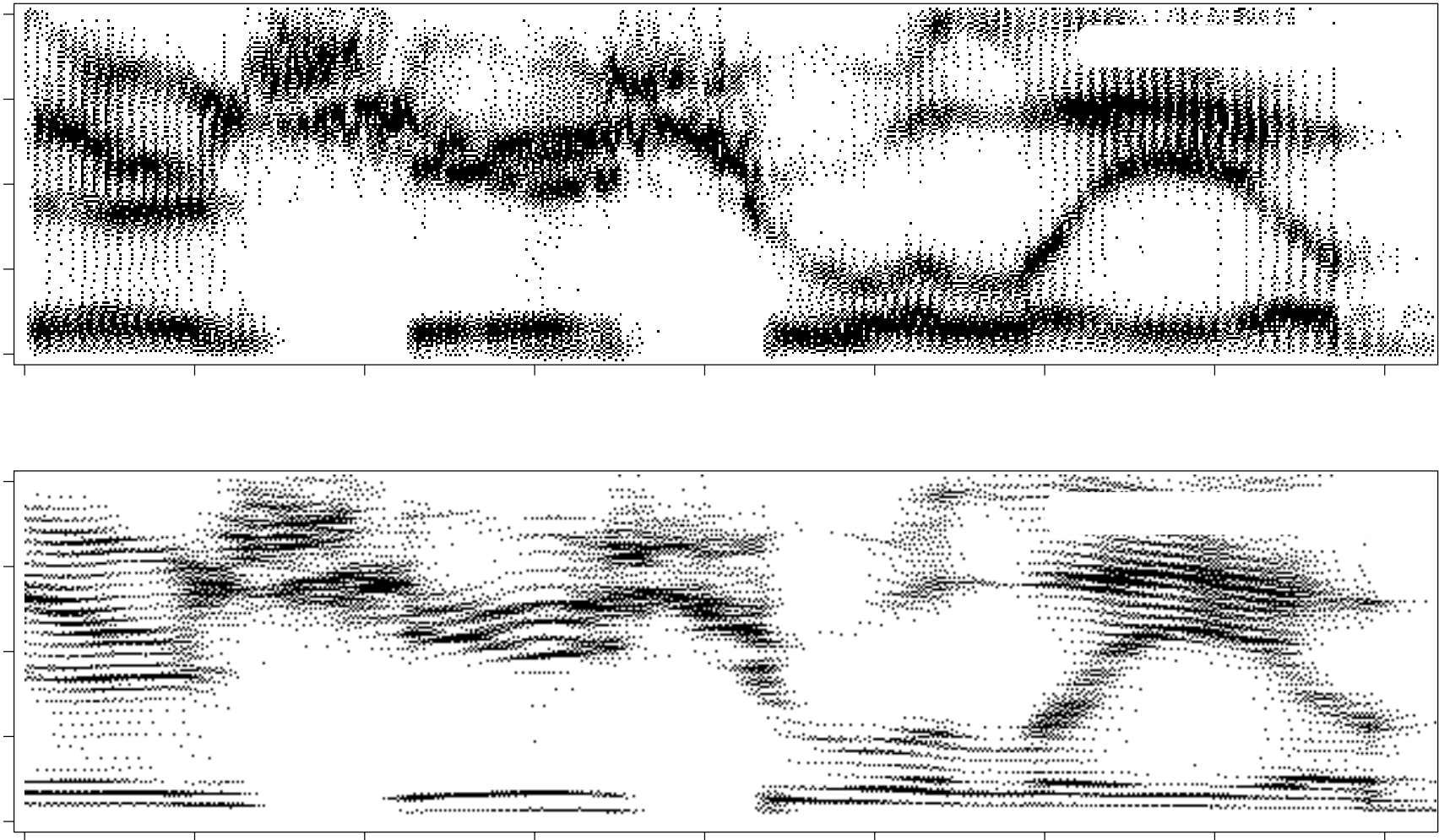
$$\begin{array}{lll} w_n^R = 1 & 13 \text{ dB} & \text{(rectangular window)} \\ w_n^M = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right) & 43 \text{ dB} & \text{(Hamming window)} \end{array}$$

$$F^{(m)}(e^{i\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} W(e^{-i\phi}) e^{-i\phi m} F(e^{i(\omega-\phi)}) d\phi$$

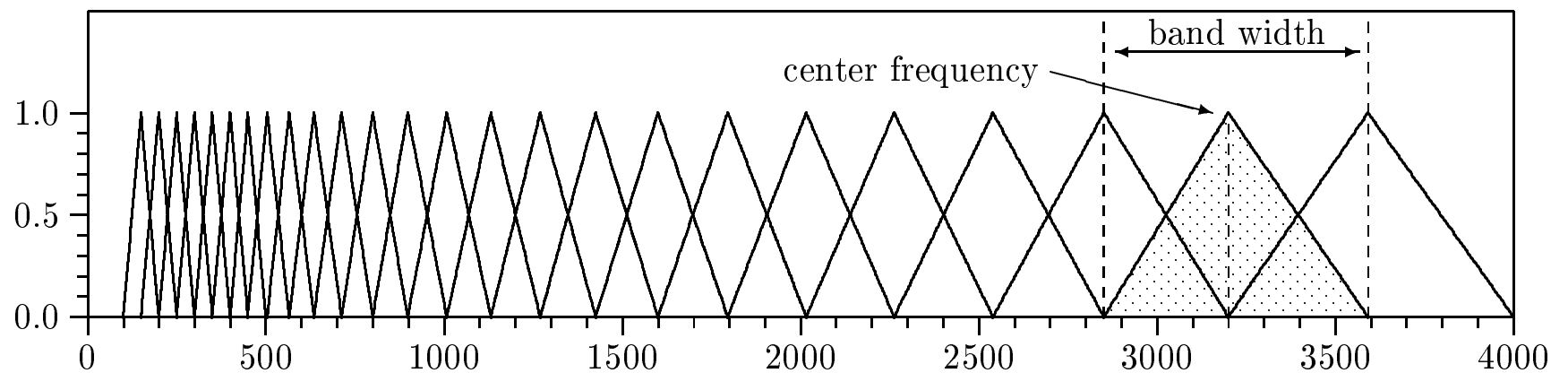
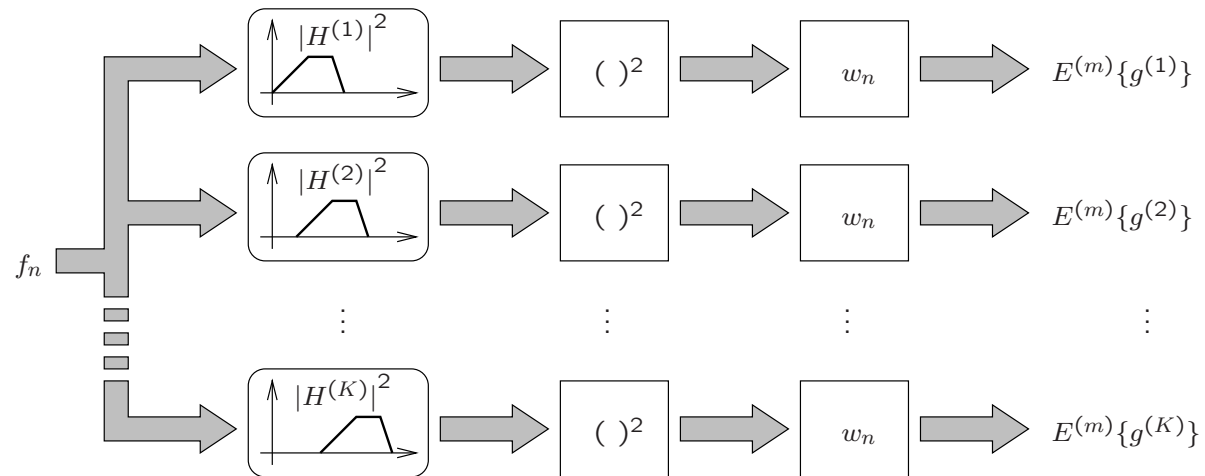


# WIDEBAND/NARROWBAND SPECTROGRAM

(frequency resolution 125 Hz respectively 19 Hz)



# (MEL) FREQUENCY BANDS

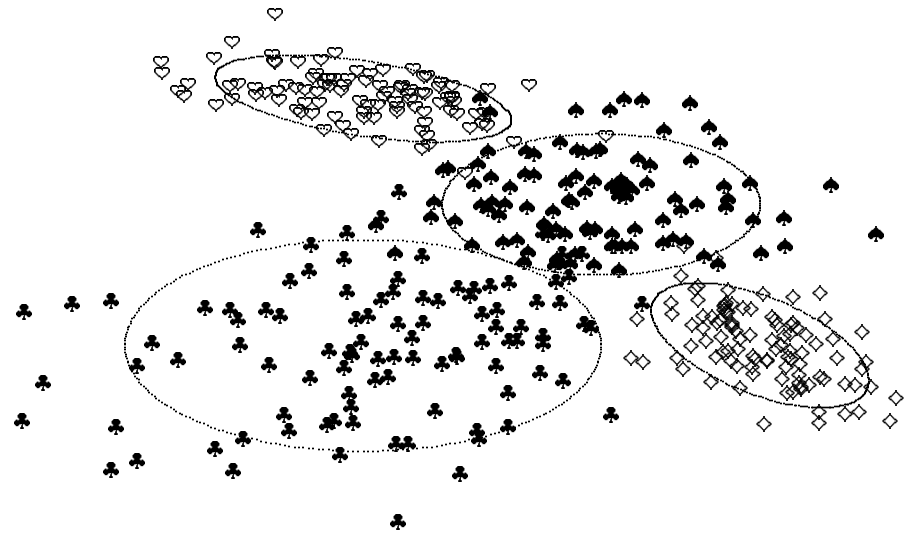


# MEL CEPSTRUM

- reduction from 128 to 256 Fourier coefficients to  $\approx 20$  band energies
- take logarithm
- apply inverse DFT or cosine transform
- take  $\approx$  first 12 coefficients
- add energy to vector of coefficients
- take derivative of each vector component over time

# VECTOR QUANTIZATION

- feature vector  $x$  belongs to unknown class  $\kappa$  ( $\approx$  phone/phone component,  $\approx$  64–512)
  - assume number of classes and estimate a priori probabilities and classwise conditional probabilities  
 $\Rightarrow$  unsupervised learning, (soft) vector quantization
  - estimate parameters with EM–algorithm
- 
- 4 normally distributed random processes  $O_i$ : sequences of  $x_i \in \mathbb{R}^2$
  - each process represented by different symbol
  - **observable:**  
result of all 4 processes
  - **hidden:**  
generating process for each observation  $x$ , a priori probability of each process



# ACOUSTIC PROCESSING

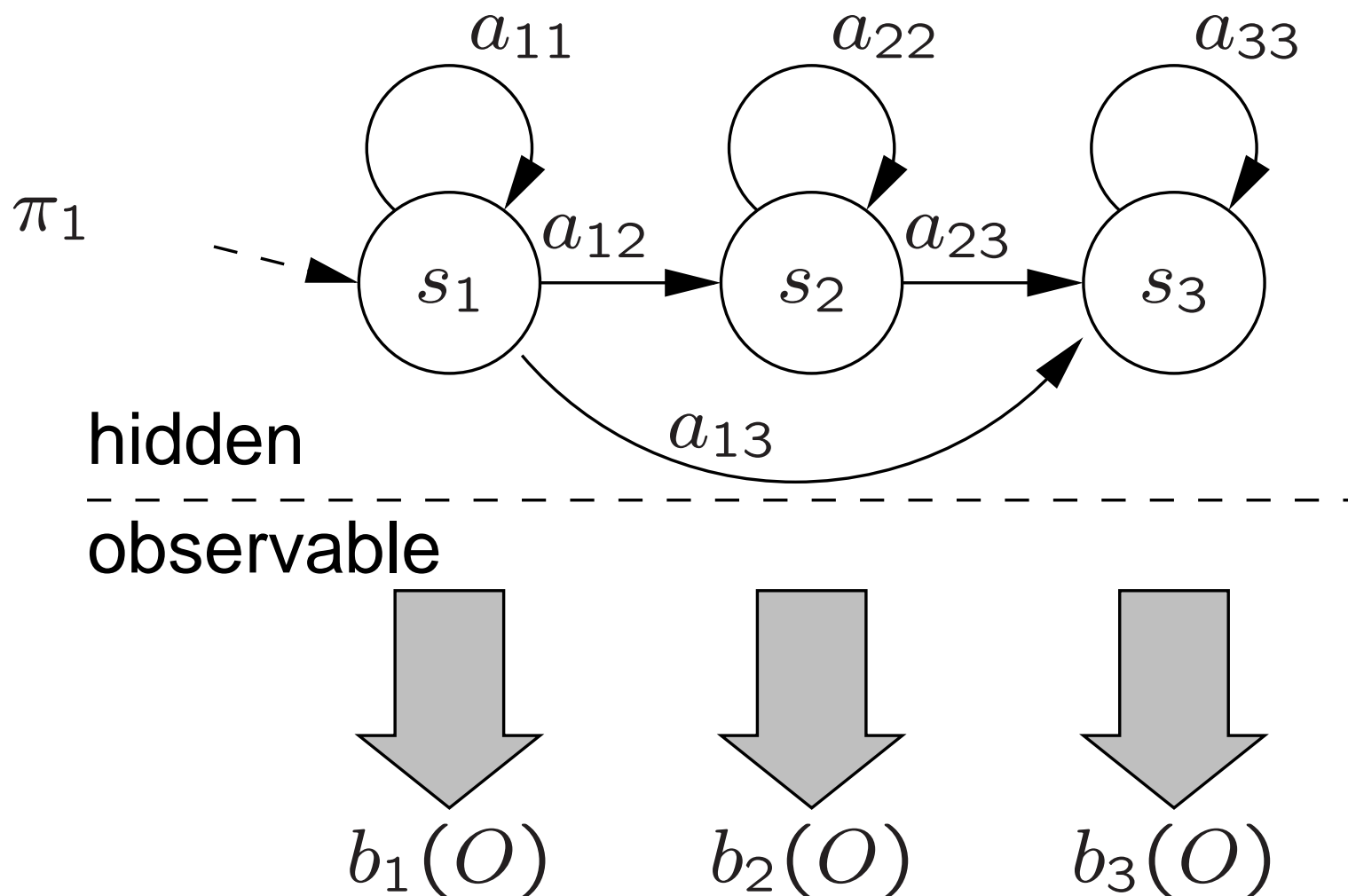
Acoustic features, computed typically every 10 msec:

- *mel cepstrum coefficients*,
  - take 10 to 20 msec
  - apply Hamming window
  - compute DFT
  - smooth spectrum, i.e. integrate over mel scaled bands
  - take logarithm
  - apply inverse DFT or cosine transform  
(separation of source and filter)
- *energy* and/or *pitch*, first (and second) order *derivative(s)*

optional: *vector quantization*

⇒ observation  $\mathbf{O}$ : sequence of  $\mathbf{x} \in \mathbb{R}^n$  or discrete class labels

# HIDDEN MARKOV MODELS



# HIDDEN MARKOV MODELS

- $\pi = (\pi_i)$ : initial state probabilities
- $A = [a_{ij}]$ : transition probabilities
- $B = (b_j)$ : output probabilities
  - discrete HMM : finite output alphabet, e.g.: VQ labels
  - *continuous HMM*: Gaussian mixture models

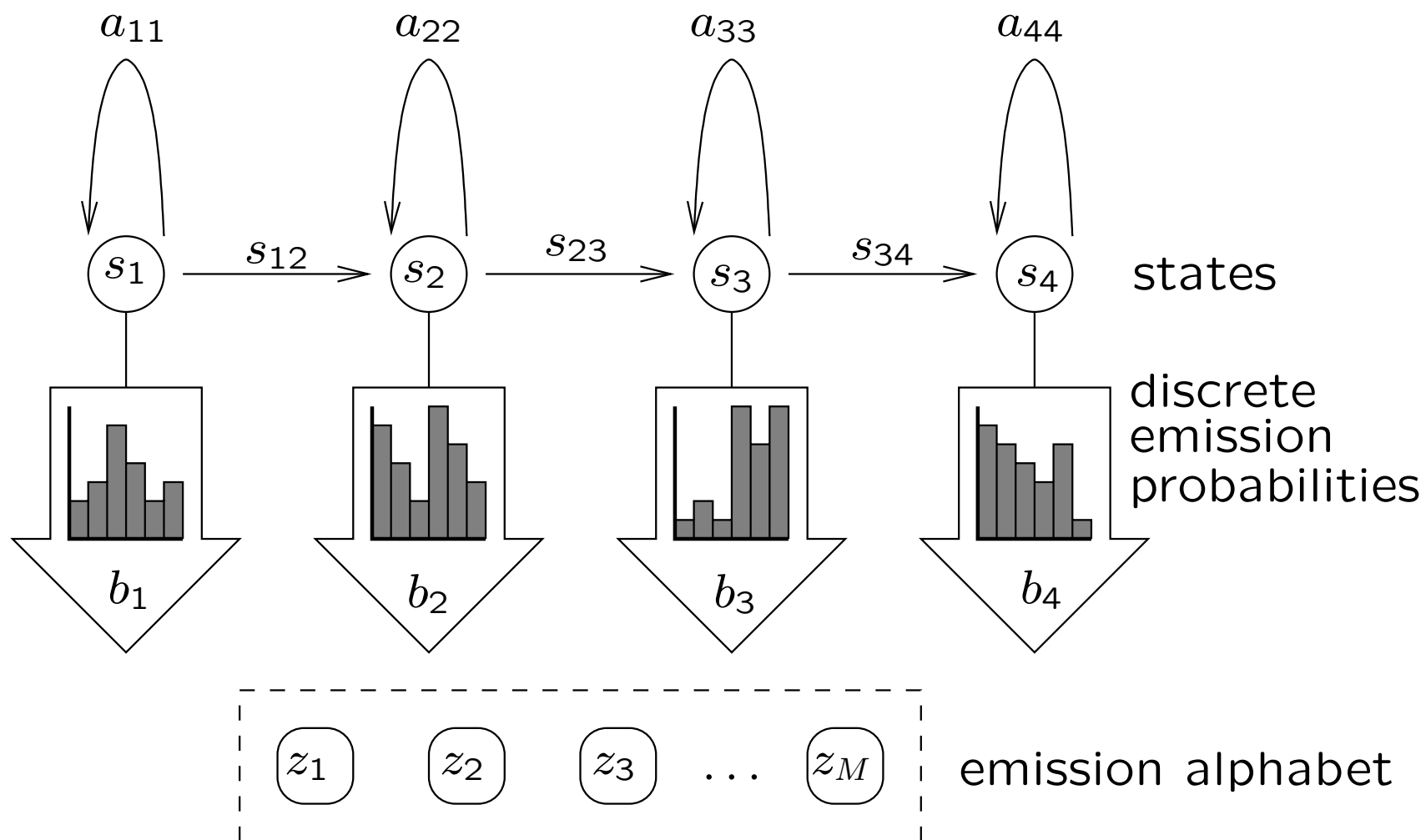
$$b_j(x) = \sum_{k=1}^{K_j} \omega_{jk} \cdot \mathcal{N}(x | \mu_{jk}, \Gamma_{jk})$$

- semi-continuous HMM : tied mixture models

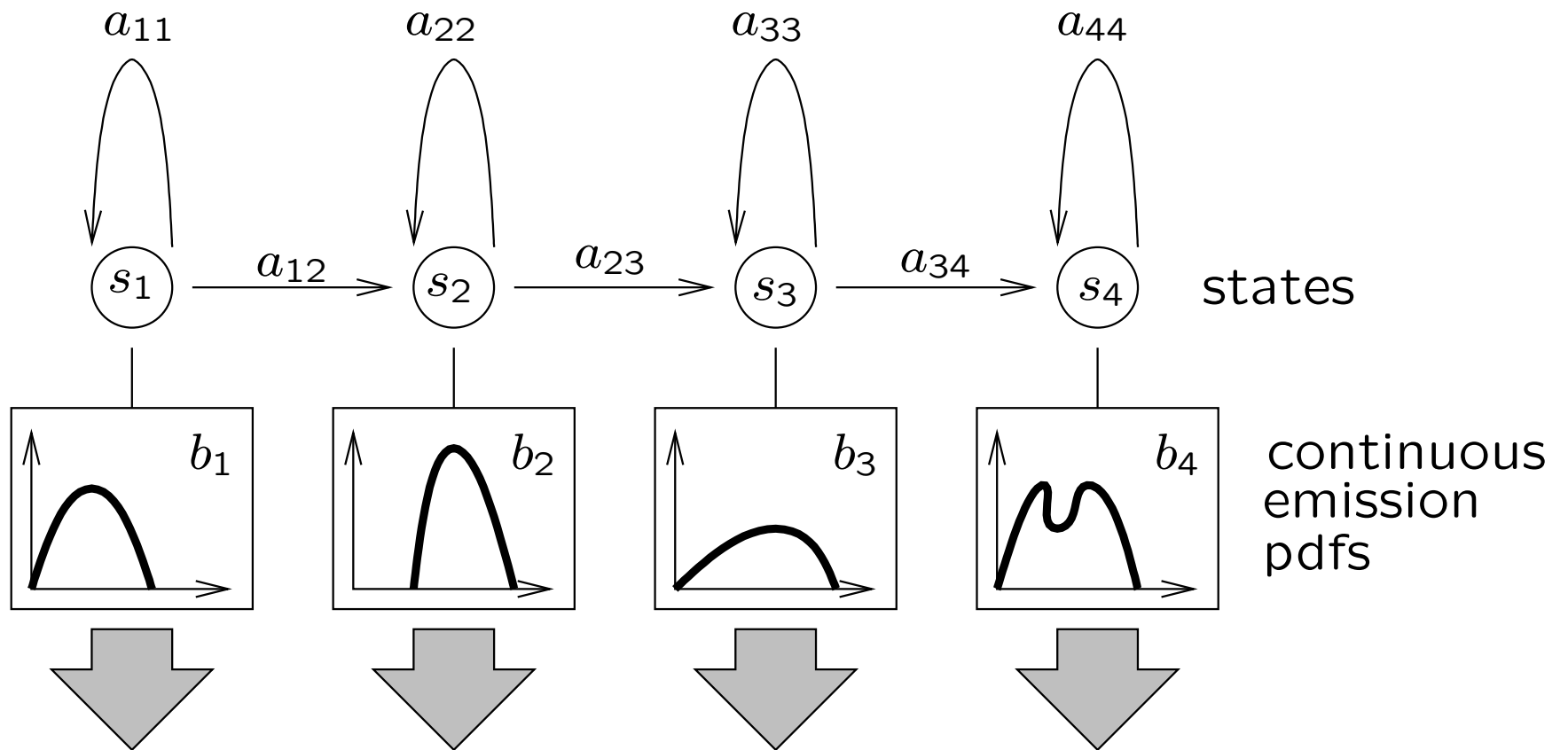
$$b_j(x) = \sum_{k=1}^K \omega_{jk} \cdot \mathcal{N}(x | \mu_k, \Gamma_k)$$

- first order assumption: transition  $ij$  only depends on  $i$ , not on history, i.e. the states before  $i$

# DISCRETE HMMS



# (SEMI)CONTINUOUS HMMS

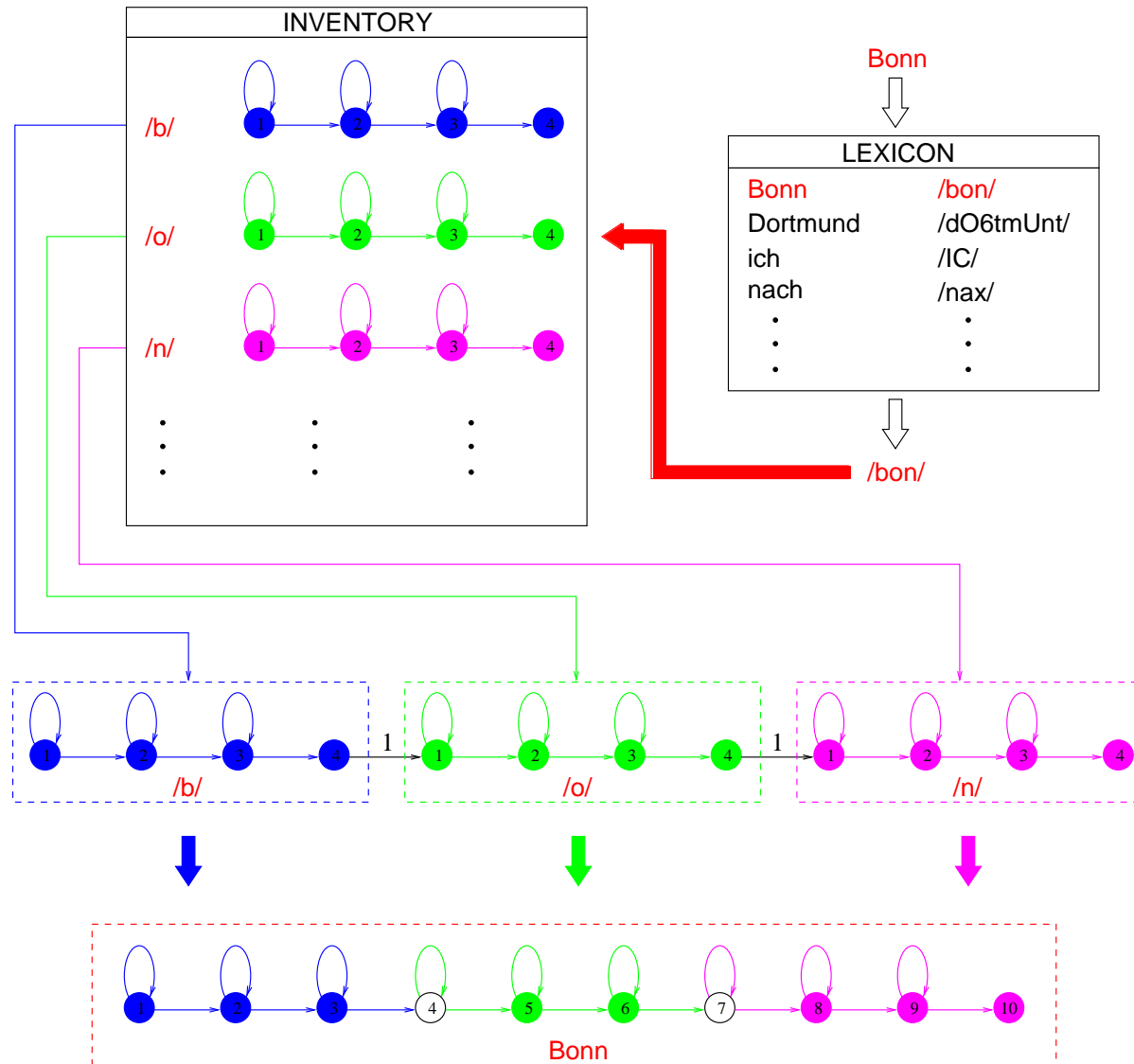


# HIDDEN MARKOV MODELS

HMM based speech recognizers have to address the following questions:

- inventory : which speech events should be modeled ?
- topology : number of states, transitions ?
- decoding : how to compute  $P(O|\lambda)$  ?
- training : how to compute estimates for  $\lambda = (\pi, A, B)$  ?

# HMM INVENTORY

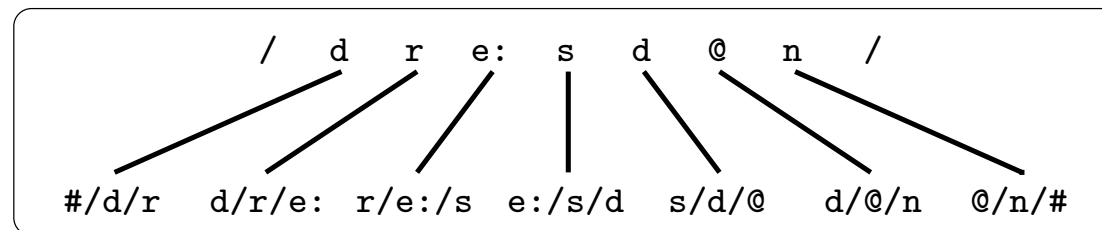
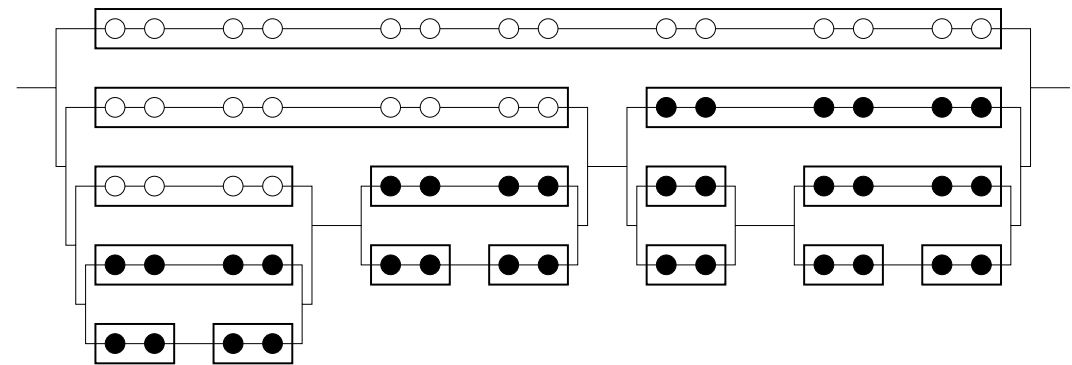
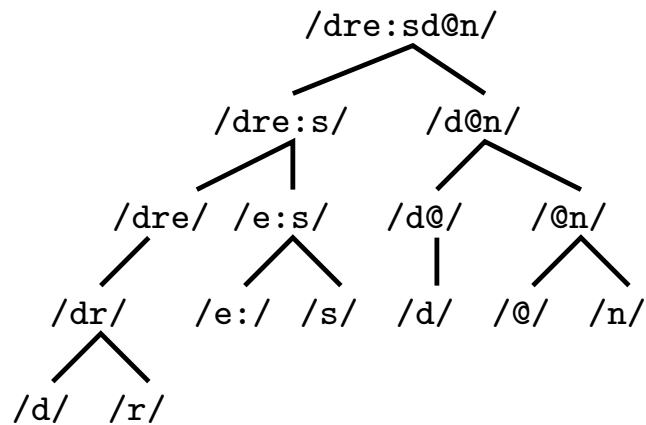


# HMM INVENTORY:

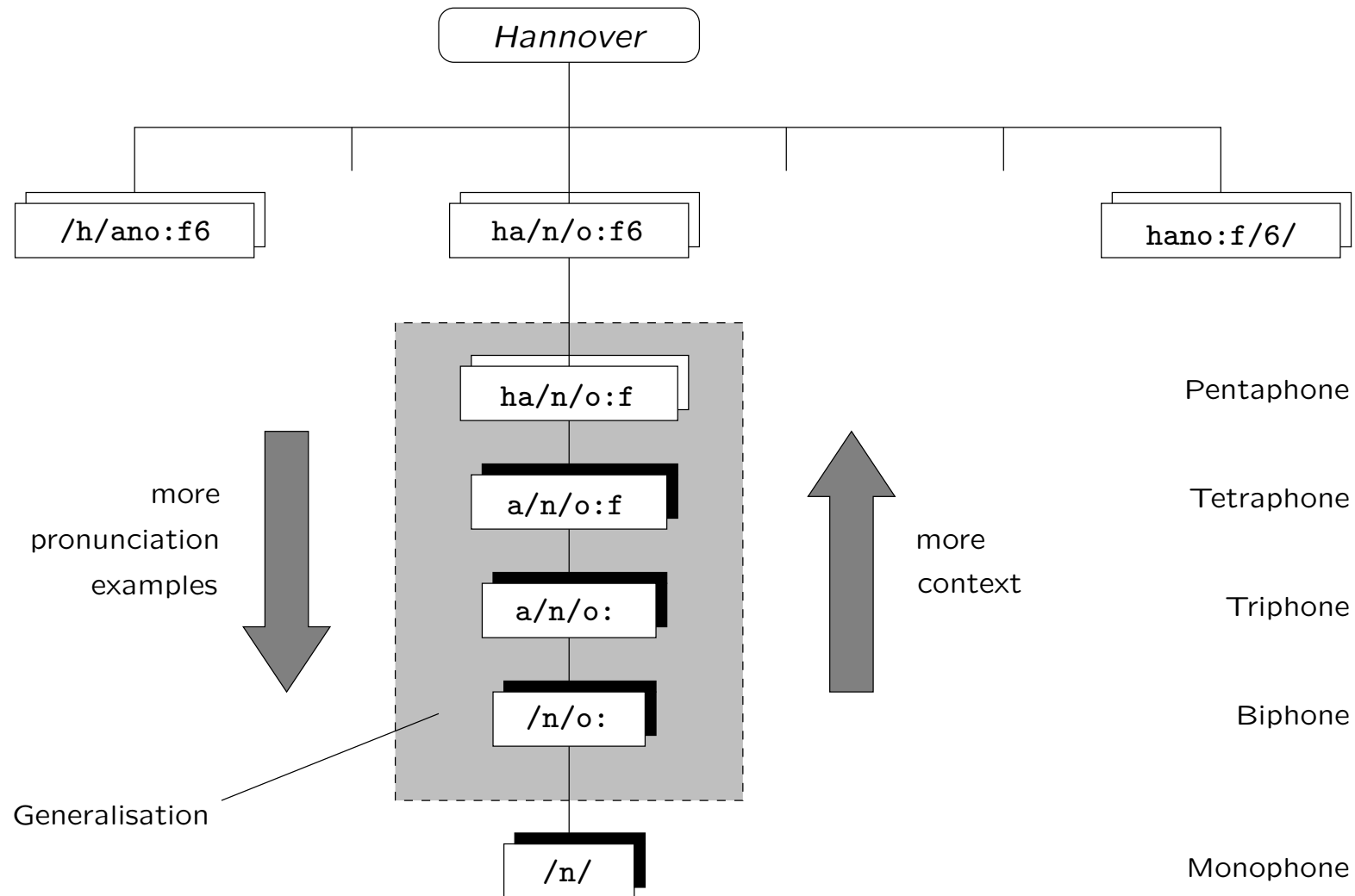
- capture effects of *coarticulation*
  - capture properties of *non-stationary sounds*
  - provide *reliable* estimates
- ⇒ *context dependent subword units:*  
subphone, phone, triphone, etc.

# HMM INVENTORY

- subword units
  - phonemes
  - syllable parts
  - syllables
  - phonemes in context

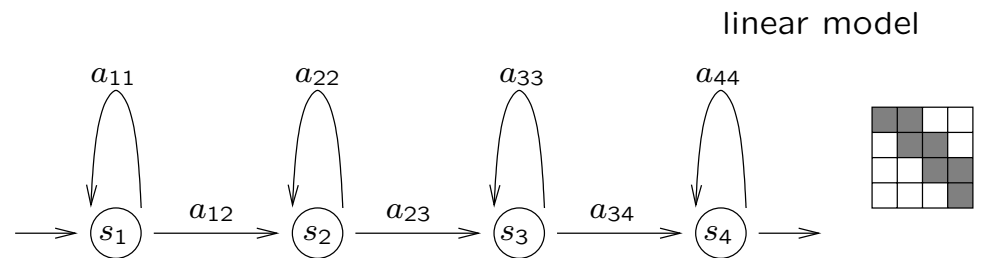
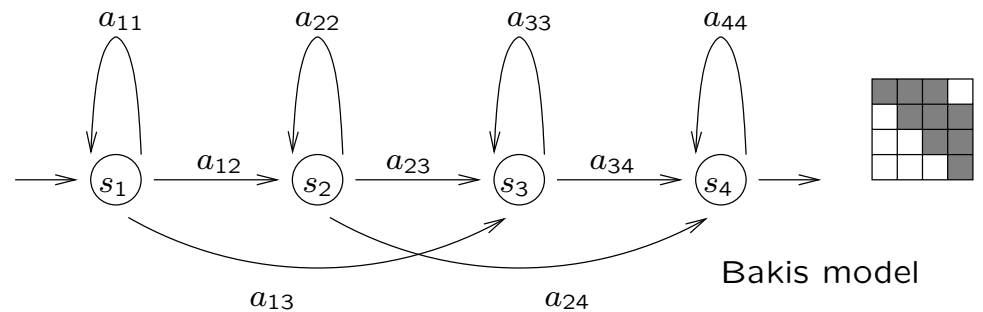
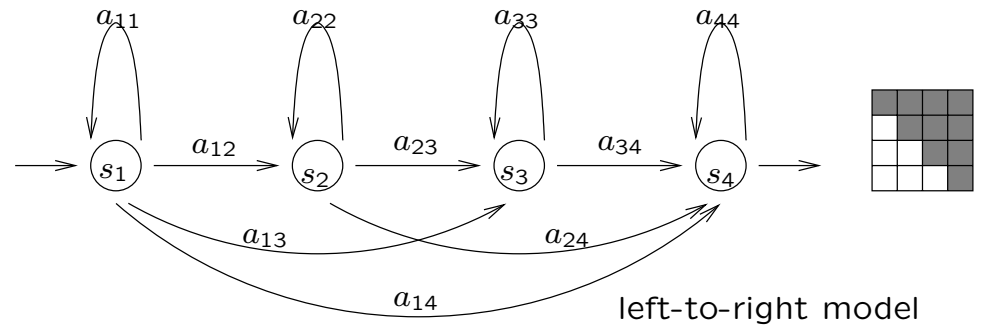
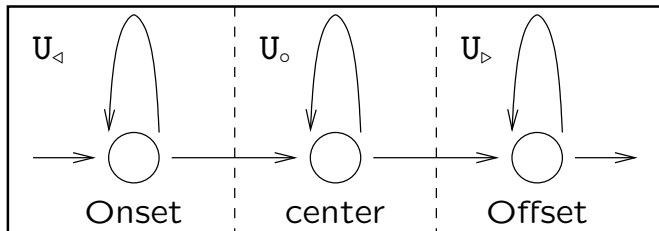


# HMM INVENTORY

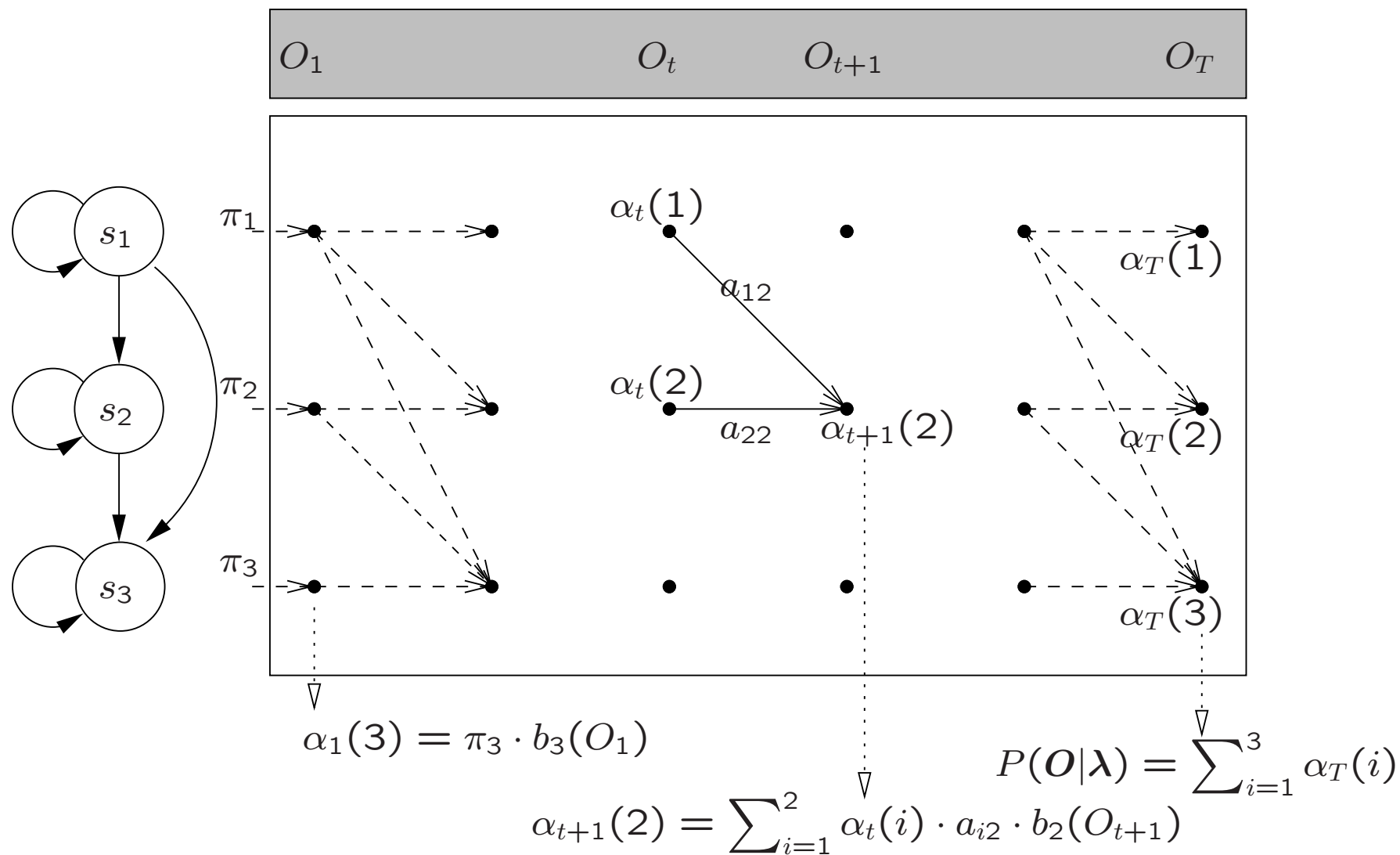


# HMM TOPOLOGY:

- intuitive design,
- *left-to-right models:*  
 $a_{ij} = 0$  if  $i > j$



# HMM DECODING: FORWARD ALGORITHM / VITERBI DECODER



# FORWARD ALGORITHM

- define *forward probability*:

$$\alpha_t(j) = P(O_1 \dots O_t, q_t = j | \lambda)$$

- initialize:

$$\alpha_1(i) = \pi_i \cdot b_i(O_1)$$

- recursion:

$$\forall j = 1, \dots, N, \forall t = 2, \dots, T :$$

$$\alpha_t(j) = \sum_{i=1}^N (\alpha_{t-1}(i) \cdot a_{ij}) \cdot b_j(O_t)$$

- termination:

$$P(\mathbf{O} | \lambda) = \sum_{i=1}^N \alpha_T(i)$$

- 
- analogous: (necessary for training)

$$\beta_t(j) = P(O_{t+1} \dots O_T, q_t = j | \lambda)$$

# VITERBI DECODER

- define *forward probability*:

$$\alpha_t(j) = P(O_1 \dots O_t, q_t = j | \lambda)$$

- initialize:

$$\alpha_1(i) = \pi_i \cdot b_i(O_1)$$

- recursion:

$$\forall j = 1, \dots, N, \forall t = 2, \dots, T :$$

$$\alpha_t(j) \approx \max_{i=1}^N \{ \alpha_{t-1}(i) \cdot a_{ij} \} \cdot b_j(O_t)$$

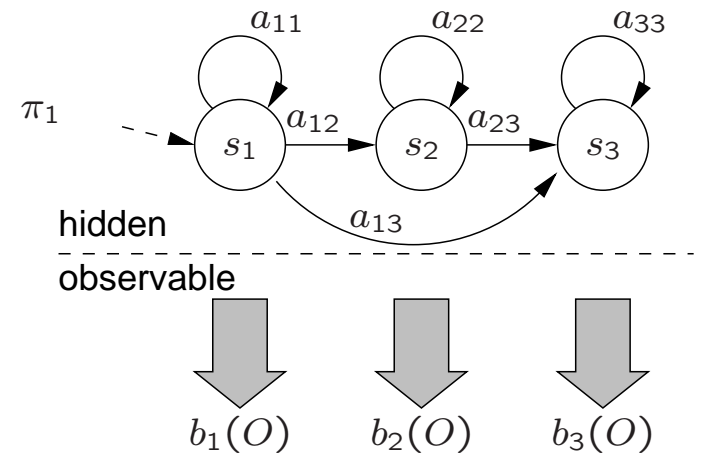
- terminate:

$$P(\mathbf{O} | \lambda) \approx \max_{i=1}^N \{ \alpha_T(i) \}$$

# HMM TRAINING (PARAMETER ESTIMATION):

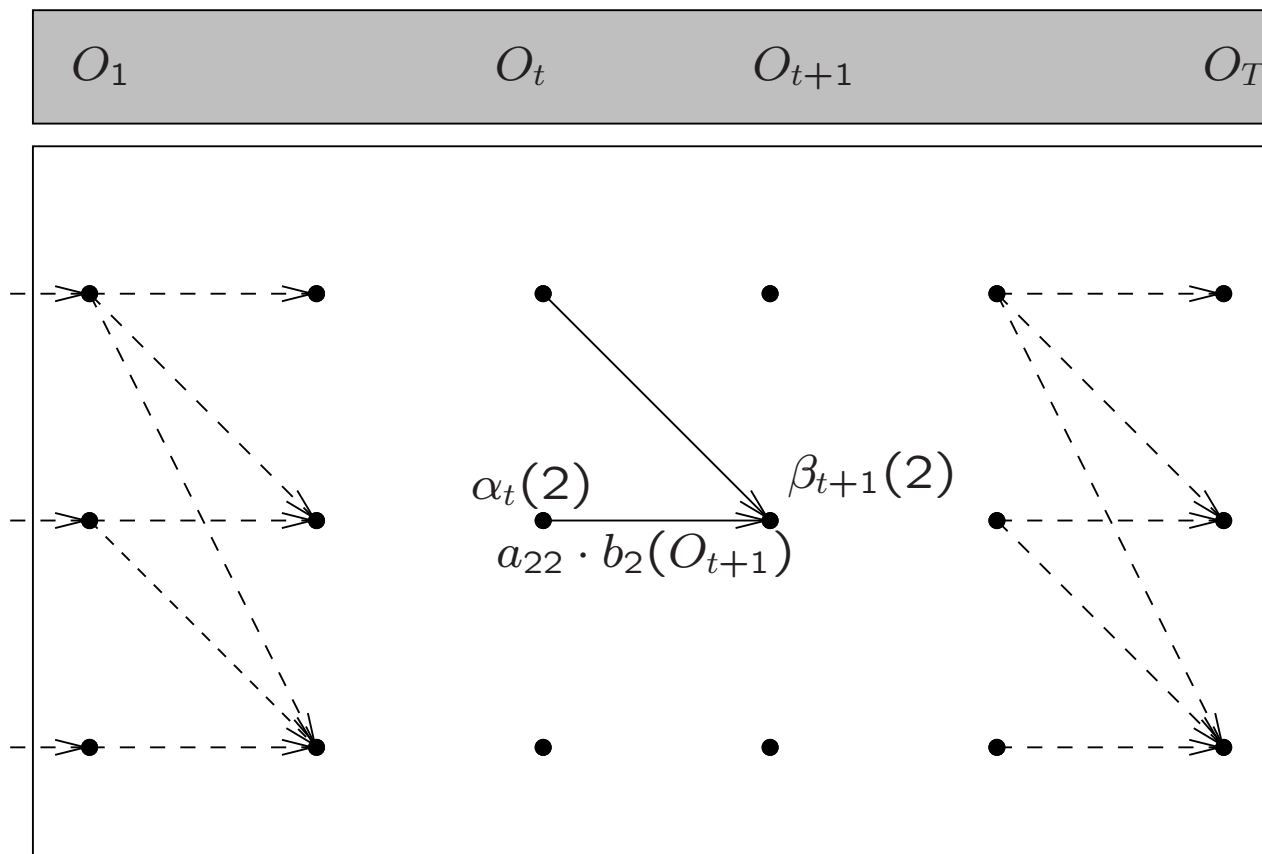
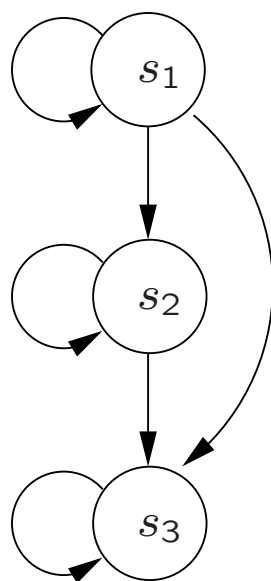
- *maximum likelihood methods:*  
forward-backward algorithm, Viterbi training
- discriminative training  
maximum mutual information training:  
more accurate, but expensive

- $\pi = (\pi_i)$ : initial state probabilities
- $A = [a_{ij}]$ : transition probabilities
- $B = (b_j)$ : output probabilities



- first order assumption: transition  $ij$  only depends on  $i$ ,  
not on history, i.e. the states before  $i$

# HMM TRAINING



## BAUM–WELCH–ALGORITHM

GIVEN: ‘old’ model parameters  $\lambda$ ,  $\alpha_t(j)$ ,  $\beta_t(i)$

WANTED: ‘new’ model parameters  $\hat{\lambda}$  with  $\mathcal{L}_{\text{HMM}}(\hat{\lambda}) \geq \mathcal{L}_{\text{HMM}}(\lambda)$

$\Rightarrow$  **improved model**

**a posteriori transition probabilities** for  $s_i \rightarrow s_j$  at time  $t$ :

$$\begin{aligned}\xi_t(i, j) &= P(q_t = i, q_{t+1} = j \mid \mathbf{O}, \lambda) \\ &= \frac{P(q_t = i, q_{t+1} = j, \mathbf{O} \mid \lambda)}{P(\mathbf{O} \mid \lambda)} \\ &= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \alpha_t(i) \beta_t(i)}, \quad 1 \leq t < T\end{aligned}$$

**a posteriori state probability** for  $s_i$  at time  $t$ :

$$\gamma_t(i) = P(q_t = i \mid \mathbf{O}, \lambda) = \frac{\alpha_t(i) \beta_t(i)}{\sum_{j=1}^N \alpha_t(j) \beta_t(j)} = \sum_{j=1}^N \xi_t(i, j)$$

# BAUM–WELCH–ALGORITHM, IMPROVED PARAMETERS $\hat{\lambda}$ FOR DISCRETE HMMS

$$\hat{\pi}_i = \gamma_1(i) = \frac{\alpha_1(i)\beta_1(i)}{\sum_{i=1}^N \alpha_t(i)\beta_t(i)}$$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i) \beta_t(i)}$$

$$\hat{b}_{jk} = \frac{\sum_{t=1}^T \gamma_t(j) \chi_{[O_t=v_k]}}{\sum_{t=1}^T \gamma_t(j)} = \frac{\sum_{t=1}^T \alpha_t(j) \beta_t(j) \chi_{[O_t=v_k]}}{\sum_{t=1}^T \alpha_t(j) \beta_t(j)}$$

# HMM SUMMARY

Why are Hidden Markov Models successful ?

- speech variabilities are treated in a unified statistical framework
- efficient algorithms for training: need for a transcribed sample ( $\approx 2.5 \cdot 10^6$  parameters  $\Rightarrow \approx 200$  hours of speech)
- efficient algorithms for decoding
- synthesis capabilities (e.g.: phone HMM  $\rightarrow$  word HMM):
  - $\Rightarrow$  small HMM inventory: reliable parameter estimation
  - $\Rightarrow$  unlimited vocabulary

# LANGUAGE MODELING

- finite state grammars:
  - (very) small vocabulary
  - form filling, name dialing
  - command & control
- *stochastic n-gram language models:*
  - large vocabulary systems
  - dialogue systems

# TO KNOW WHAT WAS SAID WITHOUT LISTENING

1	The	are	<b>to</b>	know	the	issues	necessary
2	This	will	...	have	this	problems	data
3	One	the	...	understand	these	<b>the</b>	information
4	Two	would	...	do	problems	...	above
5	A	also	...	get	any	...	other
6	Three	do	...	the	a	...	time
7	Please	<b>need</b>		use	problem	...	people
8	In	...	...	provide	them	...	operators
9	<b>We</b>	...	...	insert	<b>all</b>	...	tools
...	...	...	...	...	...	...	...
98	...	...	...	<b>resolve</b>	...	...	old
...	...	...	...	...	...	...	...
641	...	...	...	...	...	...	<b>important</b>

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

# TO KNOW WHAT WAS SAID WITHOUT LISTENING

1	role	and	<b>the</b>	<b>next</b>	be	meeting
2	thing	from	...	...	<b>two</b>	months
3	that	in	...	...	...	years
4	to	to	...	...	...	meetings
5	contact	are	...	...	...	to
6	parts	with	...	...	...	week
7	point	were	...	...	...	<b>days</b>
8	for	requiring	...	...	...	...
9	<b>issues</b>	still	...	...	...	...
...	...	...	...	...	...	...
...	...	...	...	...	...	...
65	...	would	...	...	...	...
66	...	<b>within</b>	...	...	...	...

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

# N-GRAM LANGUAGE MODELS

- we need the a priori probability:

$$\begin{aligned} P(\mathbf{w}) &= P(w_1) \cdot P(w_2|w_1) \cdot \dots \cdot P(w_N|w_1, \dots, w_{N-1}) \\ &= P(w_1) \cdot \prod_{i=2}^N P(w_i|w_1, \dots, w_{i-1}) \end{aligned}$$

example:  $|\mathcal{V}| = 20000$ ,  $N = 6$ :  $\Rightarrow 6.4 \cdot 10^{25}$  probabilities

- equivalence classes:

$w_i$  depends on a limited history  $\Phi(w_1, \dots, w_{i-1})$ :

$$P(\mathbf{w}) = P(w_1) \cdot \prod_{i=2}^N P(w_i|\Phi(w_1, \dots, w_{i-1}))$$

# N-GRAM LANGUAGE MODELS — EQUIVALENCE CLASSES

- unigram LM: all histories are equivalent:

$$P(\boldsymbol{w}) = \prod_{i=1}^N P(w_i)$$

- bigram LM: histories are equivalent, if they end in the same word:

$$P(\boldsymbol{w}) = \prod_{i=1}^N P(w_i | w_{i-1})$$

- trigram LM: histories are equivalent, if they end in the same two words:

$$P(\boldsymbol{w}) = \prod_{i=1}^N P(w_i | w_{i-2}, w_{i-1})$$

example:  $|\mathcal{V}| = 20000 \Rightarrow 8 \cdot 10^{12}$  trigrams

# N-GRAM LANGUAGE MODELS

## *categorical n-gram models (class LM):*

- vocabulary as a union of pairwise disjoint classes:
  - *parts of speech*: nouns, articles, verbs, ...
  - *semantic categories*: digit, name of city, currency
  - minimize word error rate or *perplexity*

example: 512 classes  $\Rightarrow \approx 1.34 \cdot 10^8$  trigrams

## *mixture n-gram models:*

- given k different n-gram LMs, we compute:

$$P(\mathbf{w}) = \sum_{i=1}^k (\omega_i \cdot P_i(\mathbf{w}))$$

# LANGUAGE MODELS — TRAINING

- compute relative n-gram frequencies

$$f(w_i|w_{i-2}, w_{i-1}), f(w_i|w_{i-1}), f(w_i)$$

from *task dependent* training corpus

- many valid trigrams are not observed in training corpora:  
 $P(\mathbf{w}) = 0, \Rightarrow$  misrecognition !!

- smoothing:

$$P(w_i|w_{i-2}, w_{i-1}) = \delta_0 + \delta_1 \cdot f(w_i) + \delta_2 \cdot f(w_i|w_{i-1}) + \delta_3 \cdot f(w_i|w_{i-2}, w_{i-1})$$

# LANGUAGE MODEL — SUMMARY

Why are stochastic n-gram LMs successful?

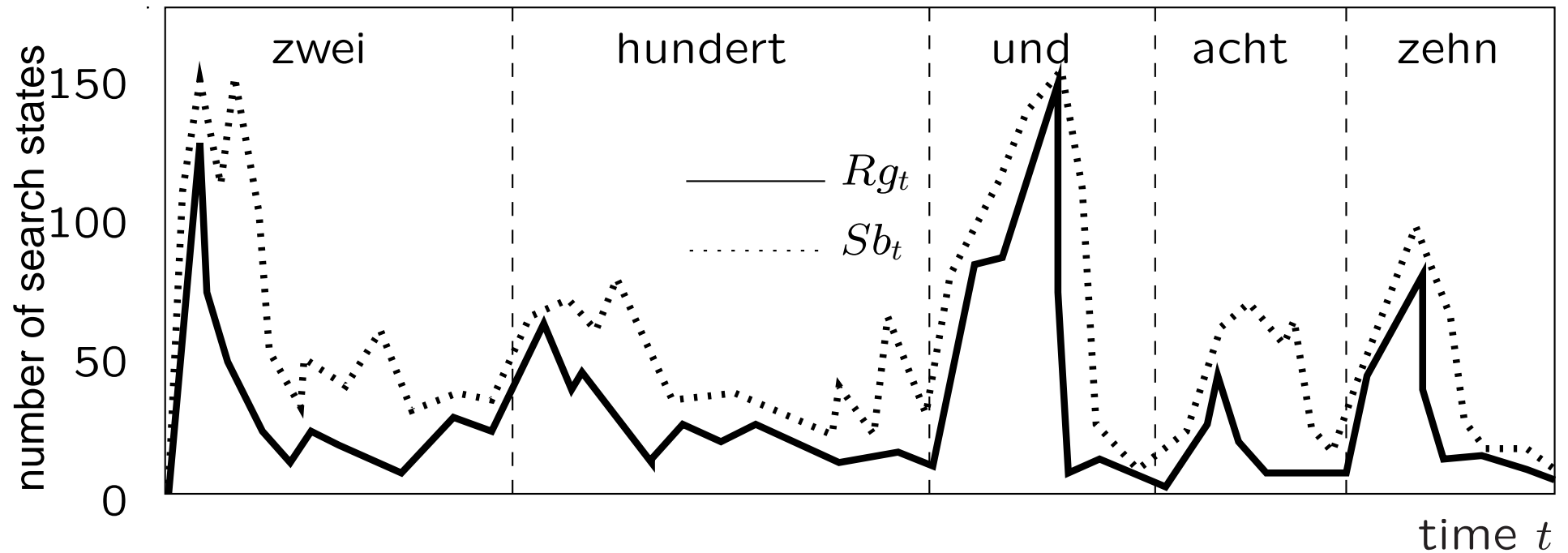
- training: no manual acquisition of rules or classes
- flexibility: (class) LMs are well suited for new domains
- soft decisions: spontaneous speech does not obey formal grammars
- efficiency: impose strong restrictions on search space

# SEARCH

- goal is to find
$$\hat{w} = \mathit{argmax}_w (P(w) \cdot P(O|w))$$
in an exponentially growing search space !!
- two efficient approaches:
  - *synchronous* search: Viterbi algorithm;  
reduced search effort by *beam search*
  - *asynchronous* search: stack decoding ( $A^*$  algorithm);  
reduced search effort by *acoustic fast match*  
tree-structured vocabulary  
context independent, simplified phone models

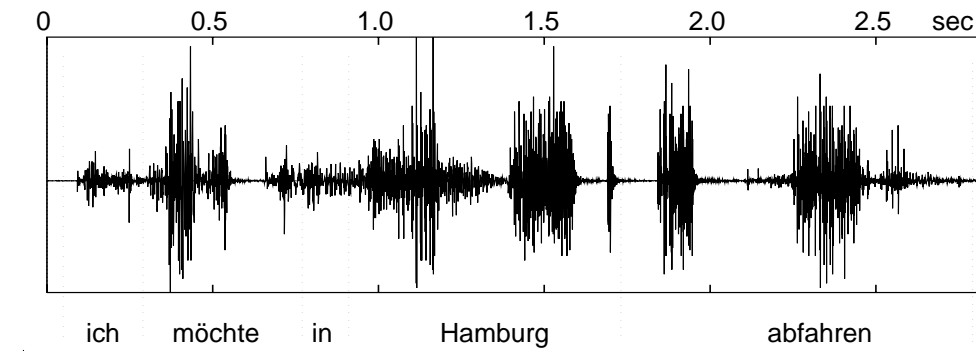
# BEAM SEARCH

$$O_t = \{i \mid \vartheta_t(i) \geq B_0 \cdot \Lambda_t\} \quad \text{with} \quad \Lambda_t = \max_j \vartheta_t(j)$$

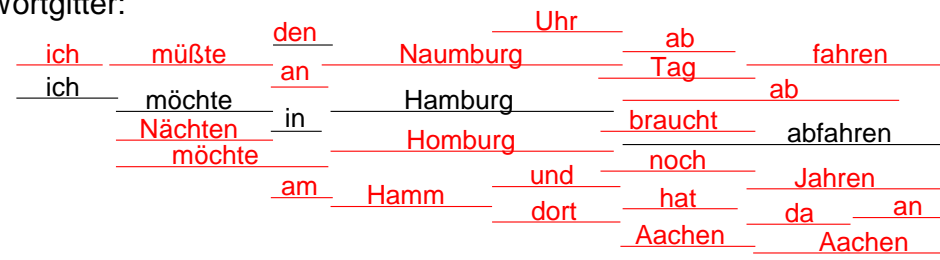


# INTERFACE TO SPEECH UNDERSTANDING: WORD CHAIN/LATTICE/GRAPH

Sprachsignal:



Wortgitter:



Wortgraph:

