

# Tecnologie per il Riconoscimento Automatico del Parlato

*Fabio Brugnara*

*ITC-irst - Istituto per la Ricerca Scientifica e Tecnologica*

Trento

`brugnara@itc.it`

`www.itc.it`

# L'Obiettivo e il Problema

Estrarre informazione linguistica da informazione sonora

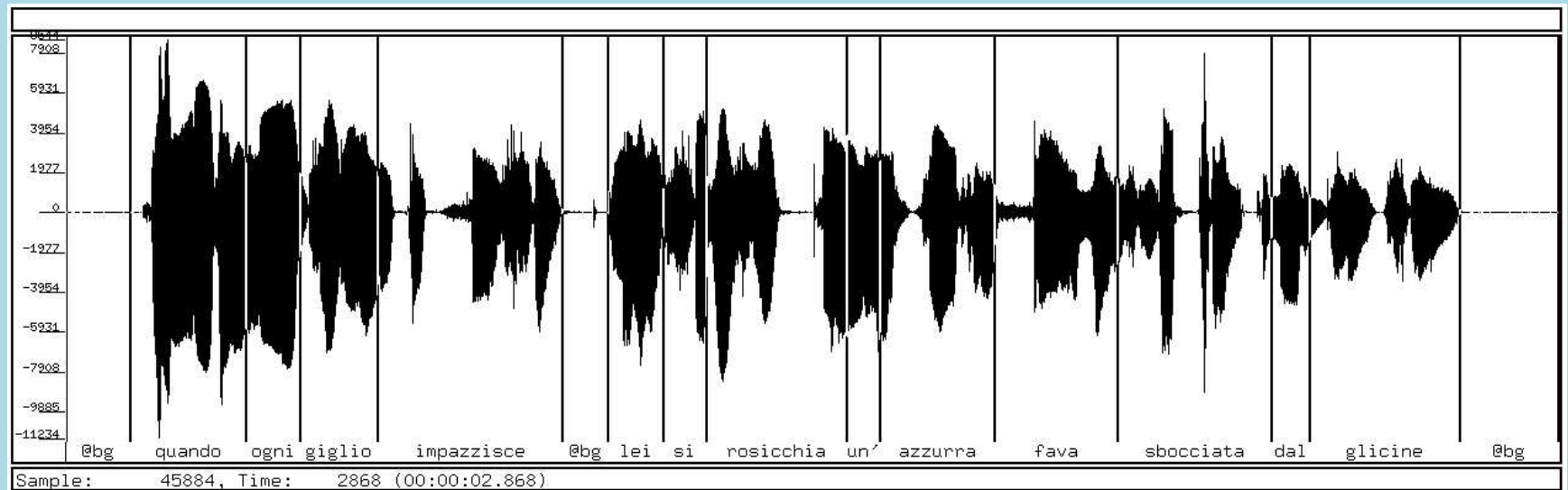
La corrispondenza fra i due spazi è *altamente ambigua*.

La realizzazione sonora di un messaggio linguistico è influenzata da:

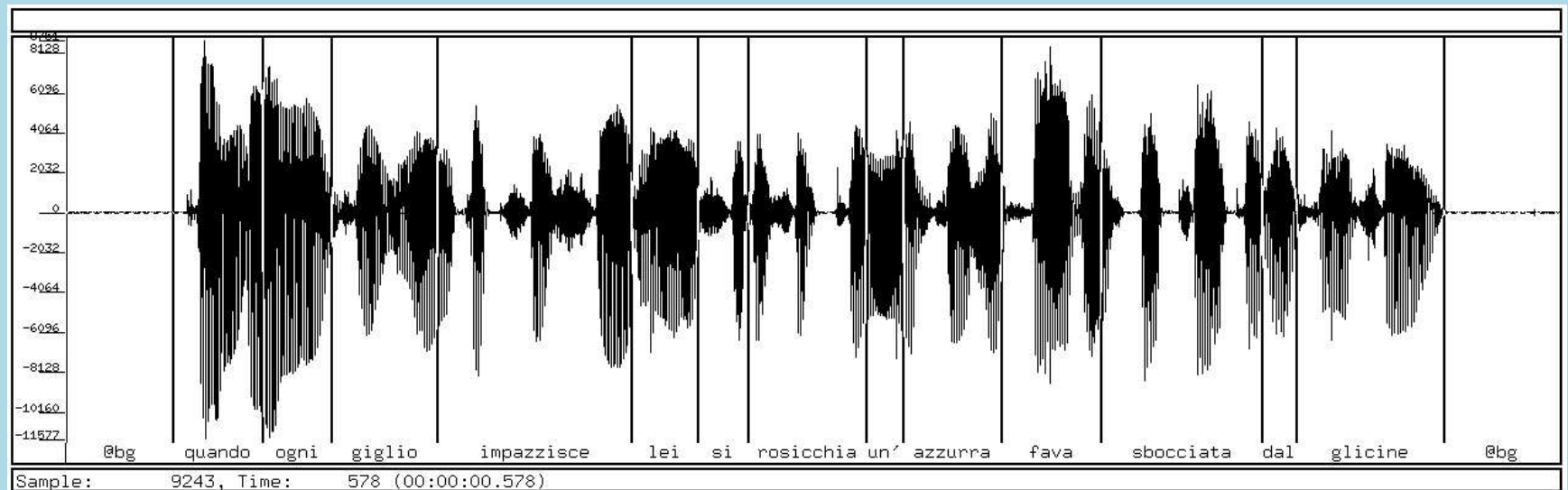
- parlatore (sesso, età, proprietà di pronuncia, stato emotivo, modalità di eloquio, ...)
- ambiente (presenza di rumore, riverbero, ...)
- canale audio (qualità di registrazione, trasmissione, ...)

# Two utterances of the same sentence: signal

female

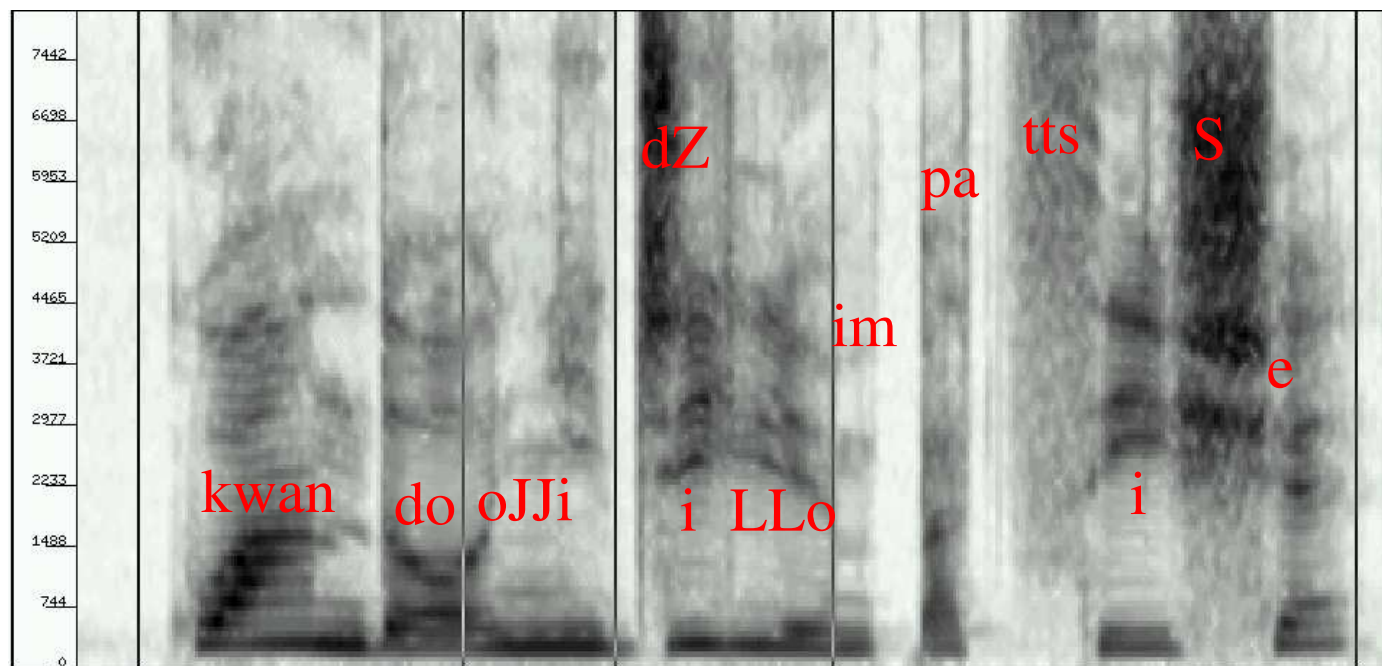


male

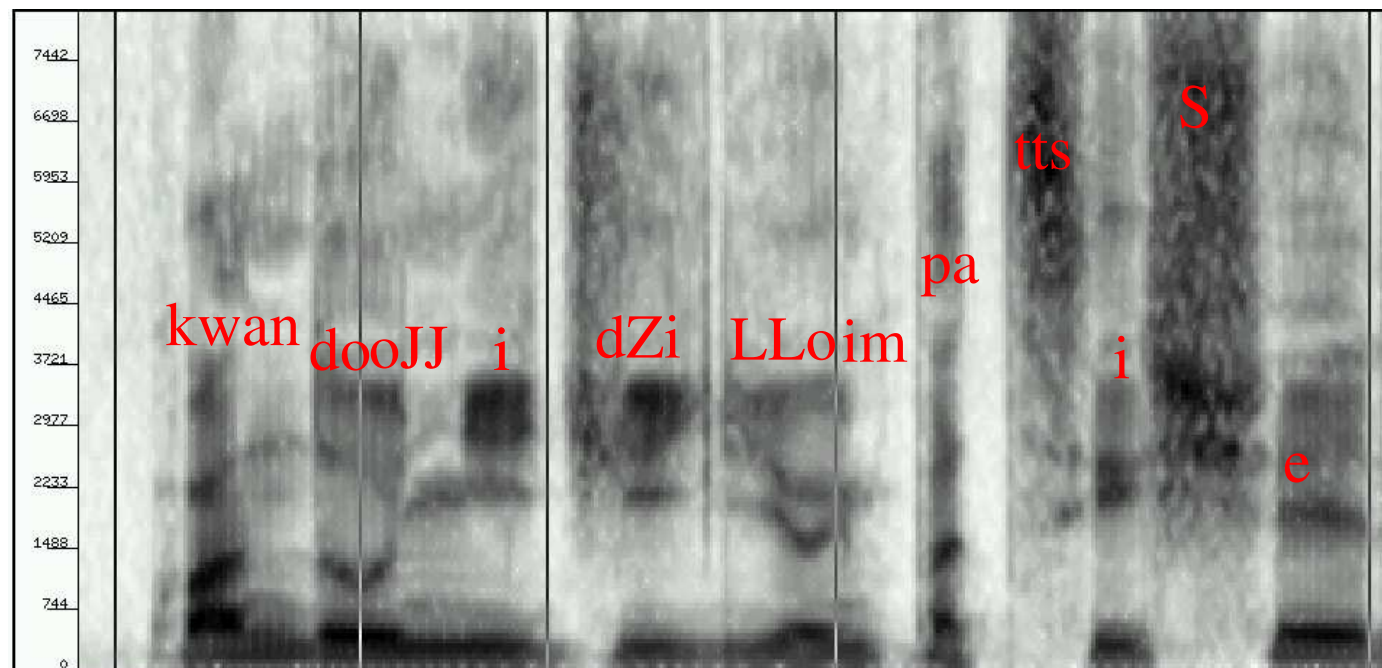


# Two utterances of the same sentence: spectrum

female



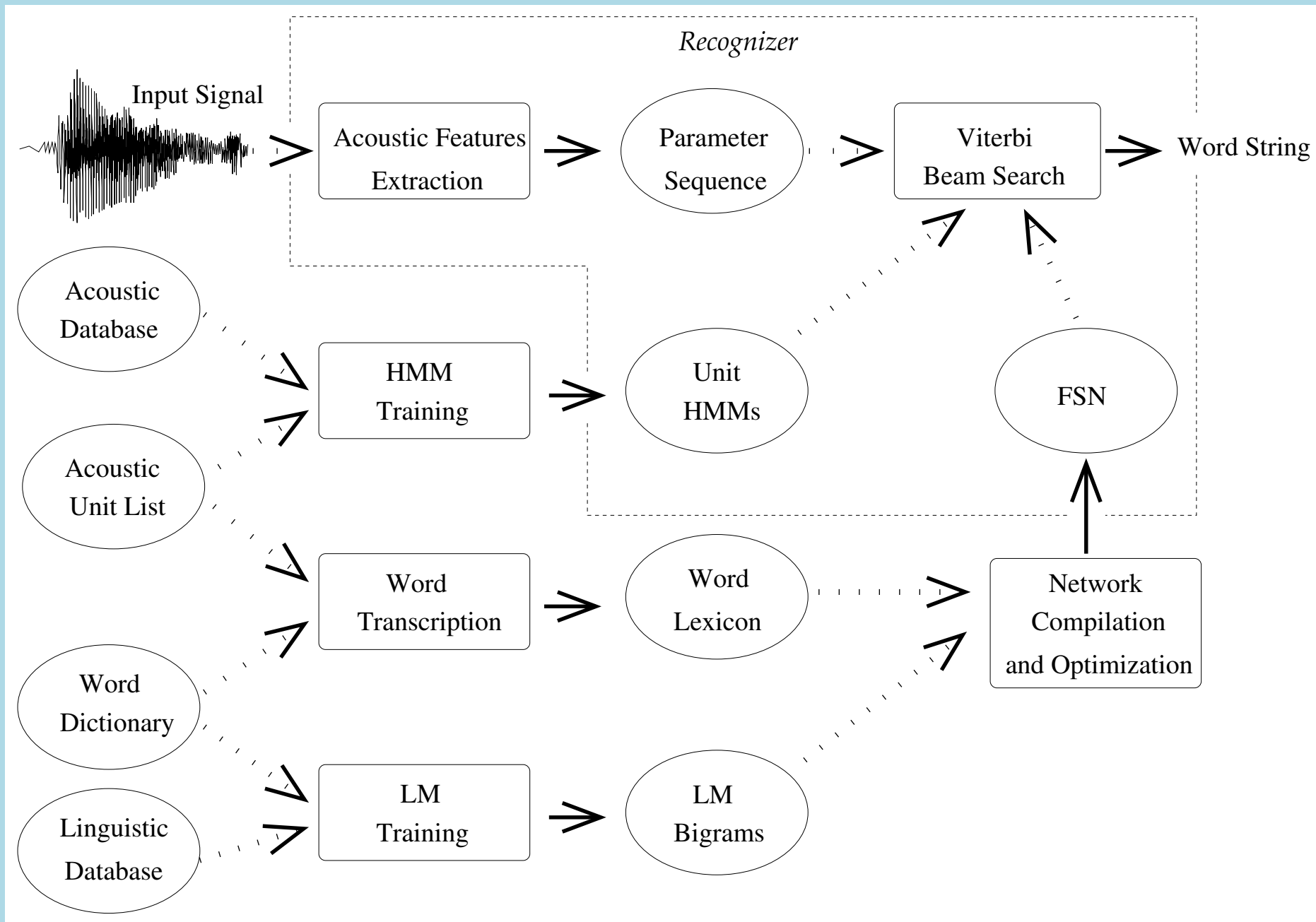
male



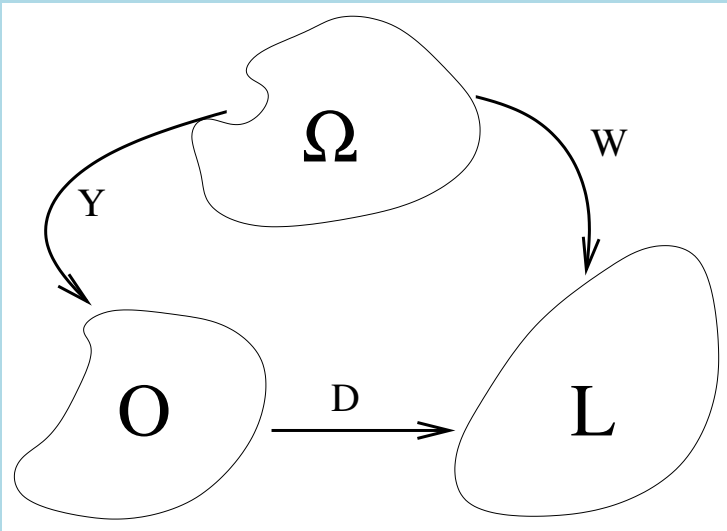
# Sommario

- Architettura
- Approccio statistico
- Parametri acustici
- Modelli di Markov Nascosti: definizioni, FB, Viterbi
- Unità acustiche
- Modelli context-dependent e tying
- Modelli del linguaggio: a regole, a  $n$ -grammi
- Composizione di modelli
- Reti di riconoscimento
- Beam-search
- Reti ad albero
- Esempio: un sistema di trascrizione di notiziari

# Componenti di un riconoscitore



# Approccio statistico



$Y, W$ : rappresentano lo stesso fenomeno, la produzione di messaggi linguistici.

Possiamo osservare solo  $Y$ , e cercare  $D$  in modo da massimizzare:

$$\Pr[D \circ Y = W]$$

$$\begin{aligned}\Pr[F \circ Y = W] &= \int \Pr[Y = y, F \circ Y = W] dy \\ &= \int \Pr[Y = y, W = F(y)] dy\end{aligned}$$

## Criterio di Bayes

$$\begin{aligned}D(y) &\triangleq \operatorname{argmax}_w \Pr[Y = y, W = w] \\ &= \operatorname{argmax}_w \Pr[W = w] \Pr[Y = y | W = w]\end{aligned}$$

## Definizione di un Task

- Lessico
- Linguaggio
- Osservazioni

## Componenti del Riconoscitore

- Calcolo osservazioni
- Modelli di base
- Descrizione del linguaggio
- Algoritmo di decodifica

# Calcolo Osservazioni

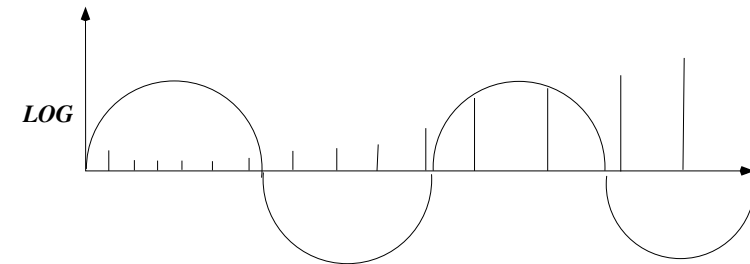
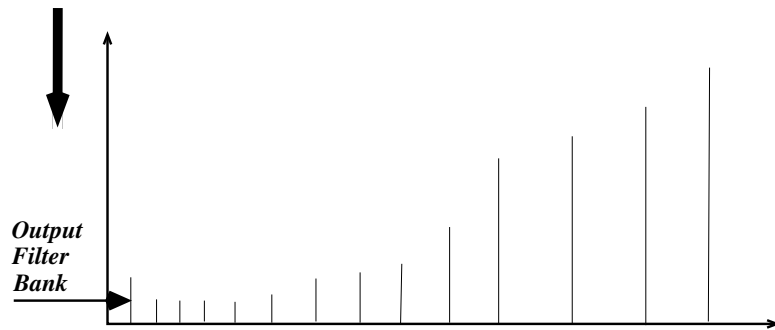
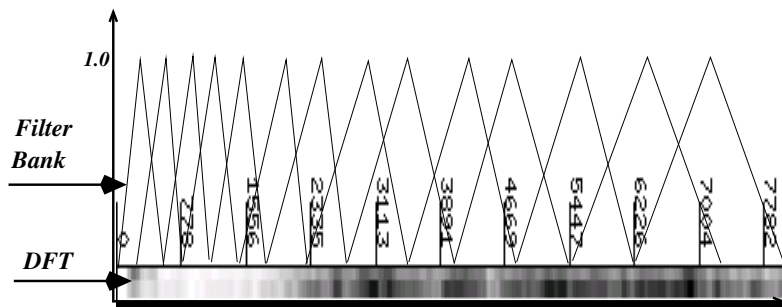
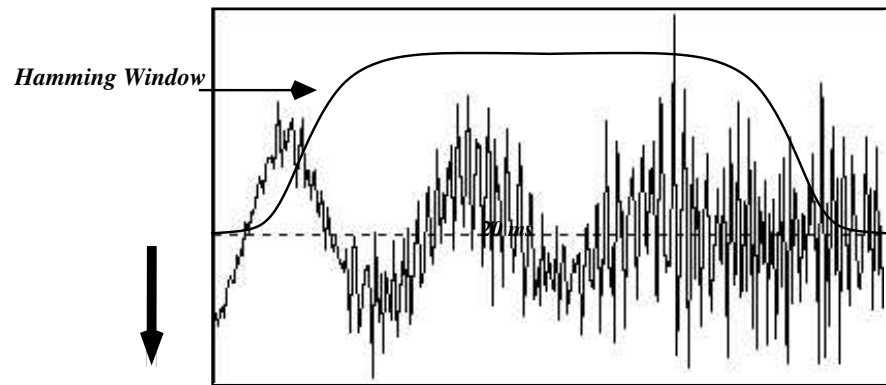
Una osservazione va descritta con un sequenza di vettori reali che contenga informazione utile e non informazione ridondante.

## Es. di **Features Extraction**

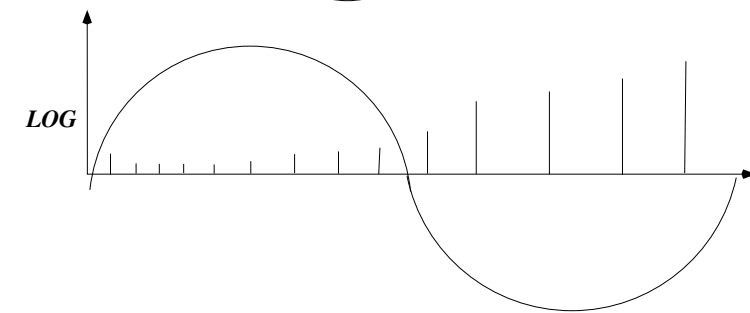
1. blocking
2. finestatura
3. analisi spettrale
4. banco di filtri
5. trasformazione coefficienti
6. aggiunta parametri dinamici

più comune: *MEL scaled cepstral coefficients*

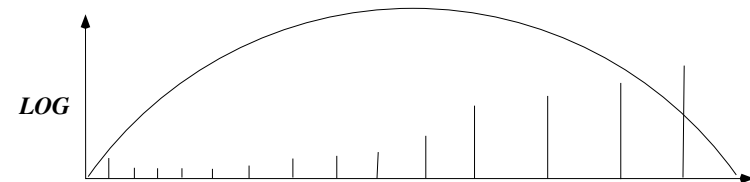
# Feature Extraction



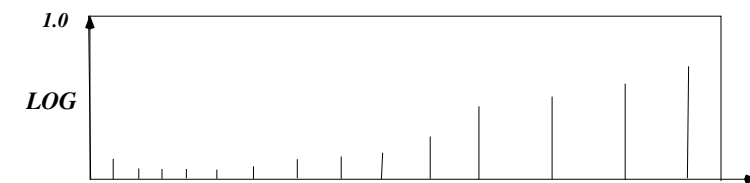
MFCC 3



MFCC 2



MFCC 1



Energy

# Hidden Markov Model

Un Modello di Markov Nascosto (HMM) e' composto da una coppia  $(\mathbf{I}, \mathbf{X})$  di processi stocastici che soddisfano le seguenti ipotesi:

$$i, j \in \mathcal{I} = \mathbb{N}_S, y \in \mathcal{Y}, t \in \mathbb{N}$$

*Markov:*

$$\Pr[\mathbf{I}_t = i | \mathbf{I}_0^{t-1} = \mathbf{i}_0^{t-1}] = \Pr[\mathbf{I}_t = i | \mathbf{I}_{t-1} = \mathbf{i}_{t-1}]$$

*Output  
Independence:*

$$\Pr[\mathbf{Y}_t = y | \mathbf{Y}_0^{t-1} = \mathbf{y}_0^{t-1}, \mathbf{I}_0^T = \mathbf{i}_0^T] = \Pr[\mathbf{Y}_t = y | \mathbf{I}_{t-1}^t = \mathbf{i}_{t-1}^t]$$

## Parametri

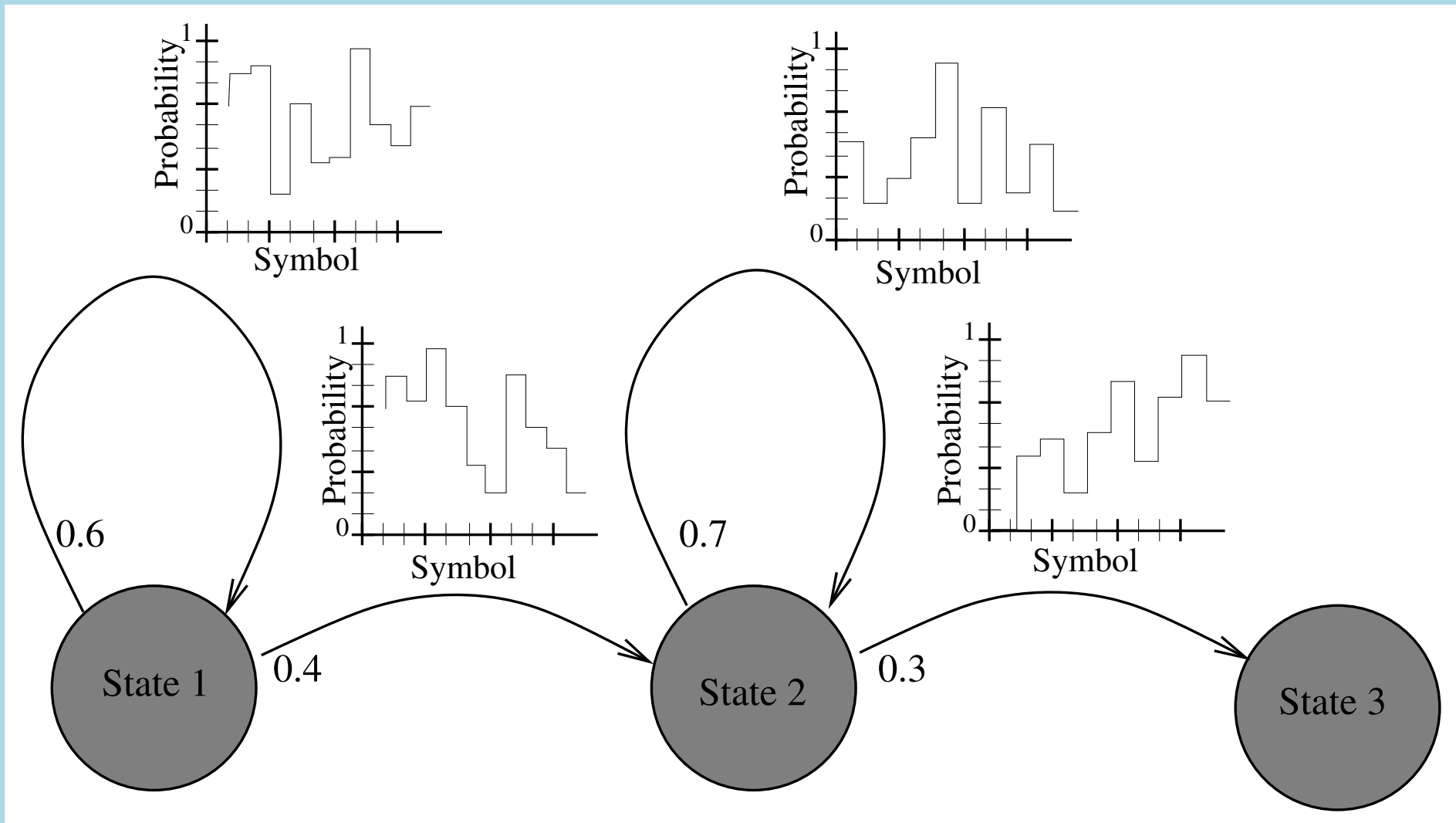
$$\lambda \triangleq (\pi, A, B)$$

$$\pi_i \triangleq \Pr[\mathbf{I}_0 = i]$$

$$a_{ij} \triangleq \Pr[\mathbf{I}_t = j | \mathbf{I}_{t-1} = i]$$

$$b_{ij}(y) \triangleq \Pr[\mathbf{Y}_t = y | \mathbf{I}_{t-1} = i, \mathbf{I}_t = j]$$

# Rappresentazione a grafo di un HMM



# Tipi di densità sulle osservazioni

*Discreta:*

$$y \in \mathbb{N}_Q, b \in \mathbb{R}_+^Q, \sum_{i=1}^Q b_i = 1, \quad b(y) = b_y$$

*Gaussiana:*

$$y \in \mathbb{R}, \mu \in \mathbb{R}, \sigma \in \mathbb{R}_+, \quad b(y) = \mathcal{N}(\mu, \sigma; y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y-\mu)^2}{2\sigma^2}}$$

La più usata, *Mistura di Gaussiane:*

$$y \in \mathbb{R}, \mu_k \in \mathbb{R}, \sigma_k \in \mathbb{R}_+, \sum_{k=1}^K w_k = 1, \quad b(y) = \sum_{k=1}^K w_k \mathcal{N}(\mu_k, \sigma_k; y)$$

In realtà si usano le equivalenti a più dimensioni.

# Calcolo di probabilità

$$\Pr[\mathbf{y}_1^T] = \sum_{\mathbf{i}_0^T} \Pr[\mathbf{y}_1^T | \mathbf{i}_0^T] \Pr[\mathbf{i}_0^T]$$

$$\Pr[\mathbf{i}_0^T] = \Pr[i_0] \prod_{t=1}^T \Pr[i_t | \mathbf{i}_0^{t-1}] \underset{\substack{\uparrow \\ \text{(Markov)}}}{=} \Pr[i_0] \prod_{t=1}^T \Pr[i_t | i_{t-1}] = \pi_{i_0} \prod_{t=1}^T a_{i_{t-1}i_t}$$

$$\Pr[\mathbf{y}_1^T | \mathbf{i}_0^T] = \prod_{t=1}^T \Pr[y_t | \mathbf{y}_1^{t-1}, \mathbf{i}_0^T] \underset{\substack{\uparrow \\ \text{(Out.Ind.)}}}{=} \prod_{t=1}^T \Pr[y_t | \mathbf{i}_{t-1}^t] = \prod_{t=1}^T b_{i_{t-1}i_t}(y_t)$$

$$\Pr[\mathbf{y}_1^T] = \sum_{\mathbf{i}_0^T} \pi_{i_0} \prod_{t=1}^T a_{i_{t-1}i_t} b_{i_{t-1}i_t}(y_t) \quad \text{Non è applicabile direttamente!}$$

# Coefficienti "Forward" e "Backward"

$$\begin{aligned}\alpha_t(\mathbf{y}_1^T, i) &\triangleq \Pr[\mathbf{I}_t = i, \mathbf{y}_1^t] \\ \beta_t(\mathbf{y}_1^T, i) &\triangleq \Pr[\mathbf{y}_{t+1}^T | \mathbf{I}_t = i] \\ \gamma_t(\mathbf{y}_1^T, i, j) &\triangleq \Pr[\mathbf{I}_{t-1} = i, \mathbf{I}_t = j | \mathbf{y}_1^T] \\ &= \frac{\alpha_{t-1}(\mathbf{y}_1^T, i) a_{ij} b_{ij}(y_t) \beta_t(\mathbf{y}_1^T, j)}{\Pr[\mathbf{y}_1^T]}\end{aligned}$$

La probabilità totale si può ricavare da:

$$\begin{aligned}\Pr[\mathbf{y}_1^T] &= \sum_{i \in \mathcal{I}} \alpha_T(\mathbf{y}_1^T, i) \\ &= \sum_{i \in \mathcal{I}} \pi_i \beta_0(\mathbf{y}_1^T, i)\end{aligned}$$

## Calcolo dei coefficienti $\alpha$

$$\alpha_t(\mathbf{y}_1^T, i) \triangleq \Pr[\mathbf{y}_1^t, \mathbf{I}_t = i]$$

$$\begin{aligned}\alpha_t(\mathbf{y}_1^T, i) &= \\ &= \Pr[\mathbf{I}_t = i, \mathbf{y}_1^t] \\ &= \sum_{j \in \mathcal{I}} \Pr[\mathbf{I}_{t-1} = j, \mathbf{y}_1^t, \mathbf{I}_t = i] \\ &= \sum_{j \in \mathcal{I}} \Pr[\mathbf{I}_{t-1} = j, \mathbf{y}_1^{t-1}] \Pr[\mathbf{I}_t = i | \mathbf{I}_{t-1} = j] \Pr[\mathbf{Y}_t = y_t | \mathbf{I}_{t-1} = j, \mathbf{I}_t = i] \\ &= \sum_{j \in \mathcal{I}} \alpha_{t-1}(\mathbf{y}_1^T, j) a_{ji} b_{ji}(y_t)\end{aligned}$$

$$\alpha_t(\mathbf{y}_1^T, i) = \begin{cases} \pi_i, & t = 0 \\ \sum_{j \in \mathcal{I}} \alpha_{t-1}(\mathbf{y}_1^T, j) a_{ji} b_{ji}(y_t), & t = 1 \dots T \end{cases}$$

## Calcolo dei coefficienti $\beta$

$$\beta_t(\mathbf{y}_1^T, i) \triangleq \Pr[\mathbf{y}_{t+1}^T | \mathbf{I}_t = i]$$

$$\begin{aligned}\beta_t(\mathbf{y}_1^T, i) &= \\ &= \Pr[\mathbf{y}_{t+1}^T | \mathbf{I}_t = i] \\ &= \sum_{j \in \mathcal{I}} \Pr[\mathbf{I}_{t+1} = j, \mathbf{y}_{t+1}^T | \mathbf{I}_t = i] \\ &= \sum_{j \in \mathcal{I}} \Pr[\mathbf{I}_{t+1} = j | \mathbf{I}_t = i] \Pr[\mathbf{Y}_{t+1} = y_{t+1} | \mathbf{I}_t = i, \mathbf{I}_{t+1} = j] \Pr[\mathbf{y}_{t+2}^T | \mathbf{I}_{t+1} = j] \\ &= \sum_{j \in \mathcal{I}} \beta_{t+1}(\mathbf{y}_1^T, j) a_{ij} b_{ij}(y_{t+1})\end{aligned}$$

$$\beta_t(\mathbf{y}_1^T, i) = \begin{cases} 1, & t = T \\ \sum_{j \in \mathcal{I}} \beta_{t+1}(\mathbf{y}_1^T, j) a_{ij} b_{ij}(y_{t+1}), & t = T - 1 \dots 0 \end{cases}$$

# Coefficienti "Viterbi"

$$\hat{\Pr}[\mathbf{y}_1^T] \triangleq \max_{\mathbf{i}_0^T} \Pr[\mathbf{y}_1^T, \mathbf{i}_0^T]$$

$$\nu_t(\mathbf{y}_1^T, i) \triangleq \max_{\mathbf{i}_0^{t-1}} \Pr[\mathbf{i}_0^{t-1}, \mathbf{y}_1^t, \mathbf{I}_t = i] \quad \Rightarrow \quad \hat{\Pr}[\mathbf{y}_1^T] = \max_{i \in \mathcal{I}} \nu_T(\mathbf{y}_1^T, i)$$

$$\nu_t(\mathbf{y}_1^T, i) =$$

$$= \max_{\mathbf{i}_0^{t-2}} \max_{j \in \mathcal{I}} \Pr[\mathbf{i}_0^{t-2}, \mathbf{y}_1^t, \mathbf{I}_{t-1} = j, \mathbf{I}_t = i]$$

$$= \max_{j \in \mathcal{I}} \max_{\mathbf{i}_0^{t-2}} \Pr[\mathbf{i}_0^{t-2}, \mathbf{y}_1^{t-1}, \mathbf{I}_{t-1} = j] \Pr[\mathbf{I}_t = i | \mathbf{I}_{t-1} = j] \Pr[\mathbf{Y}_t = y_t | \mathbf{I}_{t-1} = j, \mathbf{I}_t = i]$$

$$= \max_{j \in \mathcal{I}} \nu_{t-1}(\mathbf{y}_1^T, j) a_{ji} b_{ji}(y_t)$$

$$\nu_t(\mathbf{y}_1^T, i) = \begin{cases} \pi_i, & t = 0 \\ \max_{j \in \mathcal{I}} \nu_{t-1}(\mathbf{y}_1^T, j) a_{ji} b_{ji}(y_t), & t = 1 \dots T \end{cases}$$

## Backtracking (“decodifica”)

Ricostruisce il singolo cammino “nascosto” più probabile, data l’osservazione.

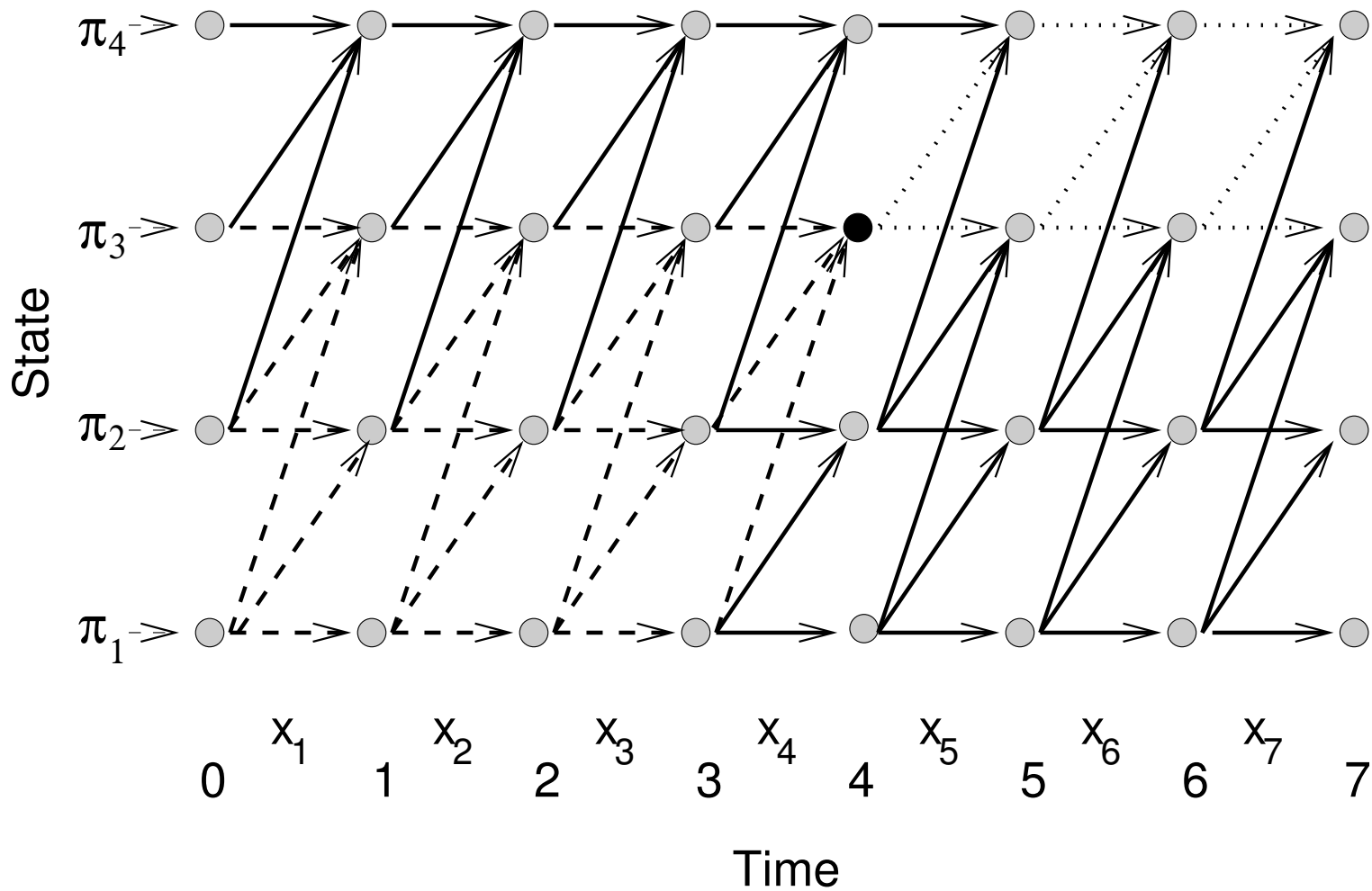
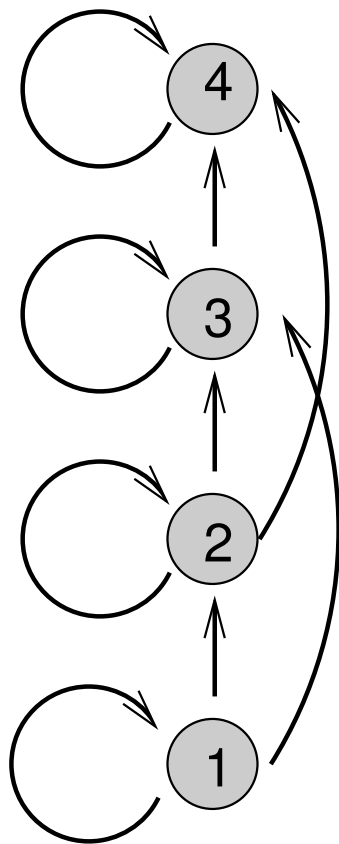
Richiede la memorizzazione di *backpointers*.

$$\begin{aligned}\nu_t(\mathbf{y}_1^T, i) &= \max_{j \in \mathcal{I}} \nu_{t-1}(\mathbf{y}_1^T, j) a_{ji} b_{ji}(y_t) \\ \phi_t(\mathbf{y}_1^T, i) &= \operatorname{argmax}_{j \in \mathcal{I}} \nu_{t-1}(\mathbf{y}_1^T, j) a_{ji} b_{ji}(y_t)\end{aligned}$$

$$\hat{i}_t = \begin{cases} \operatorname{argmax}_{i \in \mathcal{I}} \nu_T(\mathbf{y}_1^T, i), & t = T \\ \phi_{t+1}(\mathbf{y}_1^T, \hat{i}_{t+1}), & t = T - 1 \dots 0 \end{cases}$$

$$\hat{\Pr}[\mathbf{y}_1^T] = \Pr[\mathbf{y}_1^T, \hat{\mathbf{i}}_0^T]$$

# Principale struttura di calcolo: il Trellis



# L'algoritmo di Viterbi

```
function viterbi
begin
   $t := 0$ ;
  for  $i := 1$  to  $N$  do begin
     $\nu_t(i) := \pi_i$ ;  $\phi_t(i) := nil$ ;  $\tau_t(i) := nil$ 
  end
  expand_empty_trans;
  for  $t := 1$  to  $T$  do begin
    for  $i := 1$  to  $N$  do  $\nu_t(i) := 0$ ;
    expand_full_trans;
    expand_empty_trans
  end
   $\hat{\nu} := \nu_T(1)$ ;
  for  $i := 2$  to  $N$  do begin
    if  $\hat{\nu} < \nu_T(i)$  then  $\hat{\nu} := \nu_T(i)$ ;
  end
  return  $\hat{\nu}$ 
end
```

```
procedure expand_full_trans
begin
  for  $i := 1$  to  $N$  do begin
    for  $j := 1$  to  $N$  do begin
       $\hat{\nu} := \nu_{t-1}(i)a_{ij}b_{ij}(x_t)$ ;
      if  $\nu_t(j) < \hat{\nu}$  then begin
         $\nu_t(j) := \hat{\nu}$ ;  $\phi_t(j) := i$ ;  $\tau_t(j) := t - 1$ 
      end
    end
  end
end
```

---

$\mathbf{x}_1^T$  is the observation sequence

$N$  is the number of states

$[\nu_t(i)]$  stores partial path scores

$[\phi_t(i)]$  stores backtracking info

$[\tau_t(i)]$  tells if last transition is empty or not

# L'algoritmo di Viterbi

```
procedure expand_empty_trans
begin
  for  $i := 1$  to  $N$  do push( $i$ );
   $i := \text{pop}$ ;
  while  $i \neq \text{nil}$  do begin
    for  $j := 1$  to  $N$  do begin
       $\hat{\nu} := \nu_t(i)a_{ij}^\epsilon$ ;
      if  $\nu_t(j) < \hat{\nu}$  then begin
         $\nu_t(j) := \hat{\nu}$ ;  $\phi_t(j) := i$ ;  $\tau_t(j) := t$ 
        push_unique( $j$ )
      end
    end
  end
   $i := \text{pop}$ 
end
```

---

**push** (**push\_unique**): adds state  $i$  on the stack  
(only if it is not already in)

```
function backtrack
begin
   $t := T$ ;  $j := 1$ ;
  for  $i := 2$  to  $N$  do begin
    if  $\nu_t(j) < \nu_t(i)$  then  $j := i$ ;
  end
   $l := -1$ ;
  while  $j \neq \text{nil}$  do begin
     $l := l + 1$ ;  $\hat{i}_l := j$ ;  $\hat{t}_l := t$ ;
     $j := \phi_t(\hat{i}_l)$ ;  $t := \tau_t(\hat{i}_l)$ ;
  end
  reverse( $\hat{\mathbf{i}}_0^l$ ); reverse( $\hat{\mathbf{t}}_0^l$ );
  return ( $\hat{\mathbf{i}}_0^l, \hat{\mathbf{t}}_0^l$ )
end
```

---

**reverse**: inverts the order of an array

# Addestramento con Maximum Likelihood Estimation

$$\lambda \triangleq (\pi, A, B)$$

Per sequenza singola:

$$\tilde{\lambda} = \operatorname{argmax}_{\lambda} \Pr_{\lambda}[\mathbf{y}_1^T]$$

Per sequenze multiple:

$$\tilde{\lambda} = \operatorname{argmax}_{\lambda} \prod_{\mathbf{y} \in \mathcal{L}} \Pr_{\lambda}[\mathbf{y}]$$

Si usa **Baum-Welch** (alias **EM**), su **grandi** quantità di dati.

# Baum-Welch

Usa una *growth transformation* sullo spazio dei parametri  $\lambda$ .

$$P(\lambda) \triangleq \Pr_{\lambda}[\mathbf{y}_1^T] = \sum_{\mathbf{i}_0^T} \Pr_{\lambda}[\mathbf{i}_0^T, \mathbf{y}_1^T]$$

$$Q(\lambda, \lambda') \triangleq \frac{1}{P(\lambda)} \sum_{\mathbf{i}_0^T} \Pr_{\lambda}[\mathbf{i}_0^T, \mathbf{y}_1^T] \log \Pr_{\lambda'}[\mathbf{i}_0^T, \mathbf{y}_1^T]$$

Vale (*Jensen*):

$$Q(\lambda, \lambda') \geq Q(\lambda, \lambda) \quad \Rightarrow \quad P(\lambda') > P(\lambda)$$

Quindi si sceglie: 
$$\lambda^{(n+1)} = \operatorname{argmax}_{\lambda} Q(\lambda^{(n)}, \lambda)$$

Incrementa la funzione obiettivo, e, con le debite ipotesi, converge ...

# Formule di ristima

$$a'_{ij} = \frac{\sum_{t=1}^T \gamma_t(\mathbf{y}_1^T, i, j)}{\sum_{h \in \mathcal{I}} \sum_{t=1}^T \gamma_t(\mathbf{y}_1^T, i, h)} \quad b'^y_{ij} = \frac{\sum_{t=1}^T \delta(y_t, y) \gamma_t(\mathbf{y}_1^T, i, j)}{\sum_{t=1}^T \gamma_t(\mathbf{y}_1^T, i, j)}$$

$$\mu'_{ij} = \frac{\sum_{t=1}^T \gamma_t(\mathbf{y}_1^T, i, j) y_t}{\sum_{t=1}^T \gamma_t(\mathbf{y}_1^T, i, j)} \quad \sigma'^2_{ij} = \frac{\sum_{t=1}^T \gamma_t(\mathbf{y}_1^T, i, j) y_t^2}{\sum_{t=1}^T \gamma_t(\mathbf{y}_1^T, i, j)} - \mu'^2_{ij}$$

# Unità acustiche

È necessario usare unità più elementari delle parole, per:

- disponibilità di esempi
- flessibilità nel lessico

I *fonemi* sono astrazioni di un numero limitato di suoni elementari con cui si possono comporre tutte le parole di una lingua.

- Fonemi indipendenti dal contesto (decine)

casa → /k/ /a/ /z/ /a/

- Allofoni dipendenti dal contesto (migliaia)

casa → /?\_k\_a/ /k\_a\_z/ /a\_z\_a/ /z\_a\_?/

Necessità della *trascrizione fonetica*, spesso costosa.

Problemi di *sparsità dei dati* in addestramento.

Ai bordi delle parole? Si può perdere l'univocità della corrispondenza  
parola → trascrizione.

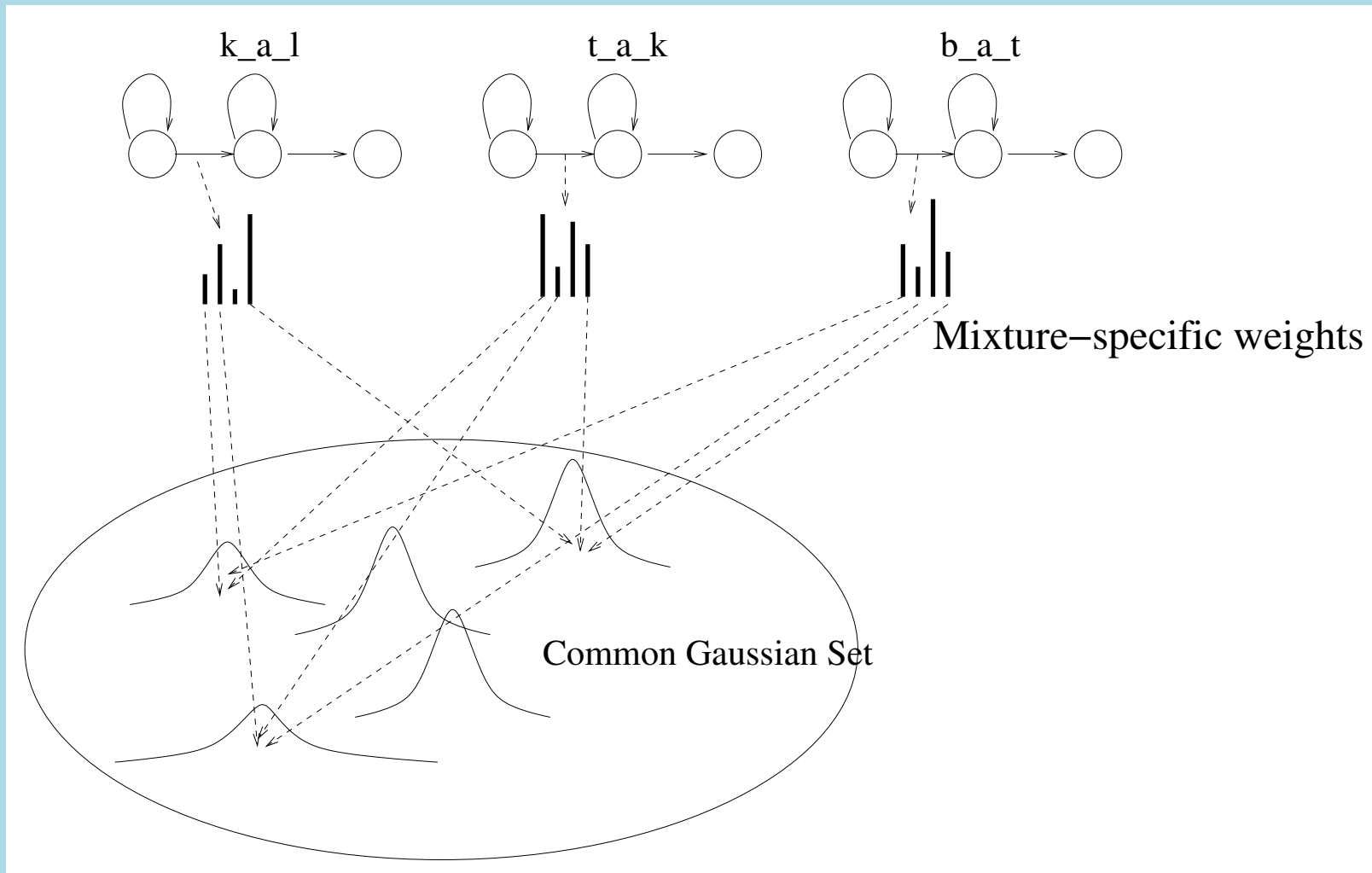
# Sparsità dei dati: Parameter Tying

Le unità context-dependent comportano un *vasto* insieme di parametri.

Molte unità sono relativamente *simili*, possono *condividere* parametri.

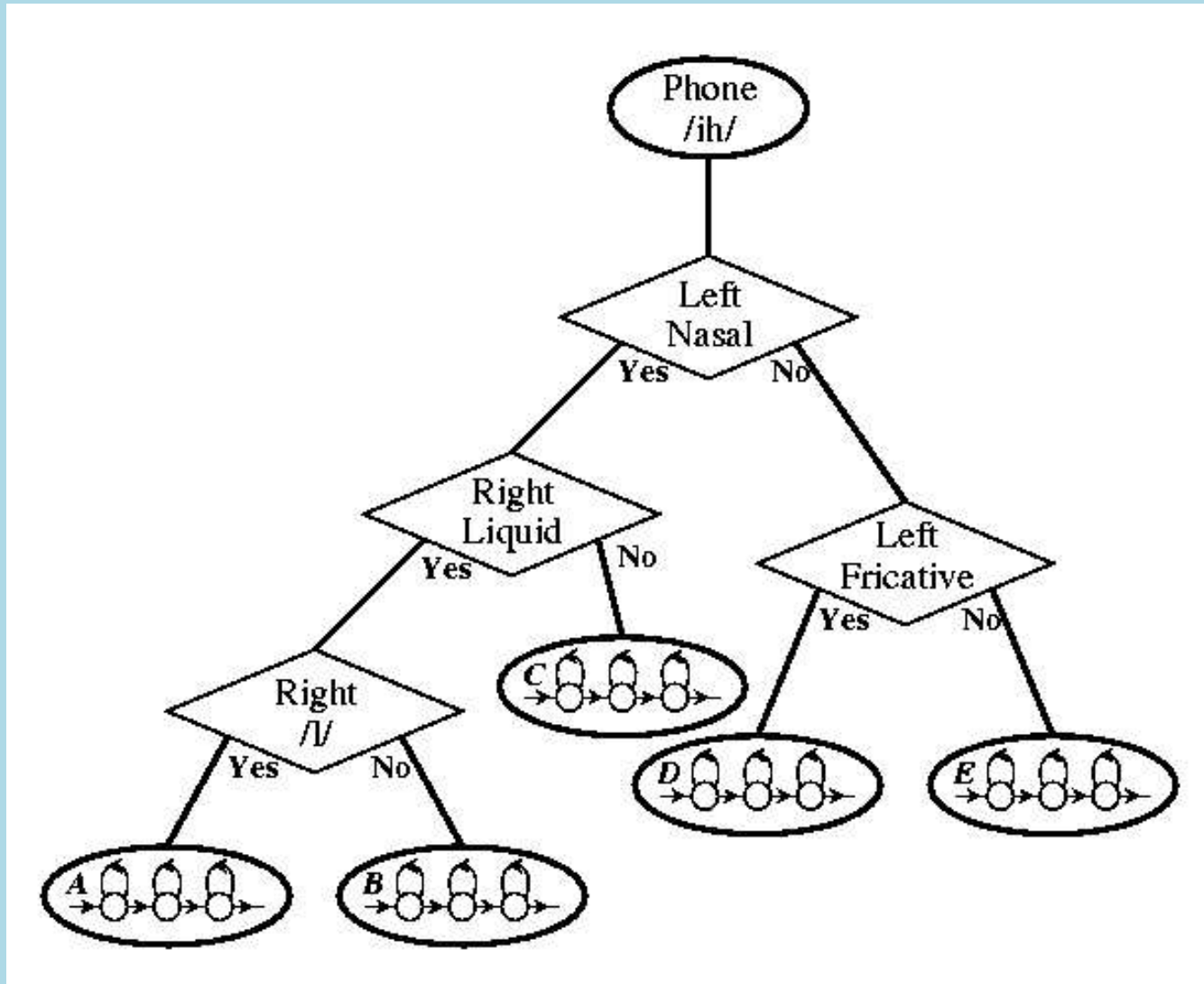
Esempio:

*Phonetically Tied Mixtures*

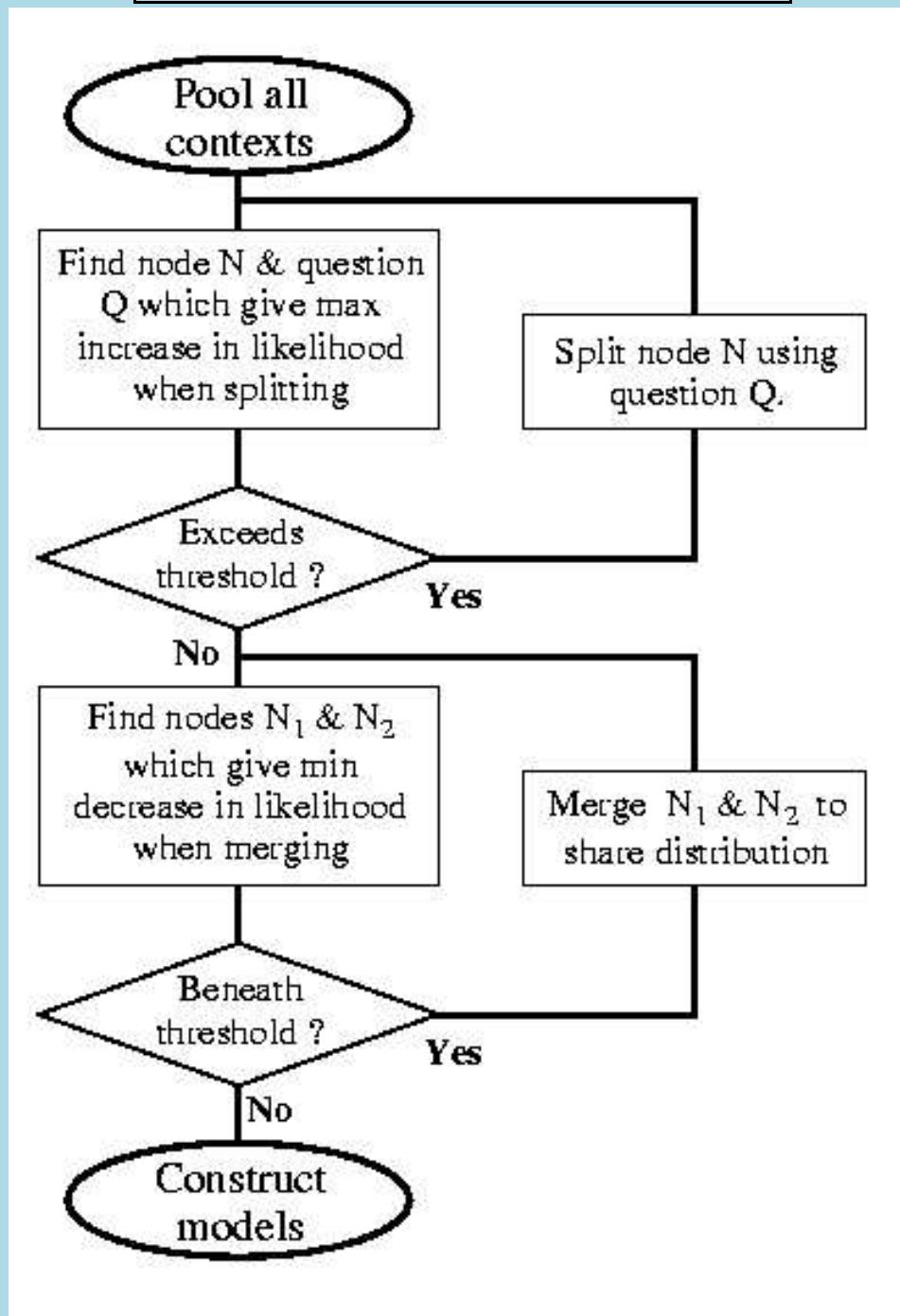


# Phonetic Decision Tree

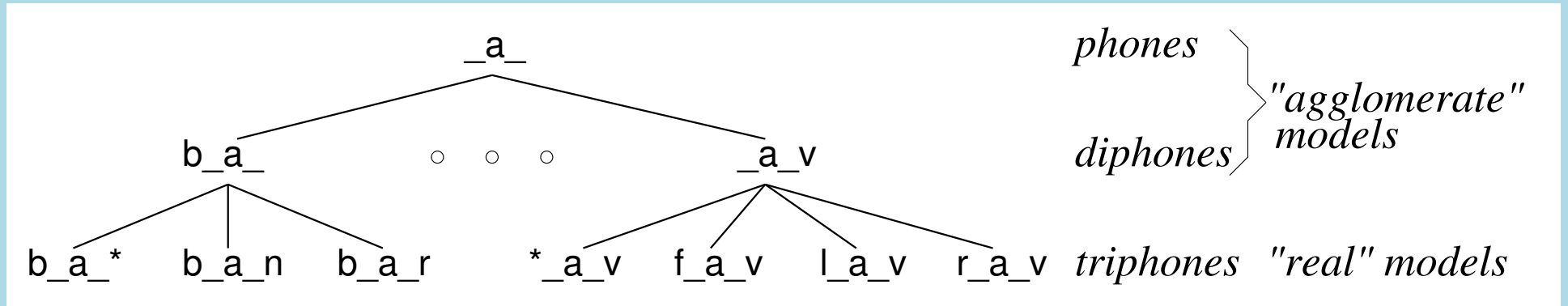
Costruiti con ML sulla base di un insieme di trifoni privo di tying.



# Costruzione di PDT



# Agglomerazione di Modelli



I modelli “agglomerati” sono costruiti sinteticamente usando le statistiche dei modelli dettagliati. Ad esempio, per le probabilità di transizione:

$$a_{i,j} = \frac{\gamma_M(i,j)}{\sum_{h=1}^S \gamma_M(i,h)}$$

dove

$$\gamma_M(i,j) \equiv \sum_{m \in \mathcal{T}(M)} \gamma_m(i,j)$$

# Modelli del linguaggio

Necessari per compensare l'imprecisione del modello acustico.

Vincoli *rigidi* o *elastici*, quindi modelli:

- *a regole (grammatiche)*: comandi, menu, espressioni comuni (date, numeri, ecc..). In genere grammatiche *regolari*.
- *statistici*: dettatura di testi, giornali, notiziari, ecc..

Una caratteristica importante per l'efficienza: *omogeneità* di rappresentazione con i modelli acustici.

# Esempio di grammatica a regole: Numeri da 1 a 1000

DIGIT2 ::= due | ... | nove

DIGIT ::= uno | DIGIT2

TEEN ::= dieci | undici | ... | diciannove

TEN ::= venti | trenta | ... | novanta

NUM2 ::= DIGIT | TEEN | TEN

NUM2 ::= TEN DIGIT

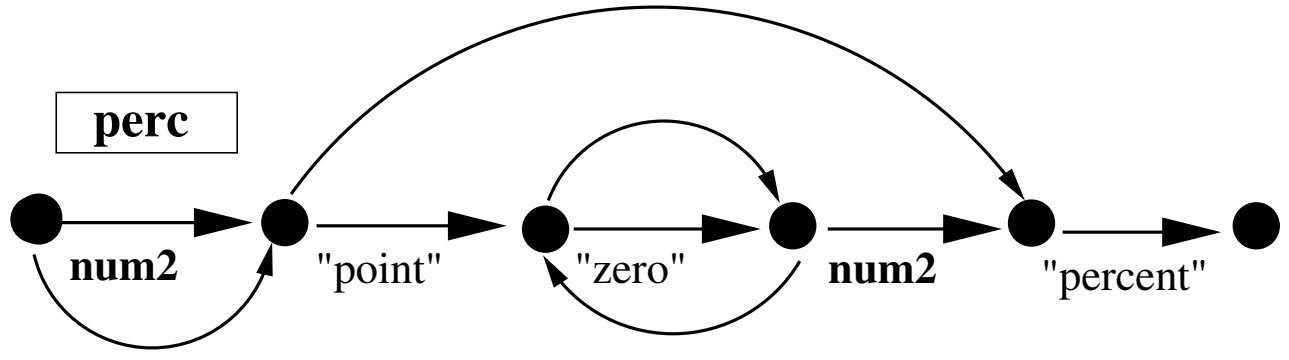
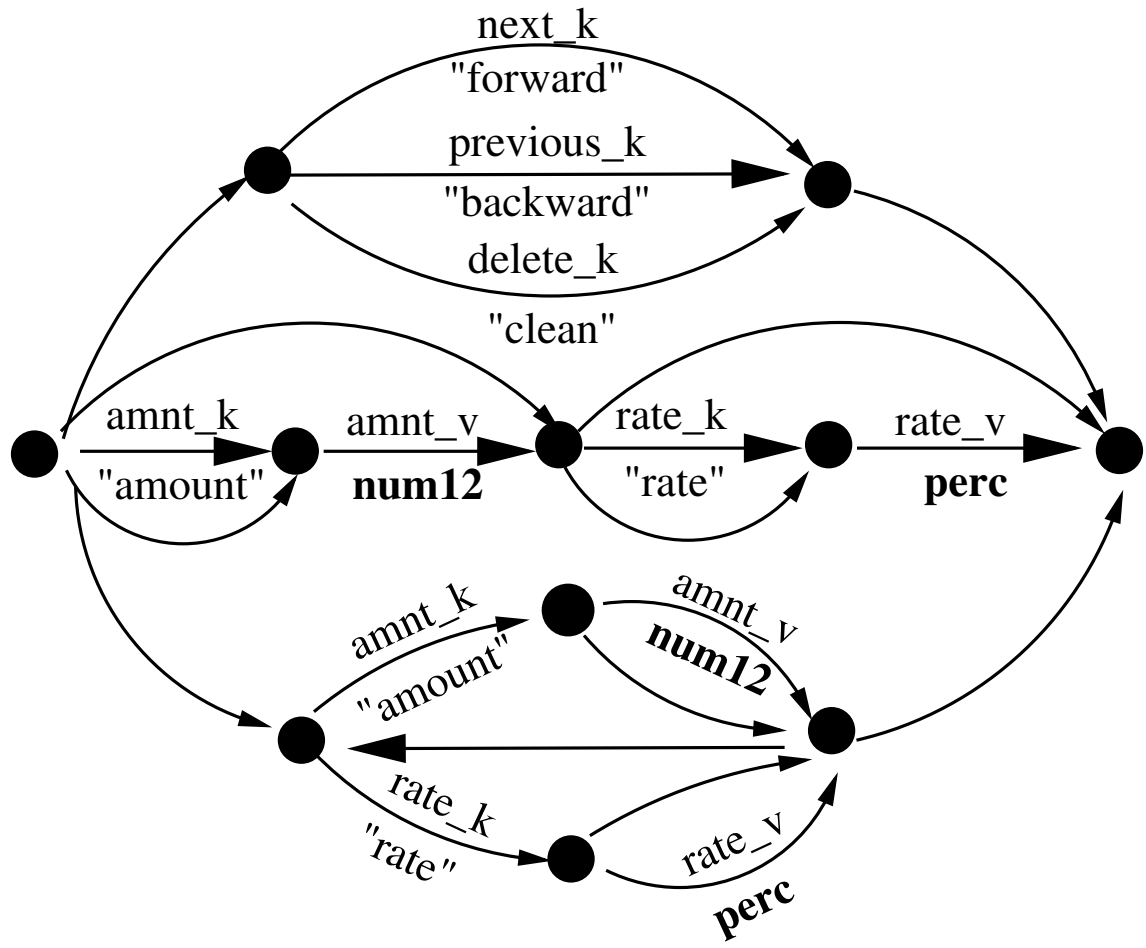
NUM3 ::= NUM2

NUM3 ::= cento NUM2 | DIGIT2 cento NUM2

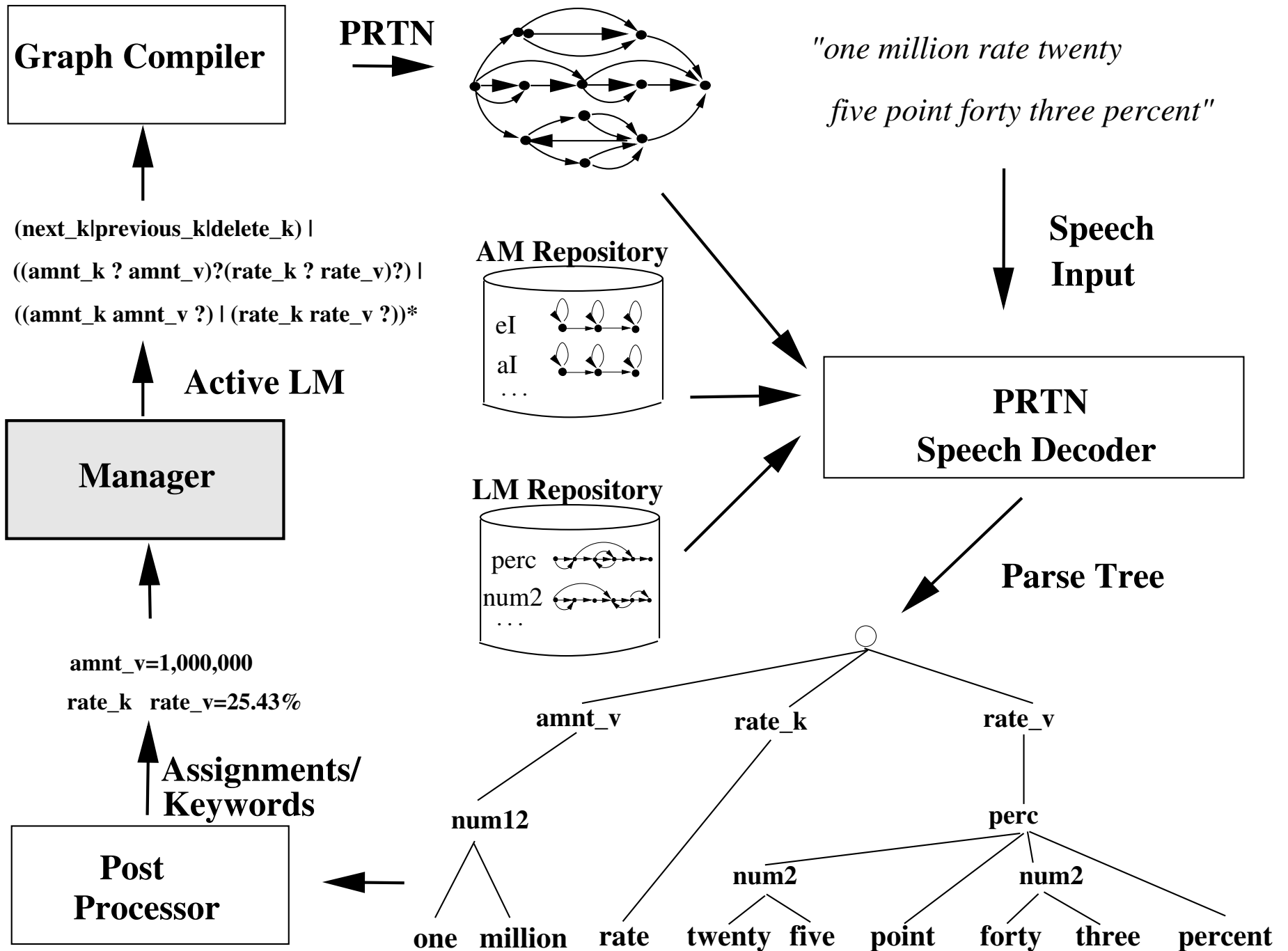
NUM3 ::= mille

Dizionario: 29 parole, Linguaggio: 1000 parole composte

# Recurrent Transition Networks



# Un sistema con linguaggio dipendente dallo stato



# Il più comune modello del linguaggio statistico: $n$ -grammi

$$\Pr[\mathbf{W}_1^T] = \prod_{t=1}^T \Pr[w_t | w_1 \dots w_{t-1}]$$

$$\Pr[\mathbf{W}_1^T] \approx \prod_{t=1}^T \Pr[w_t | w_{t-n+1} \dots w_{t-1}]$$

$n = 2 \rightarrow$  bigrammi (bigrams)

$$\Pr[\mathbf{W}_1^T] \approx \prod_{t=1}^T \Pr[w_t | w_{t-1}]$$

$n = 3 \rightarrow$  trigrammi (trigrams)

$$\Pr[\mathbf{W}_1^T] \approx \prod_{t=1}^T \Pr[w_t | w_{t-2} w_{t-1}]$$

Numero di parametri potenzialmente proibitivo!

# Sparsità dei dati: Discounting and Redistribution

Smoothing by *interpolation*:

$$\Pr(w|h) = fr^*(w|h) + \lambda(h) \Pr(w|h')$$

$$0 \leq fr^*(w|h) \leq fr(w|h) \equiv \frac{c(hw)}{c(h)}$$

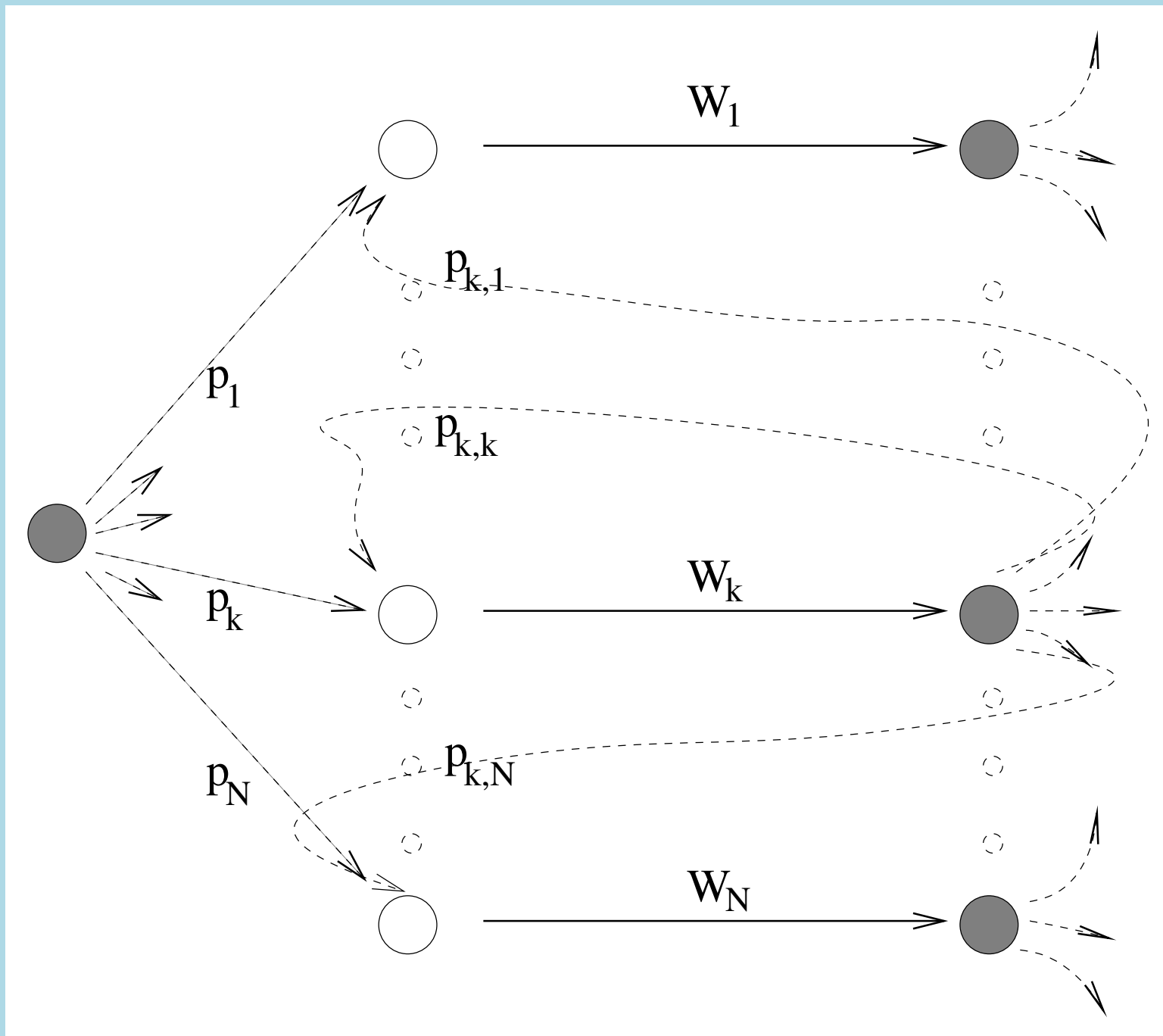
$$\lambda(h) = 1 - \sum_w fr^*(w|h)$$

E.g., with *shift-1* discounting:

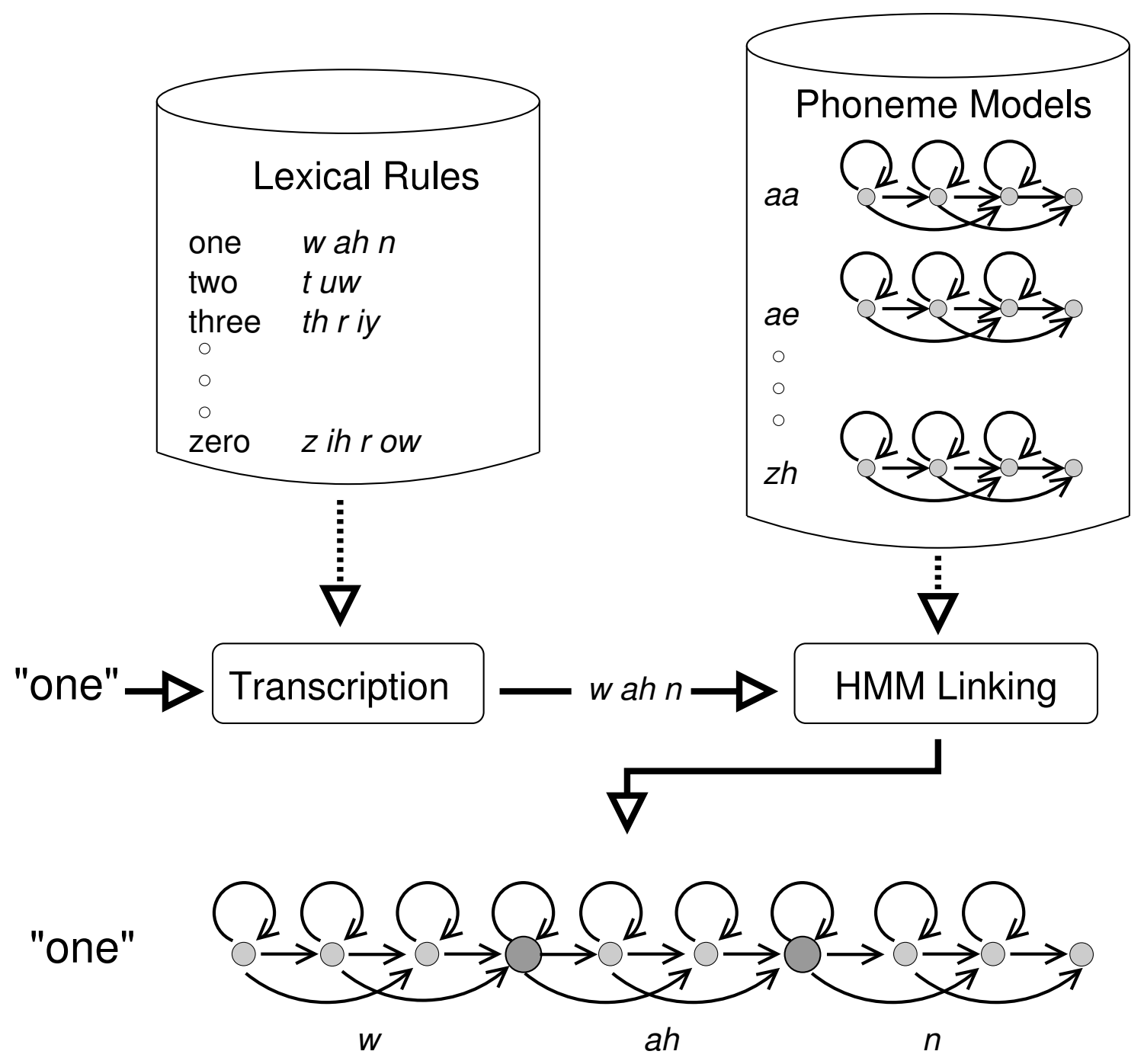
$$fr^*(w|h) = \max \left\{ \frac{c(hw) - 1}{c(h)}, 0 \right\}$$

$$\lambda(h) = \frac{|\text{succ}(h)|}{c(h)}$$

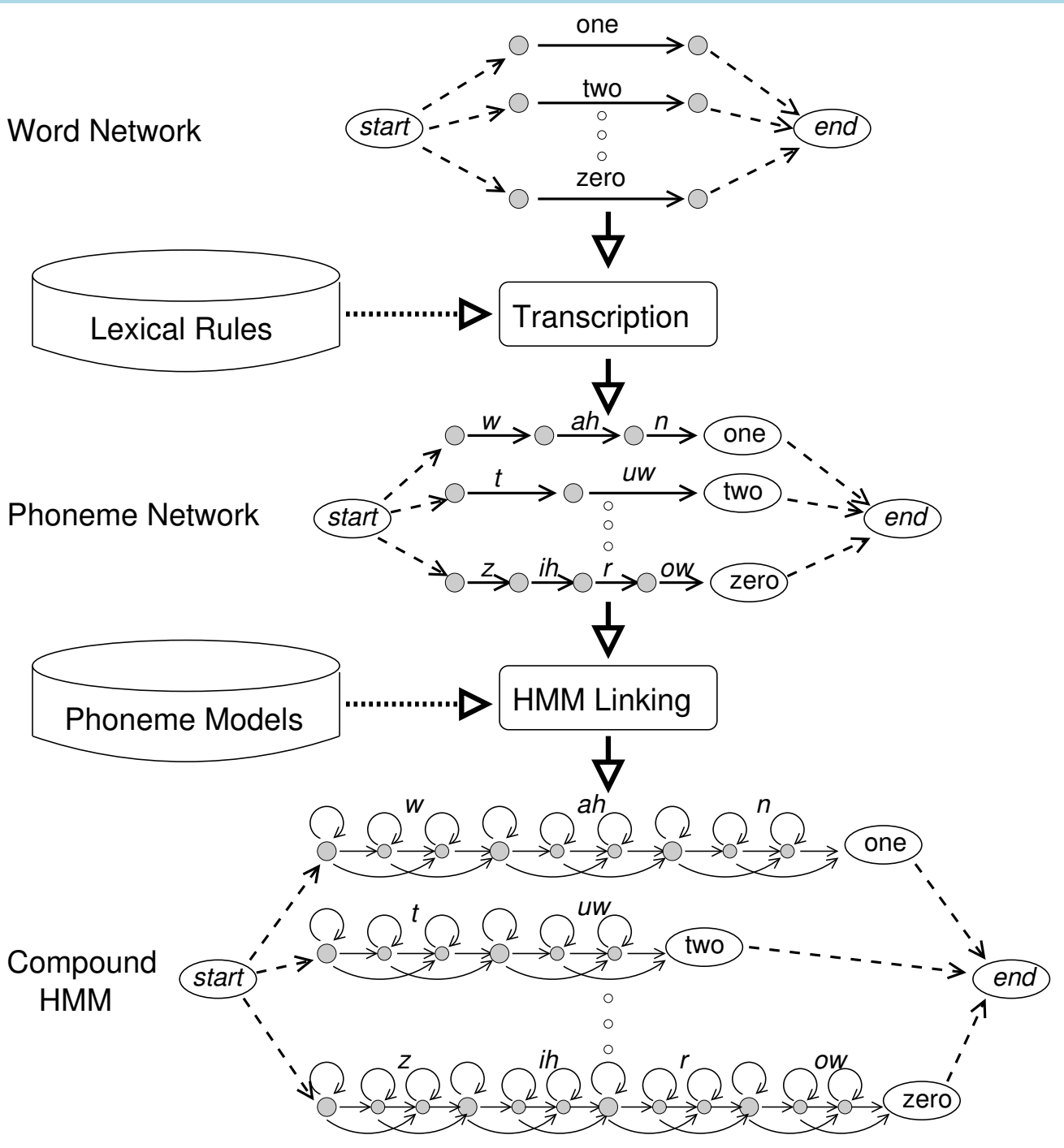
# Rappresentazione a grafo di un ML a bigrammi



# Composizione di modelli di parola



# Composizione di reti integrate



## Riconoscimento (decodifica)

Diventa la ricerca di un cammino a massima probabilità su un grafo: il trellis corrispondente ad un HMM composto.

L'algoritmo di base più usato è l'algoritmo di Viterbi.

Si mantiene però una distinzione fra stati *interni* ai modelli e stati della rete di unità.

Lo spazio di ricerca può essere **molto** grande, anche miliardi di stati ed archi.

Una esplorazione esaustiva è generalmente impossibile.

La rete può essere interamente compilata a priori, oppure espansa dinamicamente.

# Beam-Search

Il numero di stati significativi, ad un certo istante, può essere *ordini di grandezza* inferiore al numero di stati totali della rete.

Si cerca di rimuovere cammini “poco promettenti”.

Diversi tipi di *pruning*, spesso combinati:

- *Relative threshold pruning*: data una soglia relativa  $\theta$ , scarta uno stato  $i$  se

$$\nu_t(i) \leq \theta \max_j \nu_t(j)$$

- *Histogram pruning*: tiene al più un numero fissato di stati

- *Phone look-ahead pruning*: limita l'insieme dei modelli nelle prossime ipotesi

- *Output density pruning*: riduce l'insieme delle gaussiane considerate

## Beam-Search a soglia

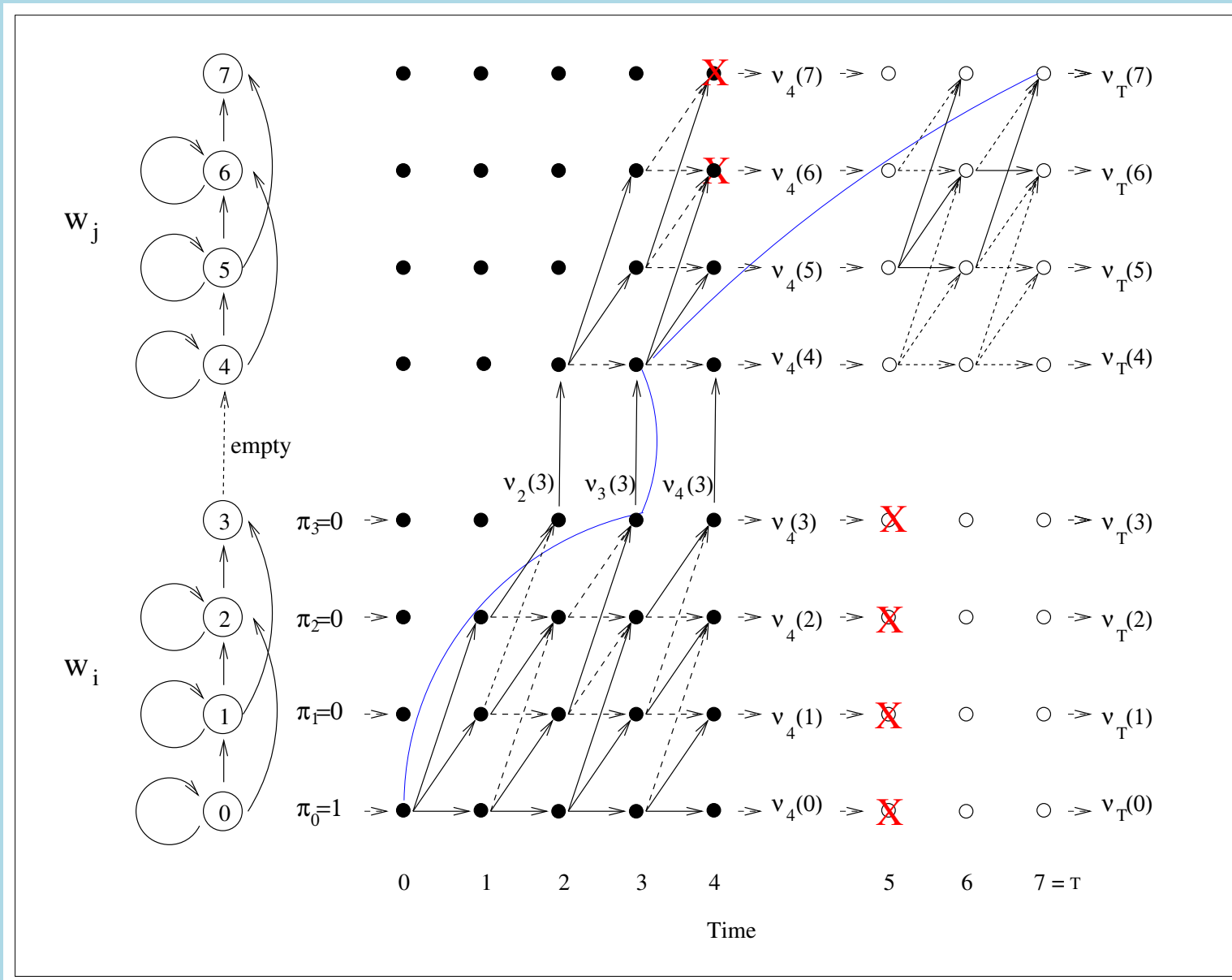
Per ciascun frame:

- Stabilisci la soglia di pruning: *best-width*
- Rimuovi gli stati al di sotto della soglia
- Propaga i cammini interni ai modelli, e calcola *best* per il prossimo frame
- Combina le probabilità sulla rete, tenendo conto delle probabilità di uscita dei modelli e delle probabilità di transizione sulla rete

Trova lo stato finale con massima probabilità

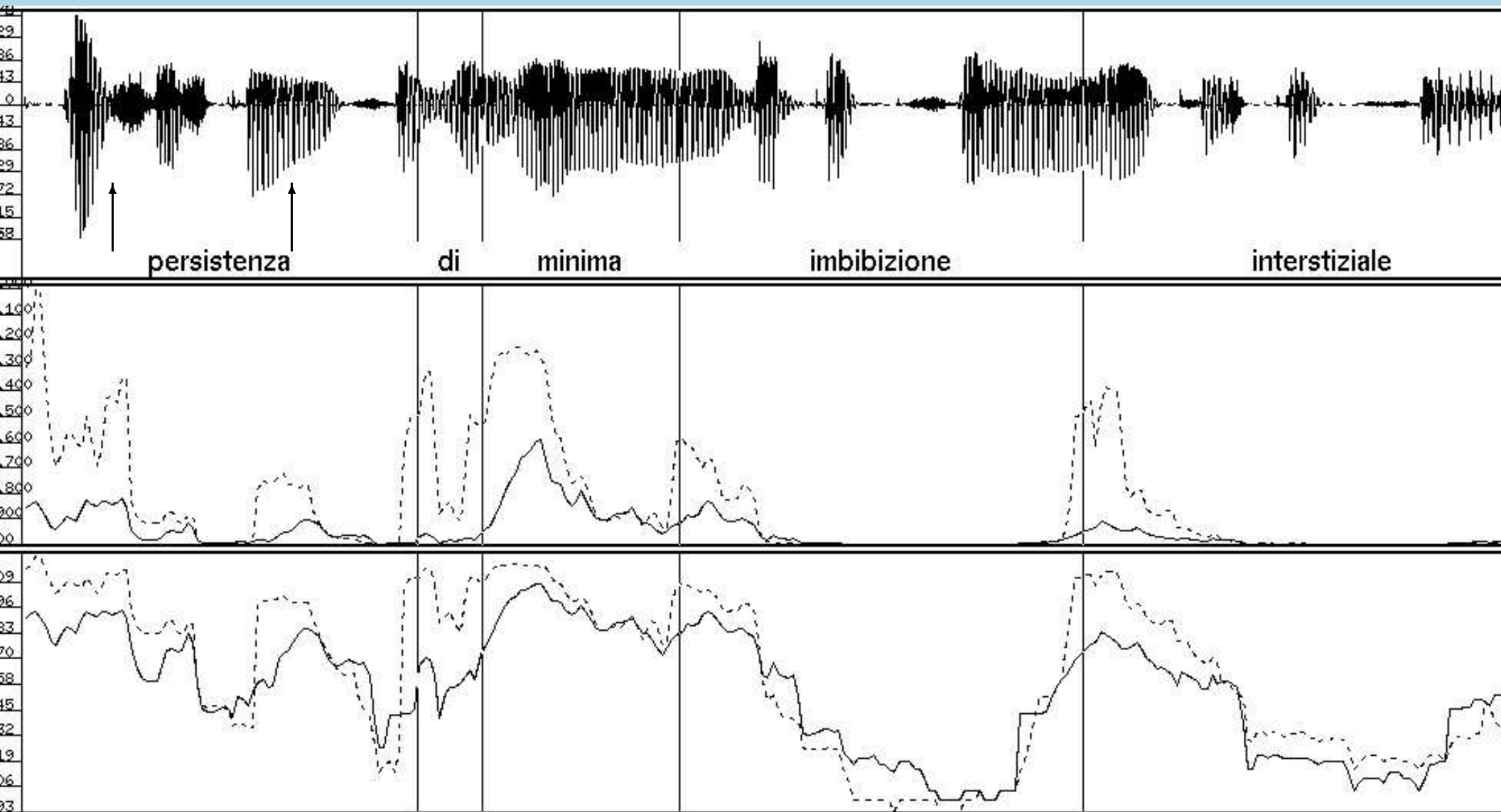
Effettua il backtracking per ricostruire la sequenza più probabile.

# Viterbi Beam Search



*Best partial paths are drawn with full lines. Crossed nodes corresponds to states not in the beam. The thin blue line indicates the backtracking information.*

# Regioni critiche in decodifica



*Tratteggiata: rete lineare, Continua: rete ad albero*

Variazione del numero di stati attivi durante la decodifica di una frase



# Rappresentazione ad albero di bigrammi

$\text{succ}(x) = \{x, y\}$

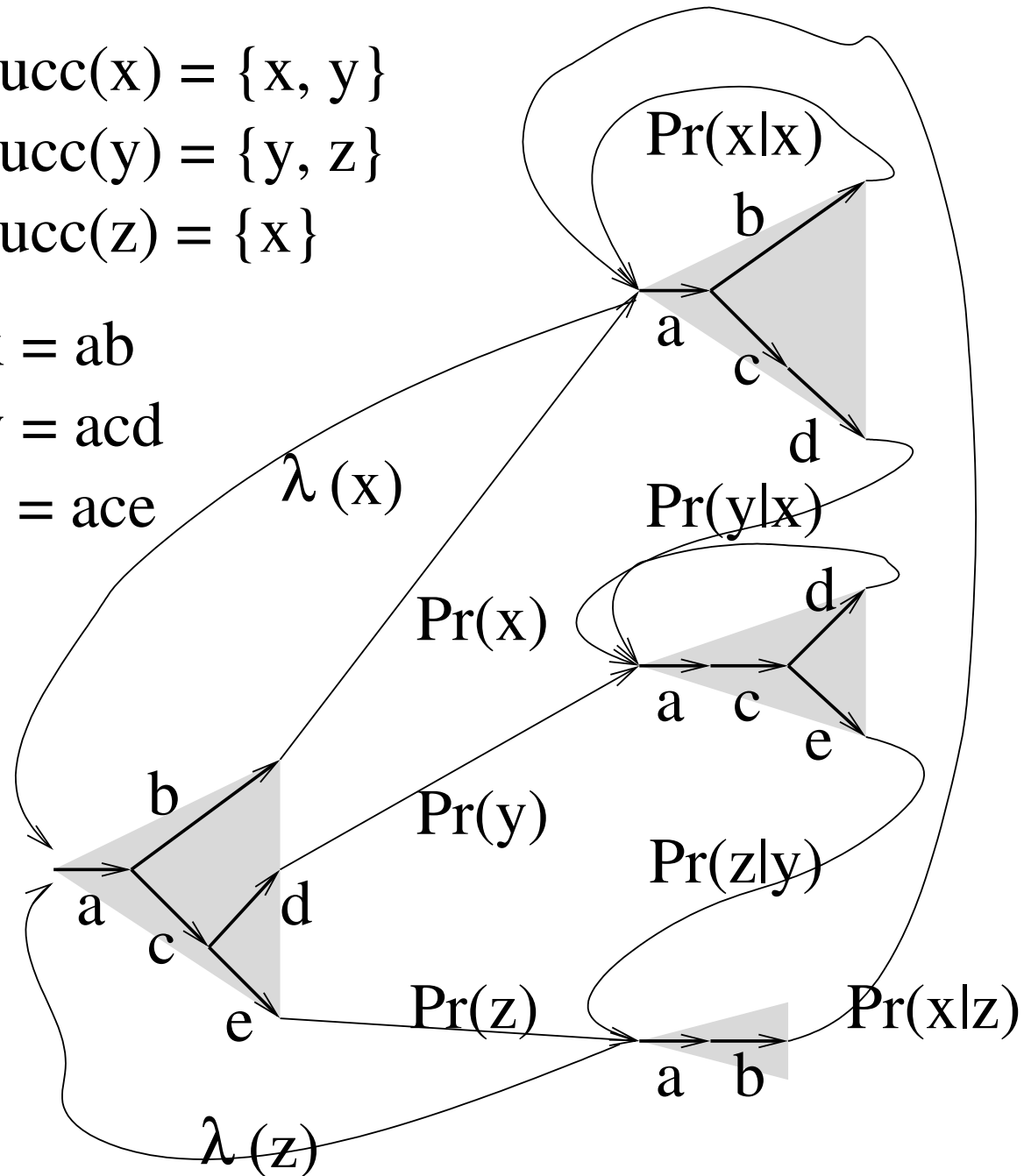
$\text{succ}(y) = \{y, z\}$

$\text{succ}(z) = \{x\}$

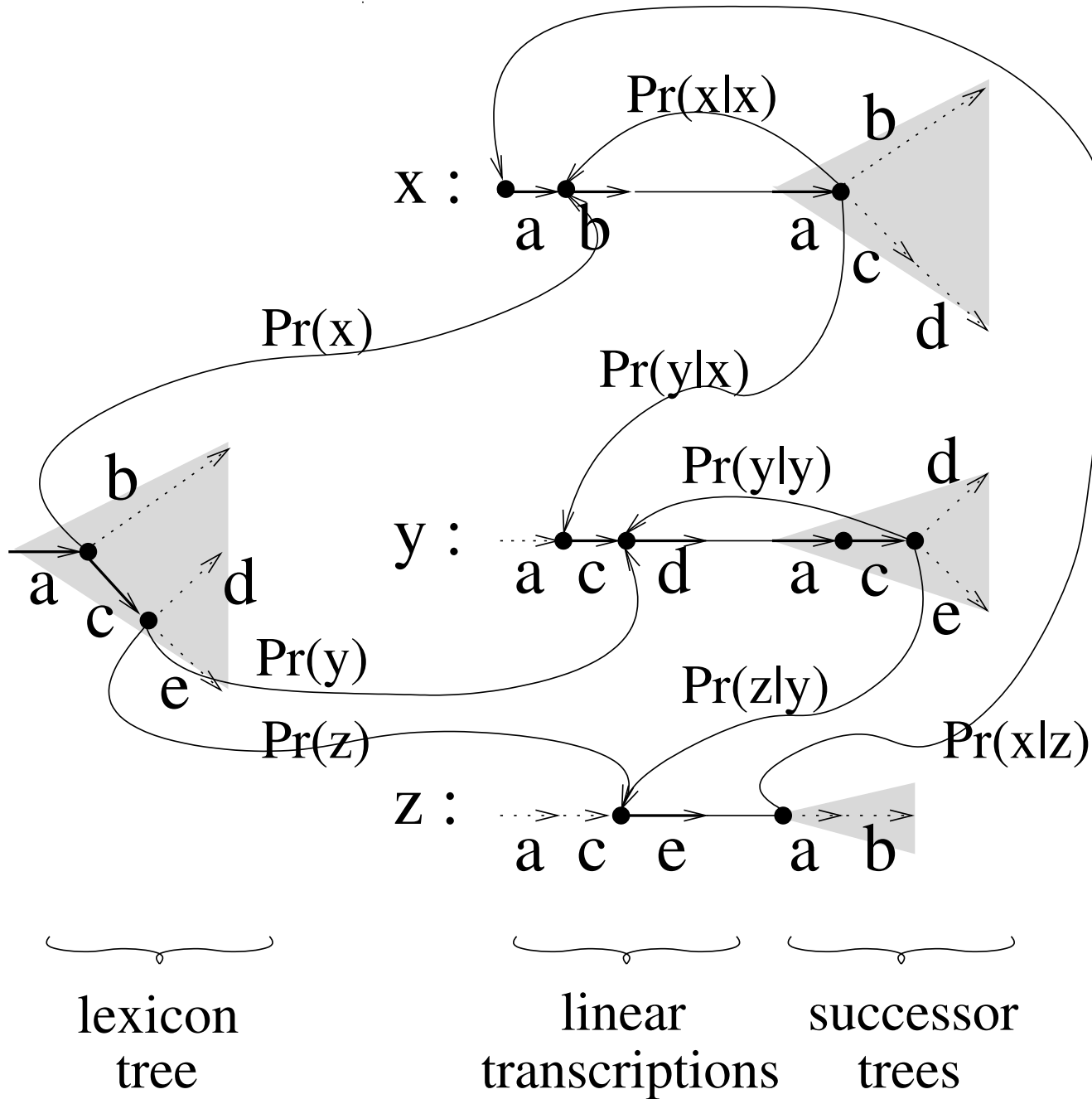
$x = ab$

$y = acd$

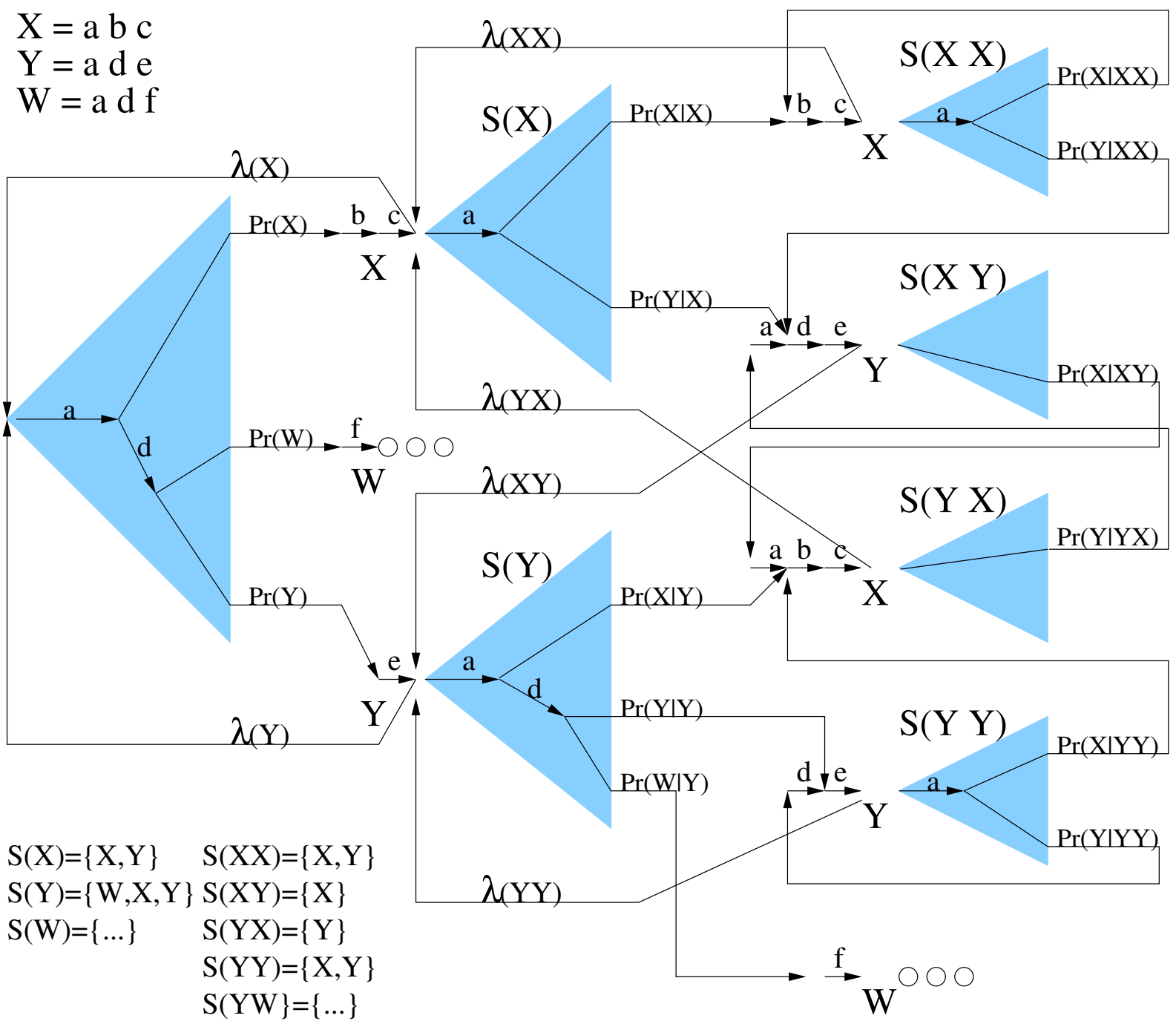
$z = ace$



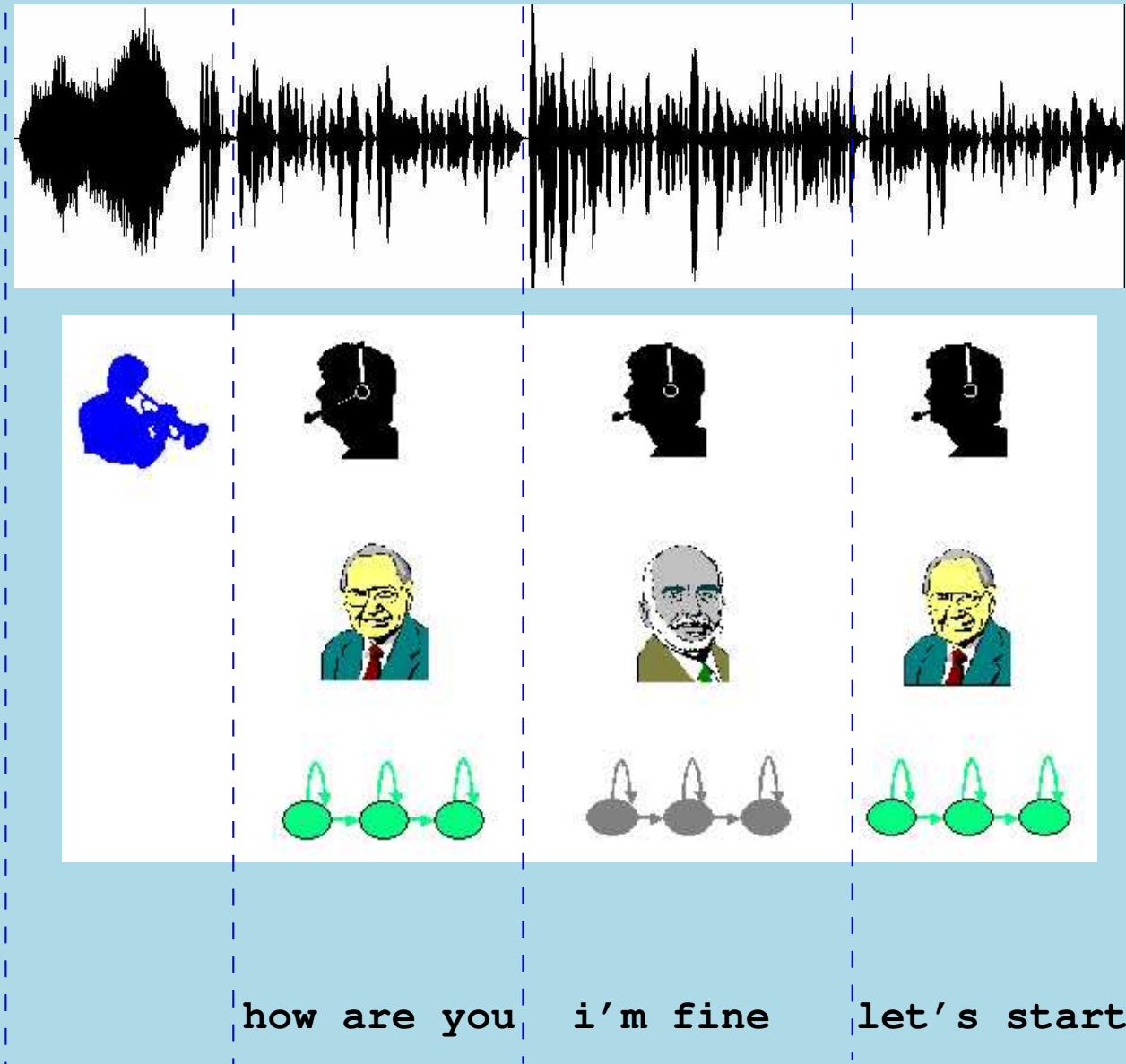
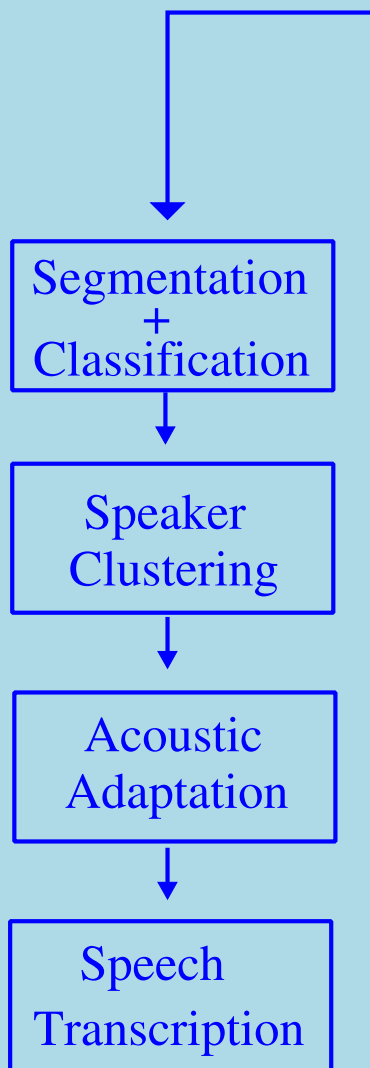
# Rappresentazione a code condivise (Shared-tail)



# Rappresentazione a code condivise di trigrammi



# Un sistema di trascrizione



# Esempio: il sistema ITC-irst per la trascrizione di notiziari

## Modelli acustici (wideband)

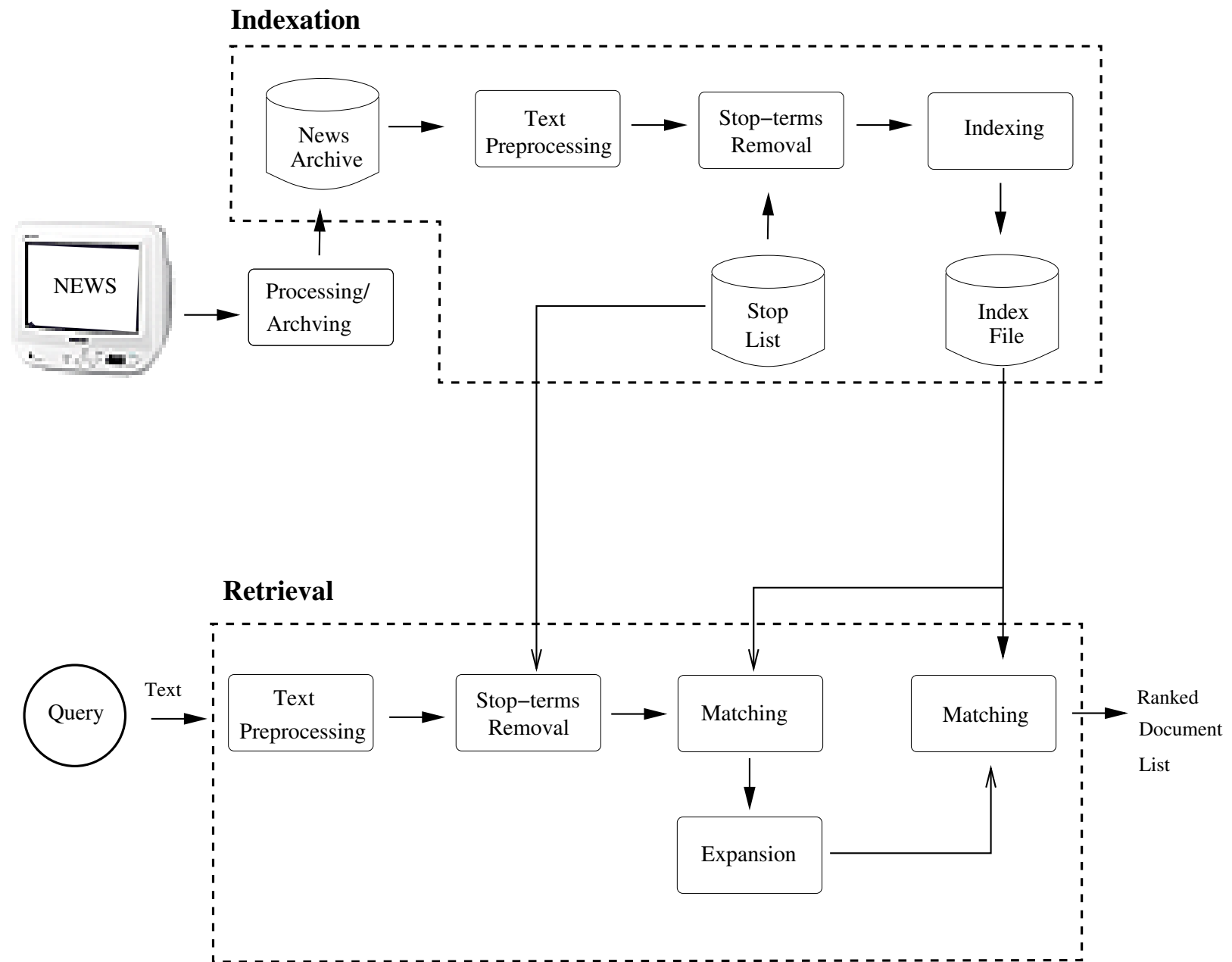
- 7700 modelli di trifoni + 3300 modelli agglomerati, 52000 gaussiane, 37500 misture
- Addestrati su 130 ore di parlato

## Modello del linguaggio

- 64000 parole, 25M trigrammi
- Vocabolario “esteso” a oltre 100000 parole
- Addestrato su un corpus di 226M parole (quotidiani) + 0.5M parole (notiziari). Aggiornato periodicamente.
- Il grafo a livello di unità contiene 12M stati e 32M archi (20M *vuoti*).
- Sottorete per “fenomeni spontanei”

Tasso di errore: complessivo: 18.5%, audio wideband: 14.8%

# Il sistema ITC-irst per Spoken Document Retrieval



E QUESTA MATTINA HA APERTO IN RIALZO PER LA BORSA DI TOKYO CHE HA CHIUSO LE CONTRATTAZIONI DELLA MATTINATA CON UN PIU' 1 \_VIRGOLA\_ 42 % IL PROGRESSO ANCHE SINGAPORE QUALE LUMPUR IL SEGNO MENO INVECE A GIAKARTA E MANI L'

WBfemale\_0

TAGLIAVA DEGLI STATI UNITI PER LO SCANDALO SEXGATE ALLA COMMISSIONE GIUSTIZIA DELLA CAMERA A DARE IL VIA LIBERA ALLA PROCEDURA DI IMPEACHMENT PER BILL CLINTON DAL NOSTRO CORRISPONDENTE PAOLO LONGO

NBmale\_0

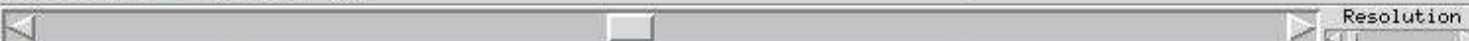
E CLINTON E' UFFICIALMENTE SOTTO INCHIESTA LA PROCEDURA DI IMPEACHMENT E' STATA AVVIATA DALLA CAMERA DEI DEPUTATI CHE COM' ERA PREVISTO SI E' SPACCATA

NBmale\_0

TUTTI REPUBBLICANI AVVERSARI DEL PRESIDENTE HANNO VOTATO A FAVORE TUTTI DEMOCRATICI CONTRO L' INCHIESTA CHE POTRA' ESSERE ALLARGATA ORA CHE HA TUTTA L' ATTIVITA' DI CLINTON HA SPESO ANNI DI PRESIDENZA



GRR981006\_0600



WBfemale_0		NBmale_0
DI TOKYO CHE HA ...	TAGLIAVA DEGLI STATI UNITI PER LO SCANDALO SEXGATE ALLA... ... NOSTRO CORRISPONDENTE PAOLO LONGO	E CLINTON E' UFFICIALMENTE SOTTO ... ... SPACCATA

4:35 4:40 4:45 4:50 4:55

Cursor : 04:38.0 Selection : 04:38.0 - 04:49.0 (11.0)

File Play View Content Help



CARCERE AD ERICA QUATTORDICI  
A DON QUESTA LA DECISIONE DEI  
DUE GIUDICI



Clip info:

[Empty text field]



## Riferimenti Bibliografici

L.R. Rabiner and B.W. Juang,  
*Fundamentals of Speech Recognition*,  
Prentice-Hall, 1993, ISBN: 0-13-015157-2

F. Jelinek,  
*Statistical Methods for Speech Recognition*,  
MIT Press, 1998, ISBN: 0-262-10066-5

R. De Mori, (Ed.),  
*Spoken Dialogues with Computers*,  
Academic Press, 1998, ISBN: 0-12-209055-1

S. Furui,  
*Digital Speech Processing, Synthesis, and Recognition*,  
Marcel Dekker, 2000, ISBN: 0-8247-0452-5

X. Huang, A. Acero, and H.W. Hon,  
*Spoken Language Processing*,  
Prentice Hall, 2001, ISBN: 0-13-022616-5