Hidden Markov Model

Content

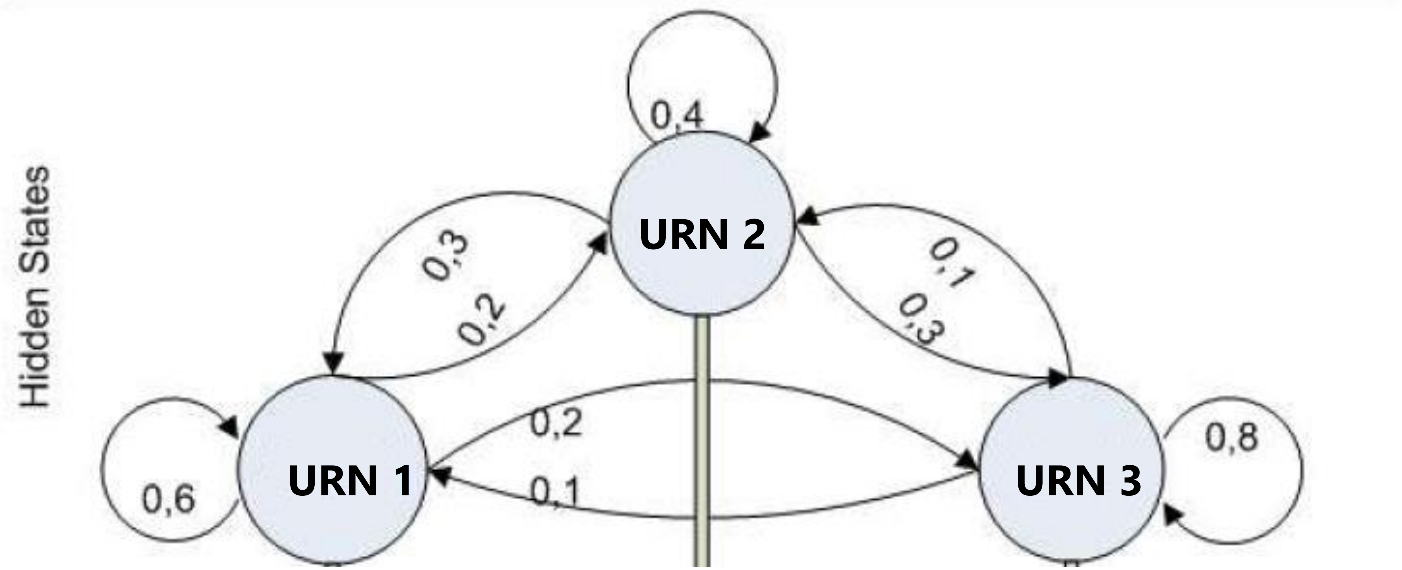
1. Hidden Markov Model (HMM)
2. The Three Basic Problems for HMMs

   Problem 1 → Solution: Forward/ Backward Algorithm

   Problem 2 → Solution: Viterbi Algorithm

   Problem 3 → Solution: Baum- Welch Algorithm
3. Example

HMM

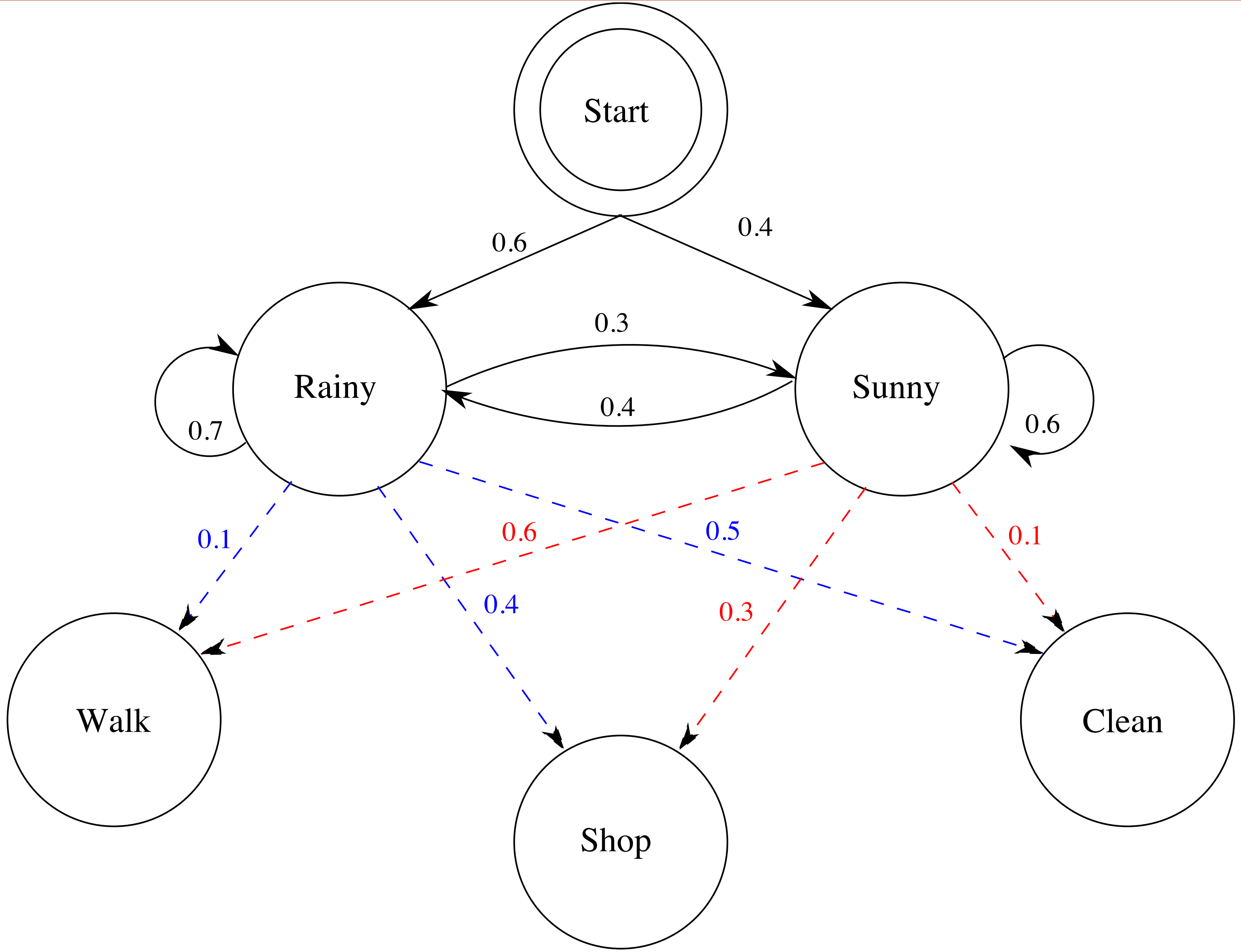


Elements of an HMM

$$
\begin{aligned}
T &= \text{the length of the observation sequence} \\
N &= \text{the number of states in the model} \\
M &= \text{the number of observation symbols} \\
Q &= \{q_0, q_1, \ldots, q_{N-1}\} = \text{the states of the Markov process} \\
V &= \{0, 1, \ldots, M-1\} = \text{set of possible observations} \\
A &= \text{the state transition probabilities} \\
B &= \text{the observation probability matrix} \\
\pi &= \text{the initial state distribution} \\
\mathcal{O} &= (\mathcal{O}_0, \mathcal{O}_1, \ldots, \mathcal{O}_{T-1}) = \text{observation sequence.}
\end{aligned}
$$

→**HMM completely characterized by:** $\lambda = (A, B, \pi)$.

Example



Start
0.6
0.4
Rainy
0.3
0.4
Sunny
0.7
0.6
0.1
0.6
0.5
0.1
0.4
0.3
Walk
Shop
Clean

HMM = triple $\lambda = (A, B, \pi)$

The matrix $A = \{a_{ij}\}$ is $N \times N$ with

$$a_{ij} = P(\text{state } q_j \text{ at } t+1 \,|\, \text{state } q_i \text{ at } t)$$

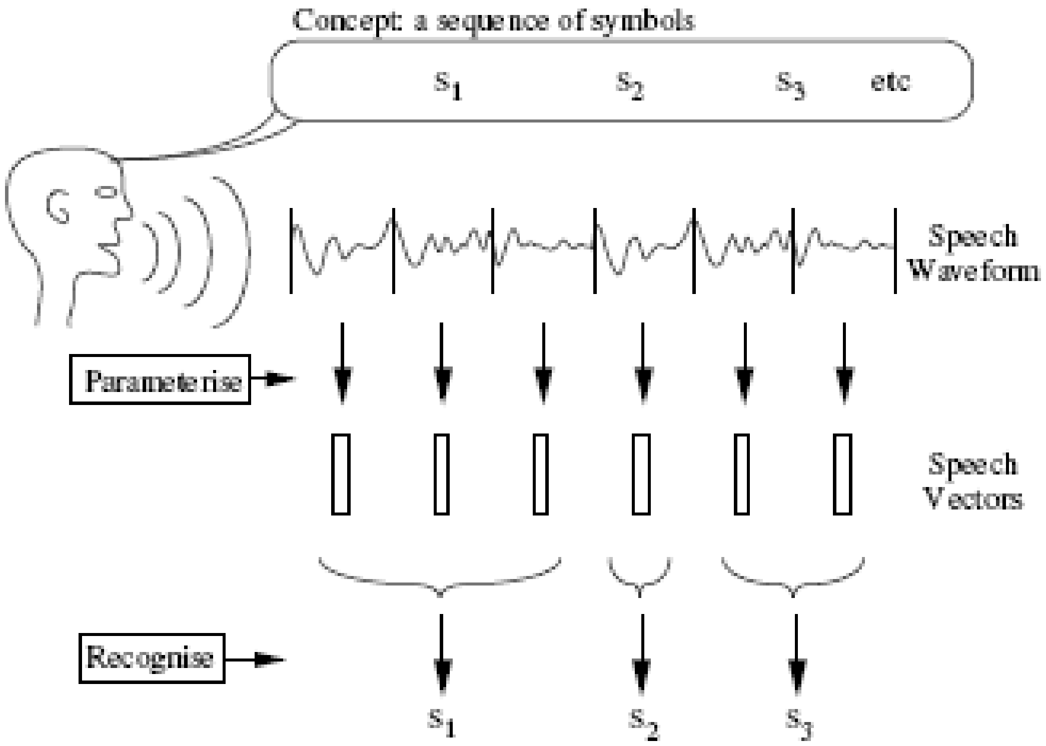
$B = \{b_j(k)\}$ is an $N \times M$ with

$$b_j(k) = P(\text{observation } k \text{ at } t \,|\, \text{state } q_j \text{ at } t).$$

$$\begin{tikzcd}
\text{Markov process:} & X_0 \arrow[r, "A"] \arrow[d, "B"'] & X_1 \arrow[r, "A"] \arrow[d, "B"'] & X_2 \arrow[r, "A"] \arrow[d, "B"'] & \cdots \arrow[r, "A"] & X_{T-1} \arrow[d, "B"'] \\
\text{Observations:} & \mathcal{O}_0 & \mathcal{O}_1 & \mathcal{O}_2 & \cdots & \mathcal{O}_{T-1}
\end{tikzcd}$$

Why HMM?

- No one-to-one mapping: speech – word symbol
- Different symbols – same sound
- Large variation in speech
  - Speaker variability
  - Mood
  - Environment



- No explicit symbol boundary detection

→ Speech waveform is NOT a concatenation of static patterns

Three classical problems for HMM

Efficient algorithms exist for solving the following three HMM problems.

Problem 1: Given the model $\lambda = (A, B, \pi)$ and a series of observations $\mathcal{O}$, find $P(\mathcal{O} \mid \lambda)$, that is, find the probability of the observed sequence given the (putative) model.
Problem 2: Given the model $\lambda = (A, B, \pi)$ and the observations $\mathcal{O}$, determine the most likely state sequence. In other words, we want to uncover the hidden part of the HMM.
Problem 3: Given the observations $\mathcal{O}$, "train" the model to best fit the observations. Note that the dimensions of the matrices are fixed, but the elements of A, B and π can vary, subject only to the row stochastic condition.

Three problems (Rabiner, 1989)

Given an observation sequence $\mathbf{O}=(o_0,o_1,\ldots,o_{T-1})$, **and an HMM** $\lambda=(\mathbf{A},\mathbf{B},\pi)$

Problem 1:

How to compute $P(\mathbf{O}|\lambda)$ *efficiently* ?

⇨ ***Evaluation Problem***

Problem 2:

How to choose an optimal state sequence $\mathbf{Q}=(q_1,q_2,\ldots\ldots,q_T)$ which best explains the observations?

⇨ ***Decoding Problem*** $P(Q^* \mid O, \lambda) = \max_Q P(Q \mid O, \lambda)$

Problem 3:

How to adjust the model parameters $\lambda=(\mathbf{A},\mathbf{B},\pi)$ to maximize $P(\mathbf{O}|\lambda)$?

⇨ ***Learning/Training Problem***

PROBLEM 1: Evaluation

- Straightforward calculation is too time consuming: O(TN^T) multiplications and additions

$$
\begin{aligned}
P(\mathcal{O} \,|\, \lambda) &= \sum_X P(\mathcal{O}, X \,|\, \lambda) \\
&= \sum_X P(\mathcal{O} \,|\, X, \lambda) P(X \,|\, \lambda) \\
&= \sum_X \pi_{x_0} b_{x_0}(\mathcal{O}_0) a_{x_0,x_1} b_{x_1}(\mathcal{O}_1) \cdots a_{x_{T-2},x_{T-1}} b_{x_{T-1}}(\mathcal{O}_{T-1}).
\end{aligned}
$$

Alpha-pass

For $i = 0, \ldots$,T-1 and $t = 0,\ldots, N$-1 we define

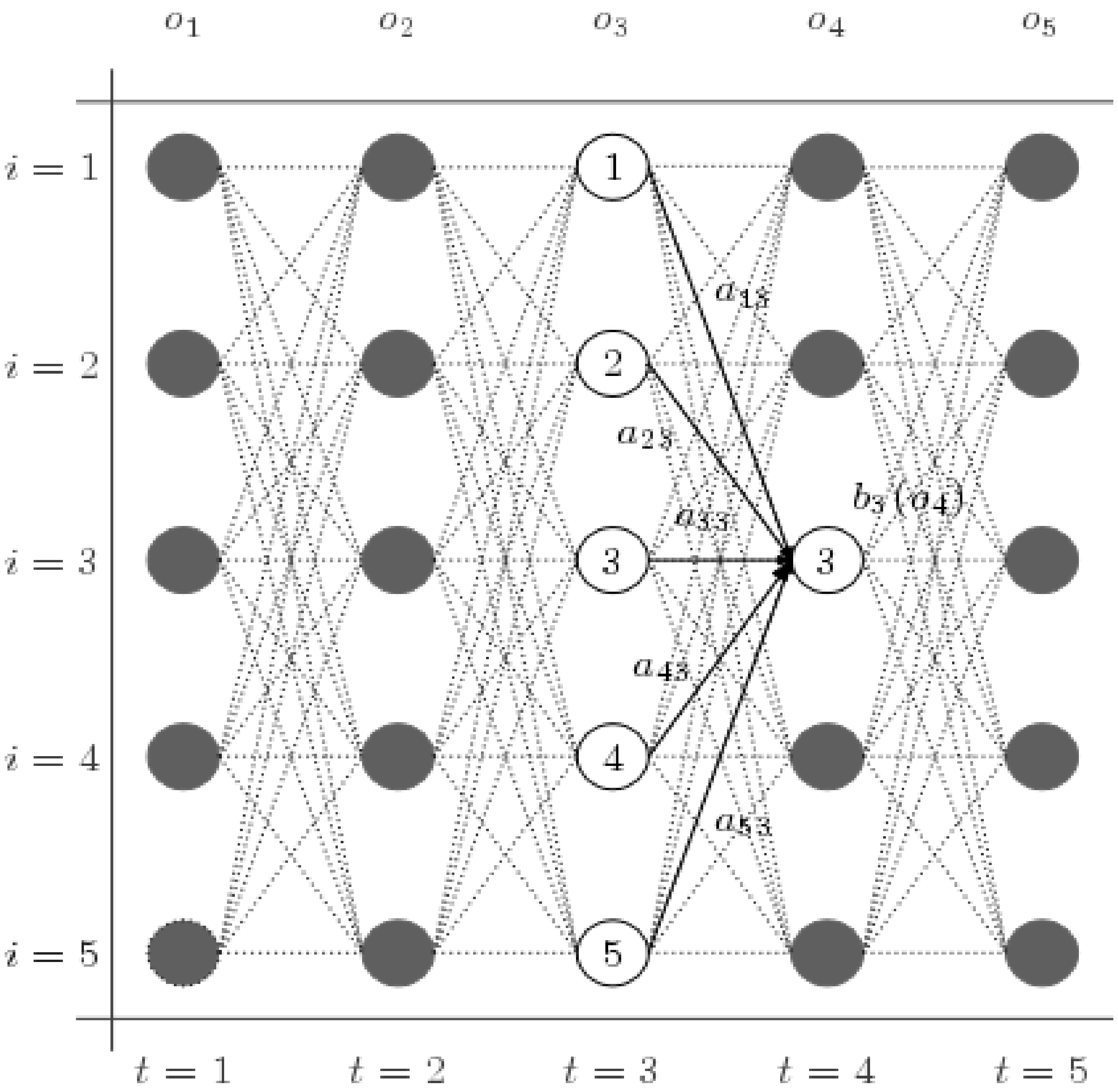
$$\alpha_t(i) = P(\mathcal{O}_0, \mathcal{O}_1, \ldots, \mathcal{O}_t, x_t = q_i \mid \lambda). \qquad (8)$$

ALGORITHM „alpha-pass”: forward method

1. Let $\alpha_0(i) = \pi_i b_i(\mathcal{O}_0)$, for $i = 0, 1, \ldots, N-1$
2. For $t = 1, 2, \ldots, T-1$ and $i = 0, 1, \ldots, N-1$, compute
$$\alpha_t(i) = \left[\sum_{j=0}^{N-1} \alpha_{t-1}(j) a_{ji}\right] b_i(\mathcal{O}_t)$$
3. Then from (8) it is clear that
$$P(\mathcal{O} \mid \lambda) = \sum_{i=0}^{N-1} \alpha_{T-1}(i).$$

Alpha-pass $\alpha_t(i) = P(\mathcal{O}_0, \mathcal{O}_1, \ldots, \mathcal{O}_t, x_t = q_i \,|\, \lambda).$

$$\alpha_t(i) = \left[\sum_{j=0}^{N-1} \alpha_{t-1}(j) a_{ji}\right] b_i(\mathcal{O}_t)$$



Problem 2: Decoding – „beta-pass”

For $i = 0, \ldots, T$-1 and $t = 0, \ldots, N$-1 we define

$$\beta_t(i) = P(\mathcal{O}_{t+1}, \mathcal{O}_{t+2}, \ldots, \mathcal{O}_{T-1} \,|\, x_t = q_i, \lambda).$$

ALGORITHM: „beta-pass”: backward calculation

1. Let $\beta_{T-1}(i) = 1$, for $i = 0, 1, \ldots, N-1$.
2. For $t = T-2, T-1, \ldots, 0$ and $i = 0, 1, \ldots, N-1$ compute

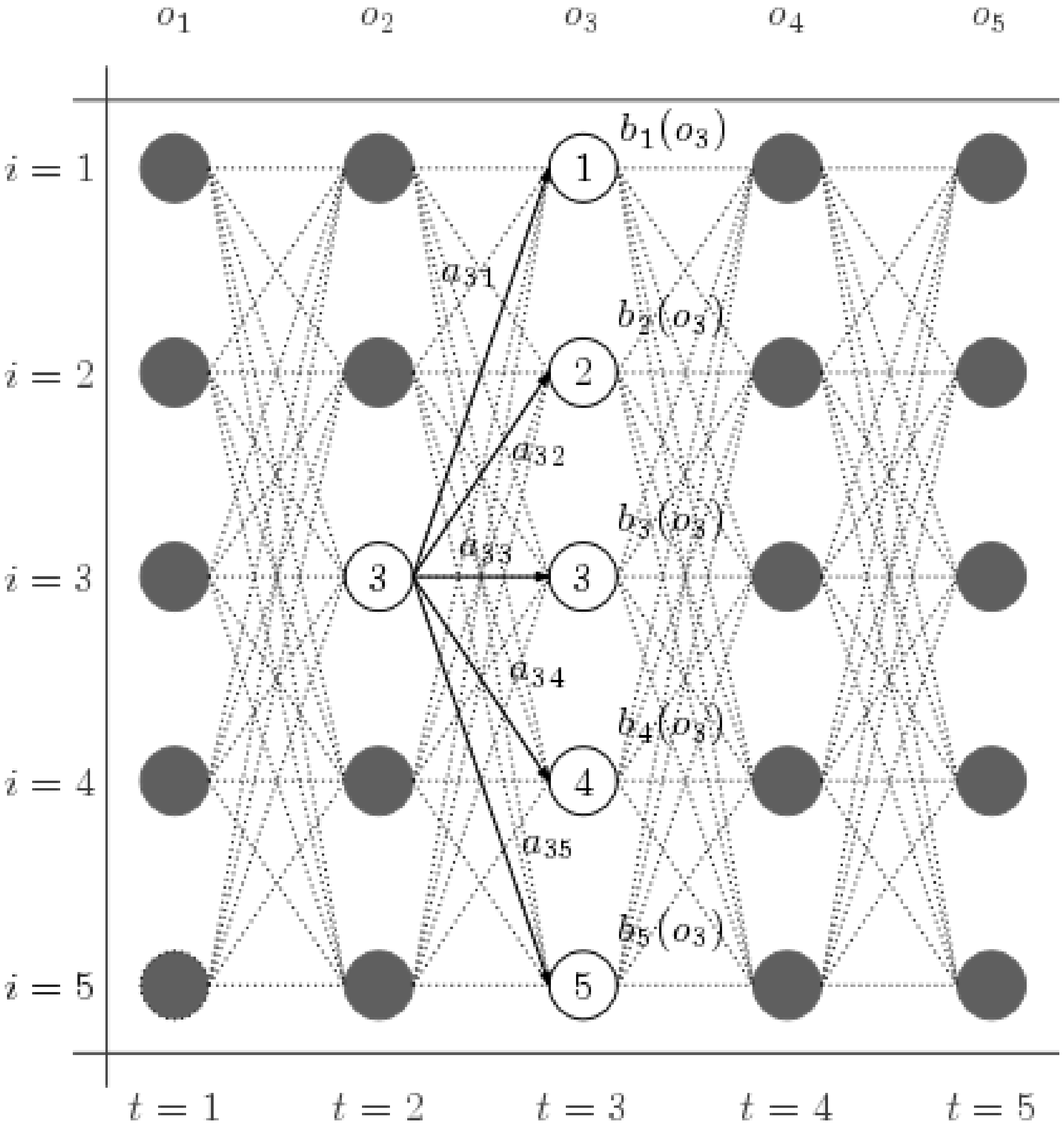
$$\beta_t(i) = \sum_{j=0}^{N-1} a_{ij} b_j(\mathcal{O}_{t+1}) \beta_{t+1}(j).$$

For $t = 0, 1, \ldots, T-2$ and $i = 0, 1, \ldots, N-1$, define

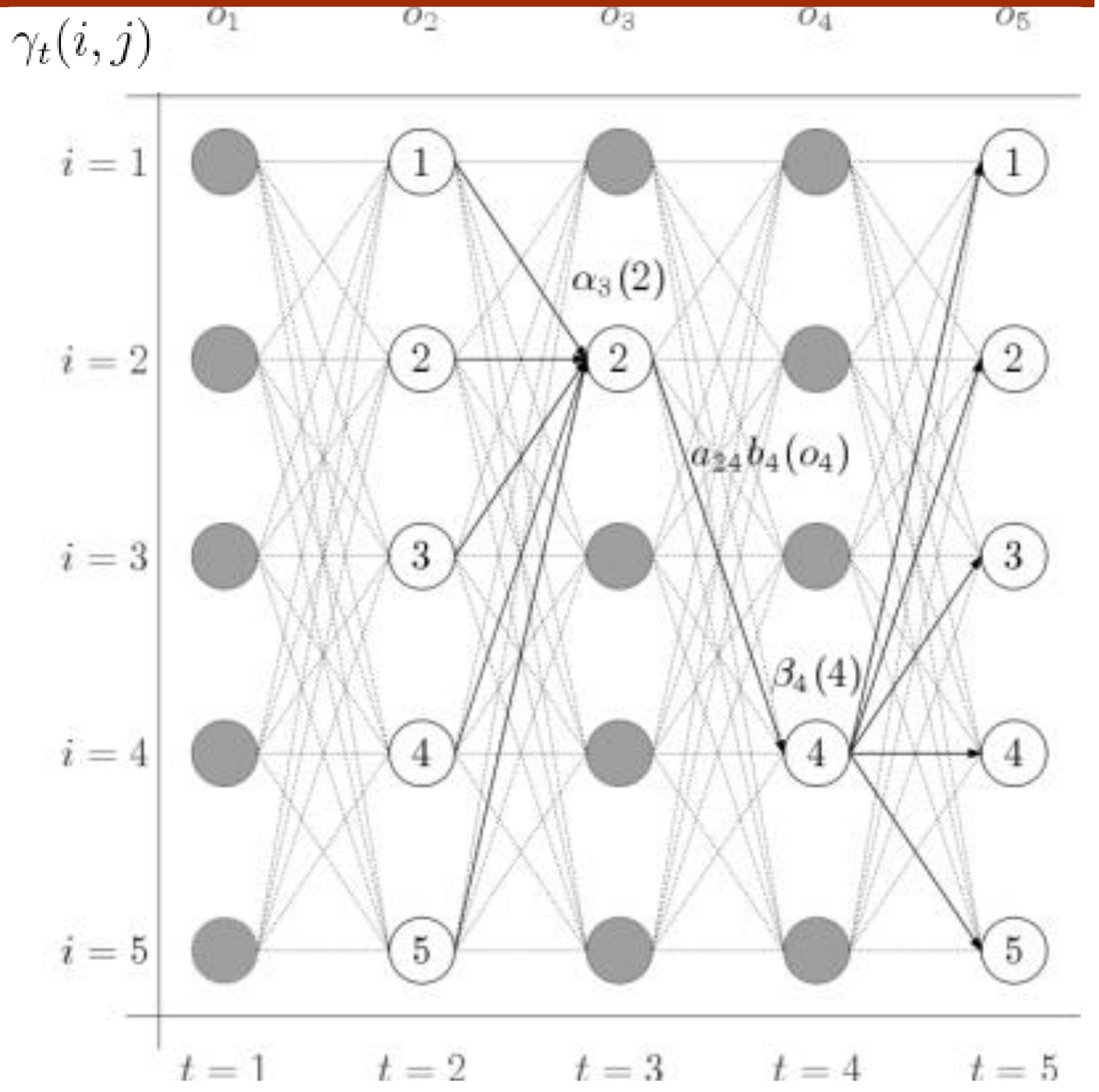
$$\gamma_t(i) = P(x_t = q_i \,|\, \mathcal{O}, \lambda) = \frac{\alpha_t(i)\beta_t(i)}{P(\mathcal{O} \,|\, \lambda)}$$

Beta-pass

$$\beta_t(i) = \sum_{j=0}^{N-1} a_{ij} b_j(\mathcal{O}_{t+1}) \beta_{t+1}(j).$$



Gamma



$\gamma_t(i, j)$
o_1
o_2
o_3
o_4
o_5
$i = 1$
$i = 2$
$i = 3$
$i = 4$
$i = 5$
$\alpha_3(2)$
$a_{24}b_4(o_4)$
$\beta_4(4)$
$t = 1$
$t = 2$
$t = 3$
$t = 4$
$t = 5$

Beta-pass: best state sequence

Q: Is the sequence Q = (q,..),
where

$$q_t^* = \arg\max_i \gamma_t(i)$$

the optimal sequence?

Answer: No
Example:

$\gamma_t(i)$	element			
	0	1	2	3
$P(H)$	0.188170	0.519432	0.228878	0.803979
$P(C)$	0.811830	0.480568	0.771122	0.196021

state	probability	normalized probability
$HHHH$	.000412	.042743
$HHHC$	.000035	.003664
$HHCH$	.000706	.073274
$HHCC$	.000212	.021982
$HCHH$	.000050	.005234
$HCHC$	.000004	.000449
$HCCH$	.000302	.031403
$HCCC$	.000091	.009421
$CHHH$	.001098	.113982
$CHHC$	.000094	.009770
$CHCH$	.001882	.195398
$CHCC$	.000564	.058619
$CCHH$	.000470	.048849
$CCHC$	.000040	.004187
$CCCH$	.002822	.293096
$CCCC$	.000847	.087929

Problem 2 and Viterbi's Algorithm

1. Initialization
$$\begin{aligned} \delta_1(i) &= \pi_i \, b_{iY_1}, \quad 1 \le i \le N \\ \psi_1(i) &= 0 \end{aligned}$$

2. Recursion

$$\begin{aligned} \delta_t(j) &= \max_{1\le i\le N}[\delta_{t-1}(i)a_{ij}]b_{jY_t}, \quad 2 \le t \le T \\ \psi_t(j) &= \arg\max_{1\le i\le N}[\delta_{t-1}(i)a_{ij}], \quad 2 \le t \le T \end{aligned}$$

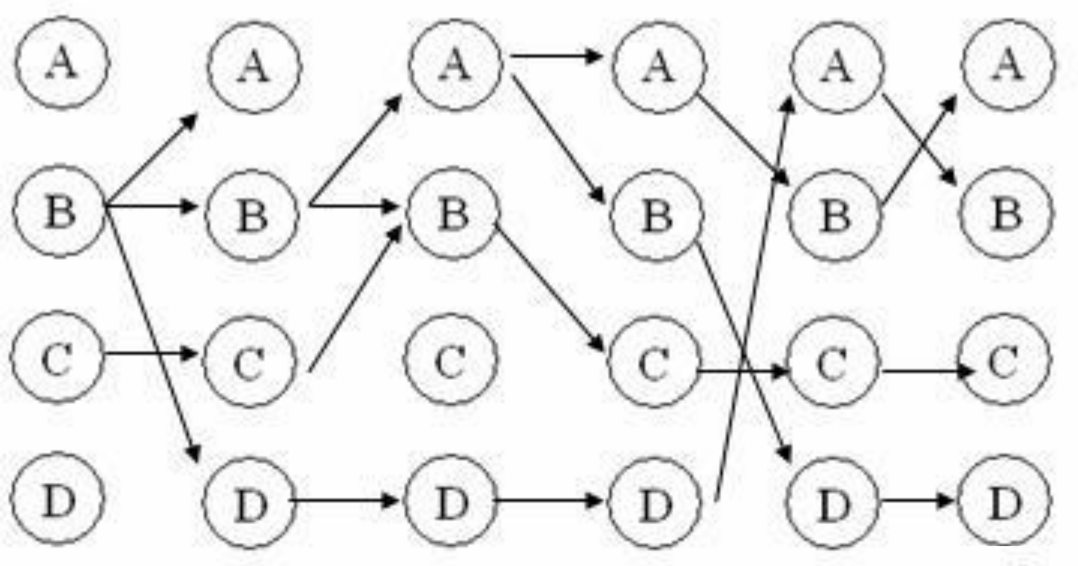
3. Termination

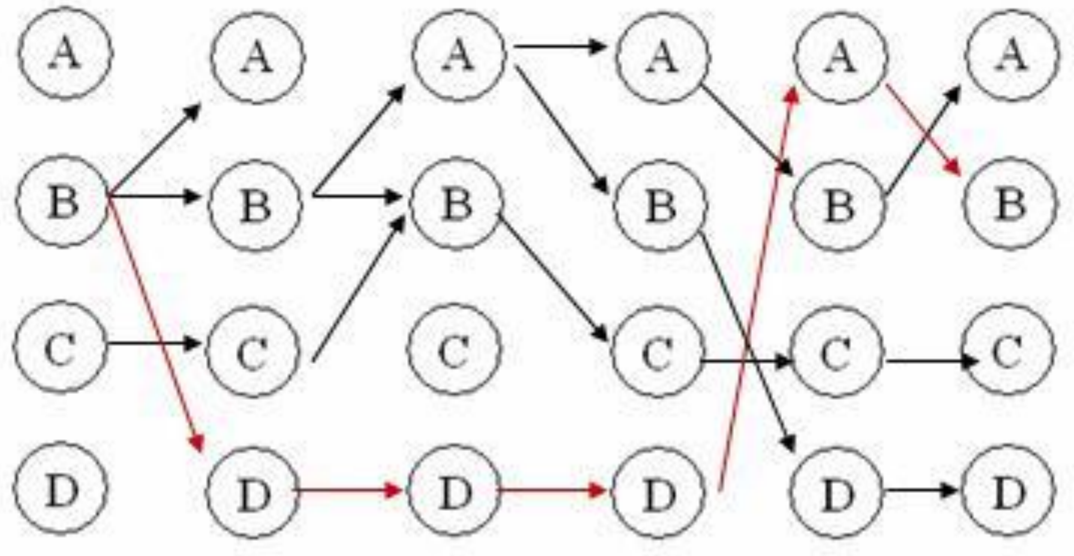
$$\begin{aligned} P^* &= \max_{1\le i\le N}[\delta_T(i)] \\ X_T^* &= \arg\max_{1\le i\le N}[\delta_T(i)] \end{aligned}$$

4. Path backtracking

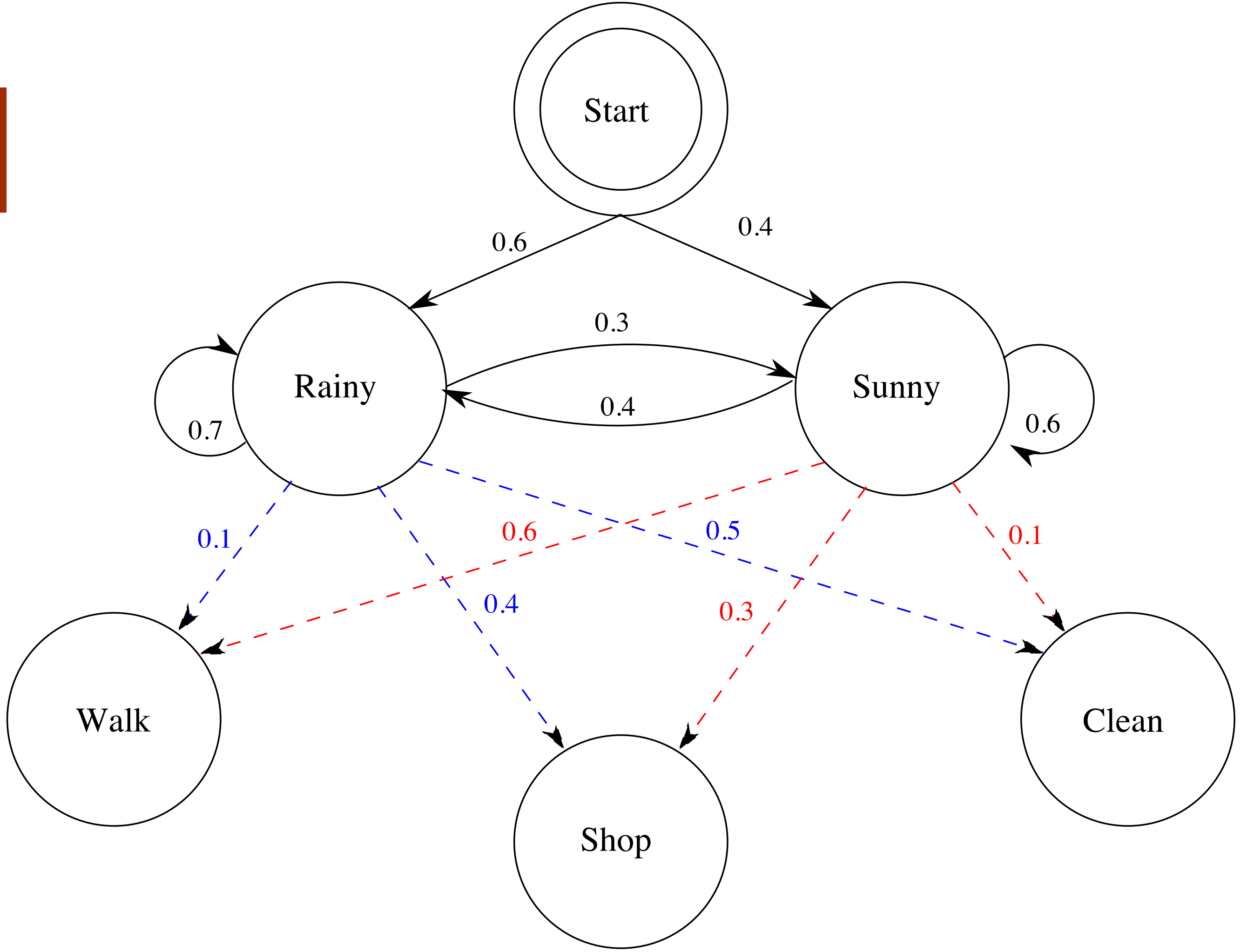
$$X_t^* = \psi_{t+1}(X_{t+1}^*), \quad t = T-1, T-2, \ldots, 1$$

Viterbi - backtracking





Example



Observation sequence: ['walk', 'shop', 'clean’]
Viterbi Solution: ['Sunny', 'Rainy', 'Rainy', 'Rainy']

Problem 3: „learning" and Baum-Welch method

We define

$$\gamma_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(\mathcal{O}_{t+1}) \beta_{t+1}(j)}{P(\mathcal{O} \mid \lambda)}. \quad \equiv P(q_t = S_i, q_{t+1} = S_j \mid O, \lambda)$$

Thus

$$\gamma_t(i) = \sum_{j=0}^{N-1} \gamma_t(i,j).$$

Algorithm EM (Baum-Welch)

1. Initialize, $\lambda = (A, B, \pi)$.
2. Compute $\alpha_t(i)$, $\beta_t(i)$, $\gamma_t(i,j)$ and $\gamma_t(i)$.
3. Re-estimate the model $\lambda = (A, B, \pi)$.
4. If $P(\mathcal{O} \mid \lambda)$ increases, goto 2.

Algorithm Baum-Welc: reestimation step

For $i = 0, 1, \ldots, N-1$, let

$$\pi_i = \gamma_0(i)$$

For $i = 0, 1, \ldots, N-1$ and $j = 0, 1, \ldots, N-1$, compute

$$a_{ij} = \sum_{t=0}^{T-2} \gamma_t(i,j) \Bigg/ \sum_{t=0}^{T-2} \gamma_t(i)$$

For $j = 0, 1, \ldots, N-1$ and $k = 0, 1, \ldots, M-1$, compute

$$b_j(k) = \sum_{\substack{t \in \{0,1,\ldots,T-2\} \\ \mathcal{O}_t = k}} \gamma_t(j) \Bigg/ \sum_{t=0}^{T-2} \gamma_t(j)$$

Cave and Neuwirth Experiments

- They selected the Brown Corpus as a representative sample of English.
  - This corpus of more than 1,000,000 words was carefully compiled (in the early 1960's) so as to contain a diverse selection of written English.
  - Cave and Neuwirth eliminated all numbers, punctuation and special characters, and converted all letters to lower-case, leaving 27 distinct symbols—the letters plus inter-word space.
- They then assumed that there exists a Markov process with two hidden states, with the observations given by the symbols (i.e., letters) that appear in the Brown Corpus.
- This results in an A matrix that is 2×2 and a B matrix that is 2×27.
- They then solved HMM Problem 3 for the optimal matrices

Cave - Neuwirth

Results after:

- 10,000 observation and
- about 200 iterations

$$\pi = \begin{bmatrix} 0.51316 & 0.48684 \end{bmatrix}$$

$$A = \begin{bmatrix} 0.47468 & 0.52532 \\ 0.51656 & 0.48344 \end{bmatrix}$$

$$\pi = \begin{bmatrix} 0.00000 & 1.00000 \end{bmatrix}$$

$$A = \begin{bmatrix} 0.25596 & 0.74404 \\ 0.71571 & 0.28429 \end{bmatrix}$$

	Initial B		Final B	
letter	state 0	state 1	state 0	state 1
a	0.0372642	0.0366080	0.0044447	0.1306242
b	0.0386792	0.0389249	0.0241154	0.0000000
c	0.0358491	0.0338276	0.0522168	0.0000000
d	0.0353774	0.0370714	0.0714247	0.0003260
e	0.0349057	0.0352178	0.0000000	0.2105809
f	0.0344340	0.0370714	0.0374685	0.0000000
g	0.0400943	0.0370714	0.0296958	0.0000000
h	0.0344340	0.0347544	0.0670510	0.0085455
i	0.0349057	0.0370714	0.0000000	0.1216511
j	0.0391509	0.0366080	0.0065769	0.0000000
k	0.0363208	0.0356812	0.0067762	0.0000000
l	0.0353774	0.0403151	0.0717349	0.0000135
m	0.0344340	0.0366080	0.0382657	0.0000000
n	0.0410377	0.0370714	0.1088182	0.0000000
o	0.0396226	0.0398517	0.0000000	0.1282757
p	0.0377358	0.0338276	0.0388589	0.0000047
q	0.0377358	0.0398517	0.0011958	0.0000000
r	0.0344340	0.0403151	0.1084196	0.0000000
s	0.0358491	0.0366080	0.1034371	0.0000000
t	0.0377358	0.0352178	0.1492508	0.0134756
u	0.0349057	0.0361446	0.0000000	0.0489816
v	0.0405660	0.0370714	0.0169406	0.0000000
w	0.0377358	0.0384615	0.0286993	0.0000000
x	0.0382075	0.0370714	0.0035874	0.0000000
y	0.0382075	0.0389249	0.0269053	0.0000003
z	0.0382075	0.0338276	0.0005979	0.0000000
space	0.0367925	0.0389249	0.0035184	0.3375209

Letter classification

V	Vowel
SP	Space
C	Consonant
FL	First Letter
LL	Last Letter
VF	Vowel Follower
VP	Vowel Proceeder
CP	Consonant Follower.

Cave Neuwirth experiment with 12 states

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	SP
V	*				*				*						*						*						
SP																											*
C						*	*			*	*																
LL											*													*			
FL	*									*					*		*										
VF																											
VP										*							*										
CF																								*			
1				*				*					*	*				*	*						*		
2					*				*						*						*						
3		*	*	*					*			*		*		*		*	*	*						*	
4								*				*			*			*		*	*						
5	*				*				*						*						*						
6		*	*	*		*	*	*		*		*	*	*		*	*	*	*			*	*		*	*	
7			*	*			*				*	*	*	*		*		*	*	*		*			*		
8					*							*								*							
9						*					*	*	*	*		*		*	*		*		*	*			
10	*				*				*						*												
11																											*
12			*			*	*									*			*				*				

Table 2: Cave and Neuwirth's 12 states result interpretation