Billio Jeverson
1002979

28 November 2019

1.)

State sequence	Observation sequence O	State, T
~~X~~, Y, Z, X	b, c, a, b	{Start, X, Y, Z, X, Stop}
X, Z, Y	a, c, a	{Start, X, Z, Y, Stop}
Z, Y, X, Z, Y	b, c, c, b, c	{Start, Z, Y, X, Z, Y, Stop}
Z, X, Y	c, b, a	{Start, Z, X, Y, Stop}
Z, X, Y	c, c, c	{Start, Z, X, Y, Stop}

Ex:

$$a'_{x,y} = \frac{Count(x,y)}{Count(x)} = \frac{3}{6} = \frac{1}{2}$$

a' u/v

	~~Start~~ X	Y	Z	Stop
Start	2/5	0	3/5	0
X	0	1/2	1/3	~~1/5~~ 1/6
Y	1/6	0	1/6	~~4/5~~ 2/3
Z	1/2	1/2	0	0

I use a' and b' to differentiate from a, b in ~~ob~~ observation (o)

Ex:

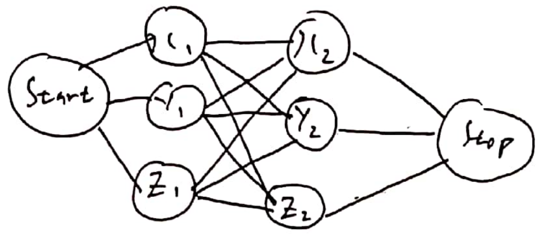
$$b'_y(o) = b_y(a) = \frac{Count(y \to a)}{Count\ y} = \frac{2}{6} = \frac{1}{3}$$

$b'_u(o)$	a	b	c
~~Start~~ X	1/6	1/2	1/3
Y	1/3	0	2/3
Z	1/6	1/3	1/2

2.) By ~~Bitt~~ Viterbi Algorithm,

1.) we do forward pass to ~~get~~ populate the nodes

2.) We do a backward pass to get the highest path

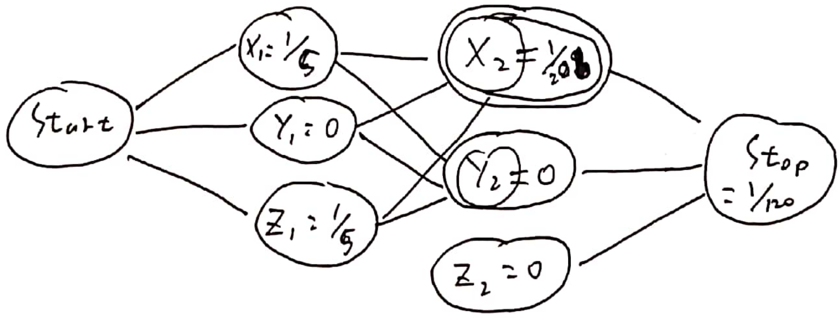


$\boxed{\text{Step 1:}}$ Populating Nodes.

for (b, b) ~~Start~~ $\boxed{\text{at Node } x_1}$, => at Y_1, Z_1 => substitute X_1 to Y_1 or Z_1 accordingly

$\cancel{a'_{start}(x_1), b}$

Val: $a'(\text{start}, X_1) \cdot b'_{X_1}(b) = 2/5 \times \frac{1}{2} = \frac{2}{10} = \frac{1}{5}$



at $\boxed{\text{Node } X_2}$, => Y_2, Z_2, substitute X_2 to Y_2, Z_2.

$$\text{Val} = \text{Max}\begin{bmatrix} x_1 \cdot a'(x_1, x_2) \cdot b'_{x_2}(b), \\ y_1 \cdot a'(y_1, x_2) \cdot b'_{x_2}(b), \\ z_1 \cdot a'(z_1, x_2) \cdot b'_{x_2}(b) \end{bmatrix}$$

$$= \text{Max}\left[0, 0, 1/24\right]$$

$$= 1/24$$

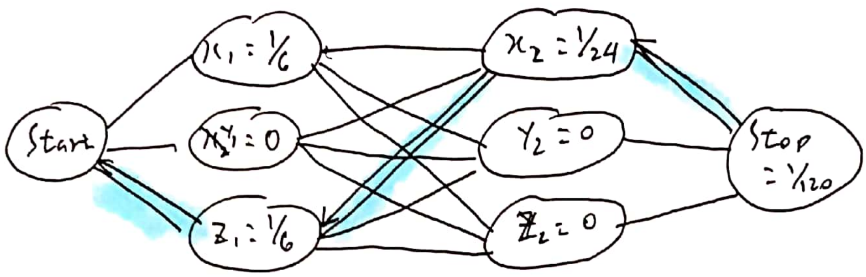
at $\boxed{\text{Stop Node}}$

$$\max\begin{bmatrix} x_2 \cdot a'(x_2, \text{stop}), \\ Y_2 \cdot a'(Y_2, \text{stop}), \\ Z_2 \cdot a'(Z_2, \text{stop}) \end{bmatrix}$$

$$= \max\left[1/20 \times 1/6, 0, 0\right]$$

$$= 1/120$$

Step 2 : We do backward to find the path



$\therefore$ Path = Start, z_1, x_2, Stop

$= \{z, x\}$

#

3.) $Y_1^*, \dots Y_n^* = \arg\max_{Y_1, \dots, Y_n} p(x_1, \dots, x_n, Y_1, \dots Y_n)$

$x_i = $ "the"

$Y_i = V$

$Y_1^* \dots Y_{i-1}^*, Y_{i+1}^*, \dots Y_n^* = \arg\max_{Y_1, \dots Y} p(x_1, \dots x_n, Y_1, \dots, Y_{i-1}, Y_{i+1} \dots Y_n | Y_i \neq V)$

$\boxed{\text{Algorithm:}}$

1. Initialization

$$\pi(0, u) \begin{cases} 1 & \text{if } u = \text{START} \\ 0 & \text{otherwise} \end{cases}$$

~~2.) for $j = 0 \dots n-1$, for each $u \in T$~~

~~$\pi(j+1, u) = \max_v \{\pi(j, v) \times b_u(x_{j+1}) \times a_{v,u}\}$~~

~~$\pi(n+1, \text{stop}) = \max \{\pi(n, N) \times a(n, \text{stop})\}$~~

2. for $j = 0 \dots n-1$, for each $u \in T$, :

if ($j+1 == i$ & $u = V$) :

$$\pi(j+1, u) = 0$$

else :

$$\pi(j+1, u) = \max_v \{\pi(j, v) \times b_u(x_{j+1}) \times a_{v,u}\}$$

Final step :

$$\pi(n+1, \text{Stop}) = \max_v \{\pi(n, v) \times a_{v, \text{stop}}\}$$

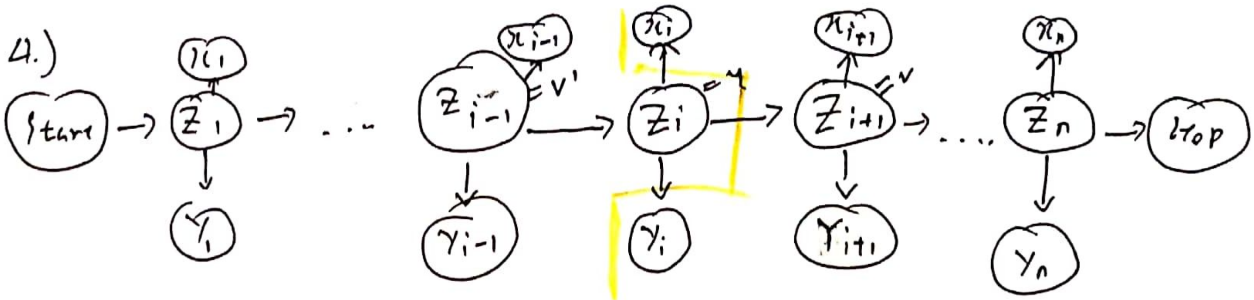
#

Explaination:

I have added an additional conditional statements to check if the stage is i and the word is V , thus value of that node is 0. which will not be the max. (different w normal viterbi algo)

Ex: If x(i) = 'the', value of the node for label "verb"(V) will be zero and thus the max label will not be verb.

4.)



$$P = \frac{P(x_1, \dots x_{i-1}, y_1, \dots y_{i-1}, z_i = u, x_i, \dots x_n, y_i, \dots y_n ; \theta)}{\sum_v P(x_1, \dots x_{k-1}, y_1, \dots y_{k-1}, z_k = v, x_k \dots x_n, y_k \dots y_n ; \theta)}$$

$$P = \frac{\alpha_u(i)\, \beta_u(i)}{\sum_v \alpha_v(k)\, \beta_v(k)}$$

$\alpha_u(i) = P(x_1, \dots x_{i-1}, y_1, \dots y_{i-1}, z_i = u)$

Sum of all scores from start to node u at stage i.

$\boxed{\text{Forward pass,}}$

$$\alpha_v(i+1) = \sum_u \alpha_u(i) \,.\, a(u,v) \,.\, b_u(x_i) \,.\, b_u(y_i)$$

$\Uparrow$

algorithm from left to Right

Base case
$\alpha_u(1) = a(\text{start}, u)$

$\beta_u(i) = P(x_i, \dots x_n, y_i, \dots y_n \mid z_i = u)$

Sum of all scores from node u at stage i to stop

$\boxed{\text{Backward Pass,}}$

$$\beta_u(i) = \sum_v \cancel{\alpha_{u,v}}\; a(u,v) \,.\, b_u(x_i) \,.\, b_u(y_i)\; \beta_v(i+1)$$

$\Uparrow$

algorithm from right to left.

4.) Time Complexity :

No of States = N , $(0, \dots N)$

No of nodes within ~~one~~ state = n nodes,
assuming there are n unique labels $\{z_1, \dots z_n\}$

In the forward pass, we need to calculate value for each node and then find the max.

Therefore, within one state, Time is $O(n^2)$

Since there are N states, Time is $O(Nn^2)$ #