* Parts of speech tagging

The main objective is to identify which grammatical categories do the word belong to.

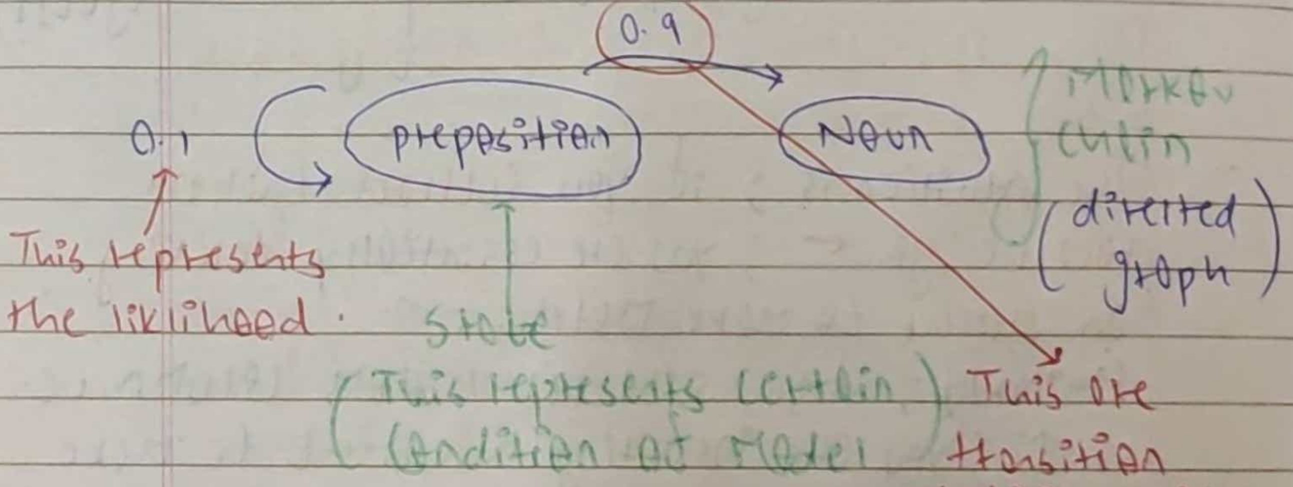
eg. Thwarting my plans ? → question (punctuation mark)

verb pronoun noun

We use Markov chain idea here;

Markov Model is a stochastic model (includes probability & randomness) which states that conditional probability distribution of future states, depends only on present state and NOT states preceding it.

eg. Chichi is wife of [ ] Noun? verb?
preposition



we build a transition matrix, very similar to adjacency matrix; but also with a start symbol since there is no one for first state.

eg.

	Noun	verb	preposition
$\boxed{S}$	0.5	0.2	0.3
Noun	0.1	0.7	0.1
verb	0.2	0.8	0.0
preposition	0.5	0.3	0.2

.... etc

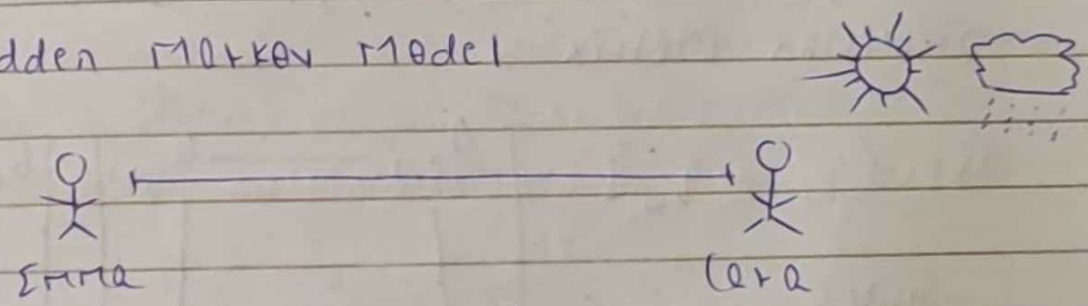
This is (N+1, N) matrix.

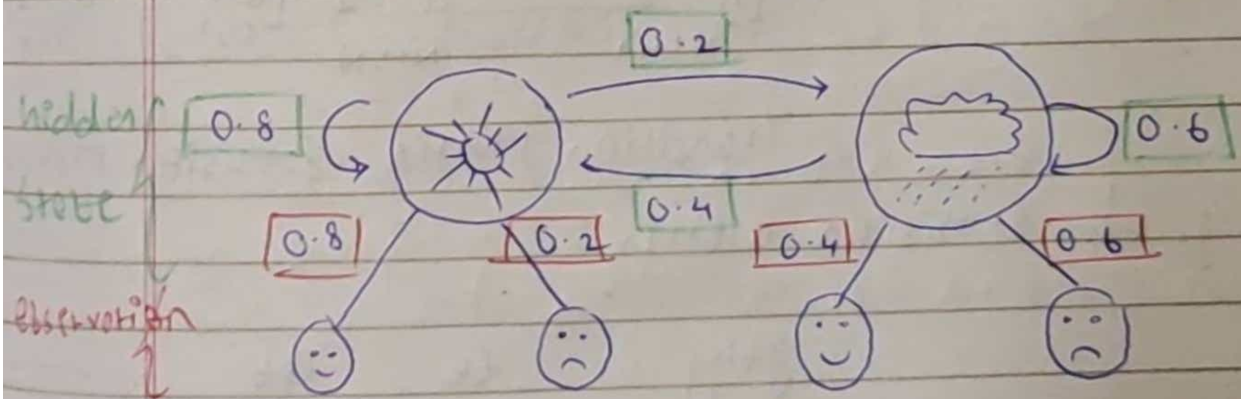
* POS Tagging & applications

POS tagging is used for preprocessing data. It is applied in:-

1) Name entity recognition
2) Q&A system
3) Word sense disambiguation
4) Chatbots

* Hidden Markov Model



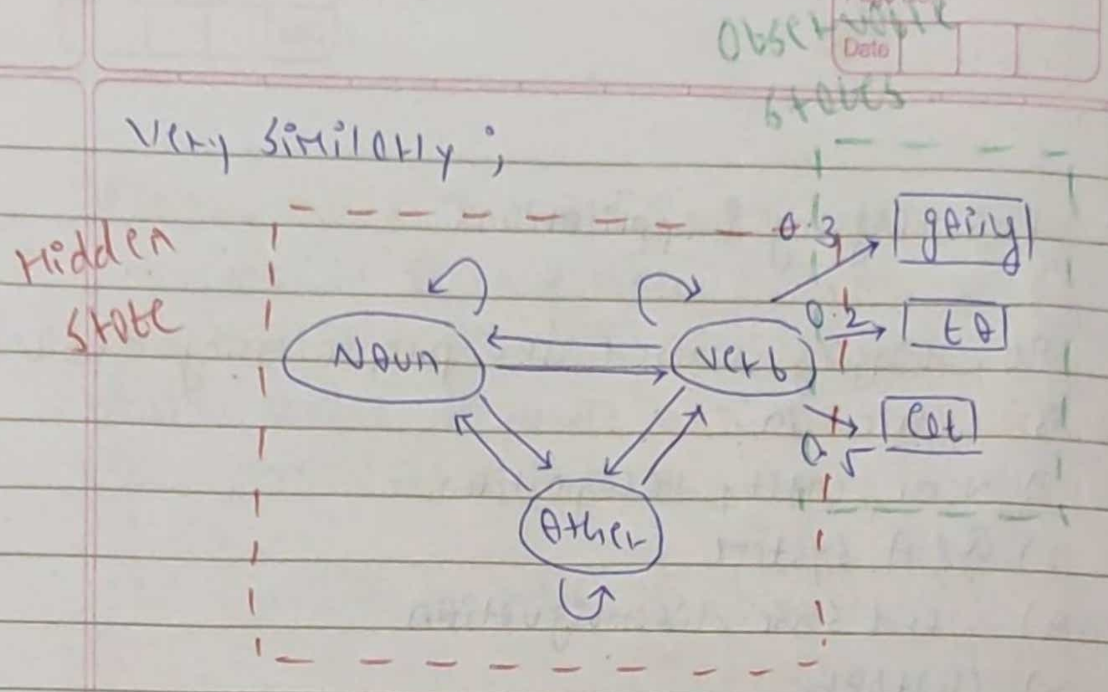


Emma can only observe Cara's mood [Happy or Sad]. This state is called observation. But, Emma has to determine weather [Sunny, rainy] which are hidden state. So this becomes a hidden Markov Model.

□ ⟶ Transition Probability

□ ⟶ Emmision Probability.

Very similarly;



Thus; we have states, Transition Matrix, Emmision Matrix.

$Q = \{q_1 — q_N\}$

States

$$\begin{bmatrix} a_{1,1} & — & a_{1,N} \\ | & & | \\ a_{N+1,1} & — & a_{N+1,N} \end{bmatrix} \quad \begin{bmatrix} b_{1,1} & — & b_{1,V} \\ | & & | \\ b_{N,1} & — & b_{N,V} \end{bmatrix}$$

Transition Matrix Emmision Matrix

g. of. Emmision Matrix:

	going	to	eat	- - -
Noun	0.5	-	-	
verb	-	-	-	
others	-	-	-	

The idea is; imagine case: 🙂 ☹️. On the basis of observation, we may conclude: ☀️ 🌧️. But it is not very likely for next day to be rainy if today is sunny. Thus, we need to use both Transition Matrix and Emmision Matrix..

Part 1: filling the probabilities.

For transmission Matrix, we need to lookup data & find freq.. Then accordingly, we convert it to probability.

eg.

S → S	8	8/10
S → R	2	2/10
R → S	2	2/5
R → R	3	3/5

	S	R
S	0.8	0.2
R	0.4	0.6

The row wise addition should be 1.

Similarly, for Emmision Matrix;

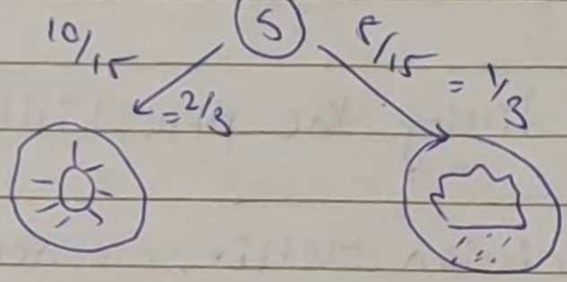
S → H	8	8/10
S → sad	2	2/10
R → H	2	2/5
R → sad	3	3/5

	Happy	sad
S	0.8	0.2
R	0.4	0.6

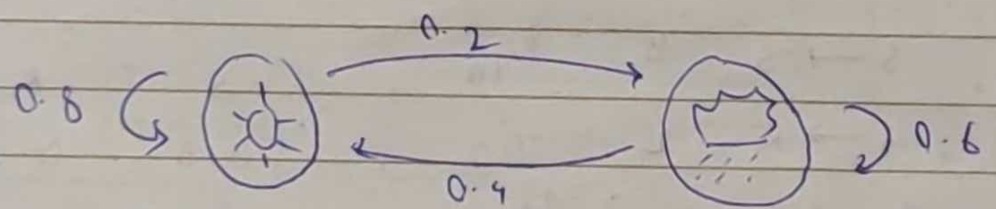
The row wise addition should be 1.

Part 2 : Finding start probability, i.e. Cara doesn't tell us if its happy / sad mood.

Approach 1 : look at data ; if there are 10 sunny days & 5 rainy days, we conclude that



Approach 2 : From transition probability.



$$\therefore \quad S = 0.8S + 0.4R$$
$$R = 0.2S + 0.6R$$
also ; $S + R = 1$ (extra eq^n) } solve & get answer for S & R.

Part 3 : Finding if it sunny or rainy, given Cara tells she is happy or sad.

Here, we use Bayes Theorem.

Part 4: If Cara says, she is Happy-Sad, find the seq of weather.

Since we are talking about 2 days; there are 4 probabilities

S-S	S-R	R-S	R-R
H, S	H, S	H, S	H, S
0.8↑ ↑0.2	0.8↑ ↑0.6	0.4↑ ↑0.2	0.4↑ ↑0.6
$S \xrightarrow{0.8} S$	$S \xrightarrow{0.2} R$	$R \xrightarrow{0.4} S$	$R \xrightarrow{0.6} R$
$\boxed{0.67}$	$\boxed{0.67}$	$\boxed{0.33}$	$\boxed{0.33}$
⇒ $\boxed{0.086}$	⇒ 0.064	⇒ 0.01	⇒ 0.048

Thus, it is most likely to be sunny-sunny.

The only problem here is we need to check 2^n paths for n days. Hence, we use Viterbi Algorithm, which is DP.

Note: for part 1; we also perform smoothing. i.e. to avoid zeros, we add ε to each cell & now division is done from total freq + $N^* \varepsilon$.

The algorithm comprises of 3 main steps:

1) initialization: Matrix C & D are initialized
2) Forward pass: DP tabular approach
3) Backward pass: Reconstruction of path.

Here, we will use Auxilary Matrix C & D.

C Matrix holds the optimal answers to subproblems & hence computing probability.

D Matrix is used to Reconstruct the path.

we fill this row for initialization

Step 1:

C =

	w_1	-----	w_N
t_1	c_{11}		
⋮	c_{12}		
t_N	c_{13}		

$$C_{i,1} = \underbrace{\boxed{\pi_i}}_{\text{prior probability}} * \underbrace{\boxed{b_{i,\,cindex(w_1)}}}_{\text{emission probability of that word}}$$

D =

	w1	-----	wN
t1	0		
⋮	0		
tN	0		

$D_{i,1} = \boxed{0}$

since there are no preceding states yet.

you choose the maximum of all these options.

Step 2:

'transmission probability from 'k' in prev column to 'i' in curr column

$$c_{i,j} = \max_{k} \left[\boxed{c_{k,j-1}} * \boxed{a_{k,i}} * \boxed{b_{i,\text{cindex}(w_j)}} \right]$$

The probability stored in prev column (chaining)

emmision probability of that word

$$d_{i,j} = \boxed{\underset{k}{\text{ArgMax}}} \left[c_{k,j-1} * a_{k,i} * b_{i,\text{cindex}(w_j)} \right]$$

This is simply the 'k' which give best probability after chaining.

Step 3:

for reconstruction; simply look into last column of C Matrix & choose cell with highest probability. This cell is mapped to D. Thus, choose a cell in D which corresponds to highest probability in C. Then keep going back.

eg.

	w1	w2	w3	w4	w5
t1	0	1	$\boxed{3}$	2	$\boxed{3}$
t2	$\boxed{0}$	2	4	1	3
t3	0	$\boxed{2}$	4	$\boxed{1}$	4
t4	0	4	4	3	1

the part 1 means go to t1.

assume this cell corresponds to best probab in C.

Path: $\pi \rightarrow 2 \rightarrow 3 \rightarrow 1 \rightarrow 3 \rightarrow 1$ ← cause the first one was row 1