# NLP -

(preprocessing step of NLP pipeline)

* parts of speech taging (pos tagging in NLP)

~> in simple words, pos tagging is a task of labelling each word in a sentence with its appropriate part of speech. In traddtional grammer, a part of speech is a category of words that have similar grammatical properties.

→ { Noun   proNoun   Adjective } ⟹ checking for particular POS in sentence.

why	not	tell	some one?	
↓	↓	↓	↓	↘
adverb	adverb	verb	noun	Punctuation marks, sentence closer
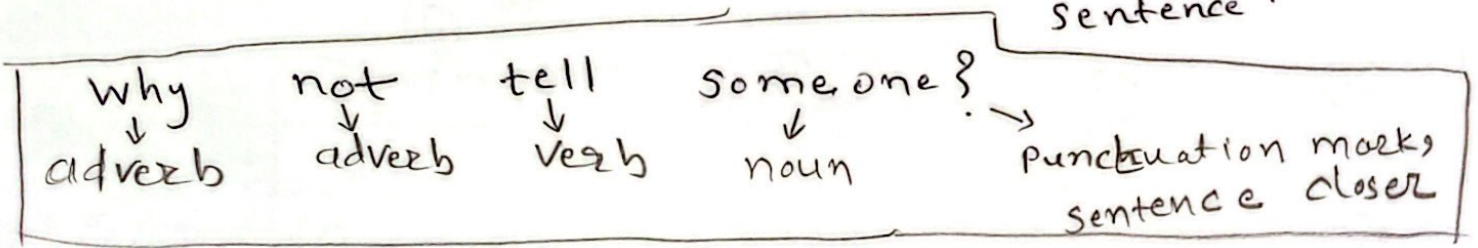
⟹ Applications

1) Named entity Recognition (entity extration from sentence)

EX - I would like to PVR Cinemas.

⤷ entity

EX - 2. Rahul will go to Delhi at 30th march.

↪ entity.

2) Question Answering System.
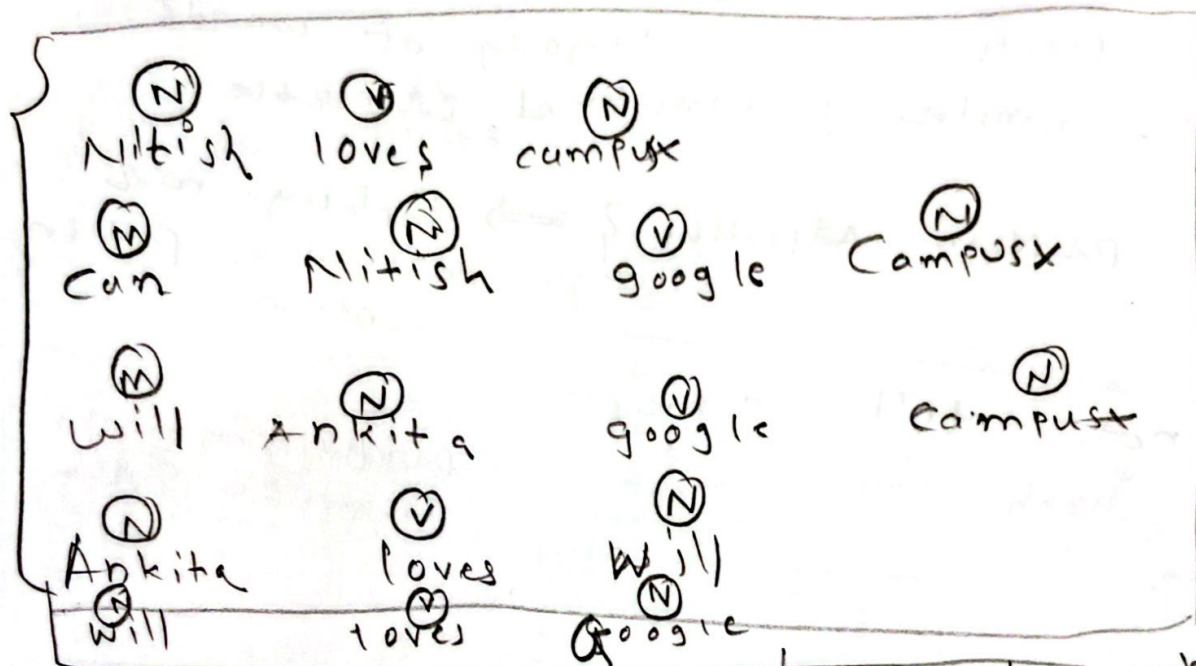
3) word sense disambiguation ⟹ { EX   I left the room   left of the room

(एक ही word अलग अलग meaning अगर
एक ही sentence मी use thay to decoding
...)

4) chat bots:
5) or Any complex NLP systems:

$\longrightarrow$ How POS Tagging works

[Will will google campux]

(N) Nitish (V) loves (N) campux

(M) Can (N) Nitish (V) google (N) Campusx

(M) will (N) Ankita (V) google (N) campusx

(N) Ankita (V) loves (N) Will
(M) loves (N) Google

→ Probability of 'Nitish' word as Noun

Emission Prob. Table	unique words	Noun	Model	Verb
	Nitish	2/10	0	0
	loves	0	0	3/5
				0
	campusx	3/10	0	0
	google	1/10	0	2/5
	Will	2/10	1/20	0
	Ankita	2/10	0	0
	Can	0	1/2	0
Total		10	2	5

→ probability of 'loves' words as verb

→ probability of 'Will' word as Model

This all Probability called Emission Probability

S (N) Nitish   (V) loves   (N) campusx   E

S (M) can   (N) Nitish   (V) google   (N) campusx   E

S (M) will   (N) Ankita   (V) google   (N) campusx   E

(S (N) Ankita   (V) loves   (N) will   E

S (N) will   (V) loves   (N) google.   E

	N	M	V	E	total
S	3/5	2/5	0	0	5
N	0	0	5/10	5/10	10
M	2/2	0	0	0	2
V	5/5	0	0	0	5

transition probability.

Transition probability

Nitish   loves   campusx   will   Ankita   can

will   Will   google   campusX

$1/5$ $1/5$ $1/10$ $3/10$

S $\quad\frac{3}{5}$ (N) $\quad$ 0 $\quad$ (N) $\quad$ 0 $\quad$ (N) $\quad$ 0 $\quad$ (N) $\quad\frac{5}{10}$ E = 0

$1/5$ $1/5$ $1/10$ 0

S $\quad\frac{3}{5}$ (N) $\quad$ 0 $\quad$ (N) $\quad$ 0 $\quad$ (N) $\quad$ 0 $\quad$ (M) $\quad$ 0 $\quad$ E = 0

$$\begin{bmatrix} & & \text{Do combinations} \\ & & \downarrow \\ S & \frac{1}{2} & \frac{1}{5} \quad \frac{2}{5} \quad \frac{1}{10} \\ & (M) \quad (N) \quad (V) \quad (N) & E = \underline{\text{Non Zero}} \\ & \frac{2}{5} \quad 1 \quad \frac{1}{2} \quad 1 \quad \frac{1}{2} \end{bmatrix}$$
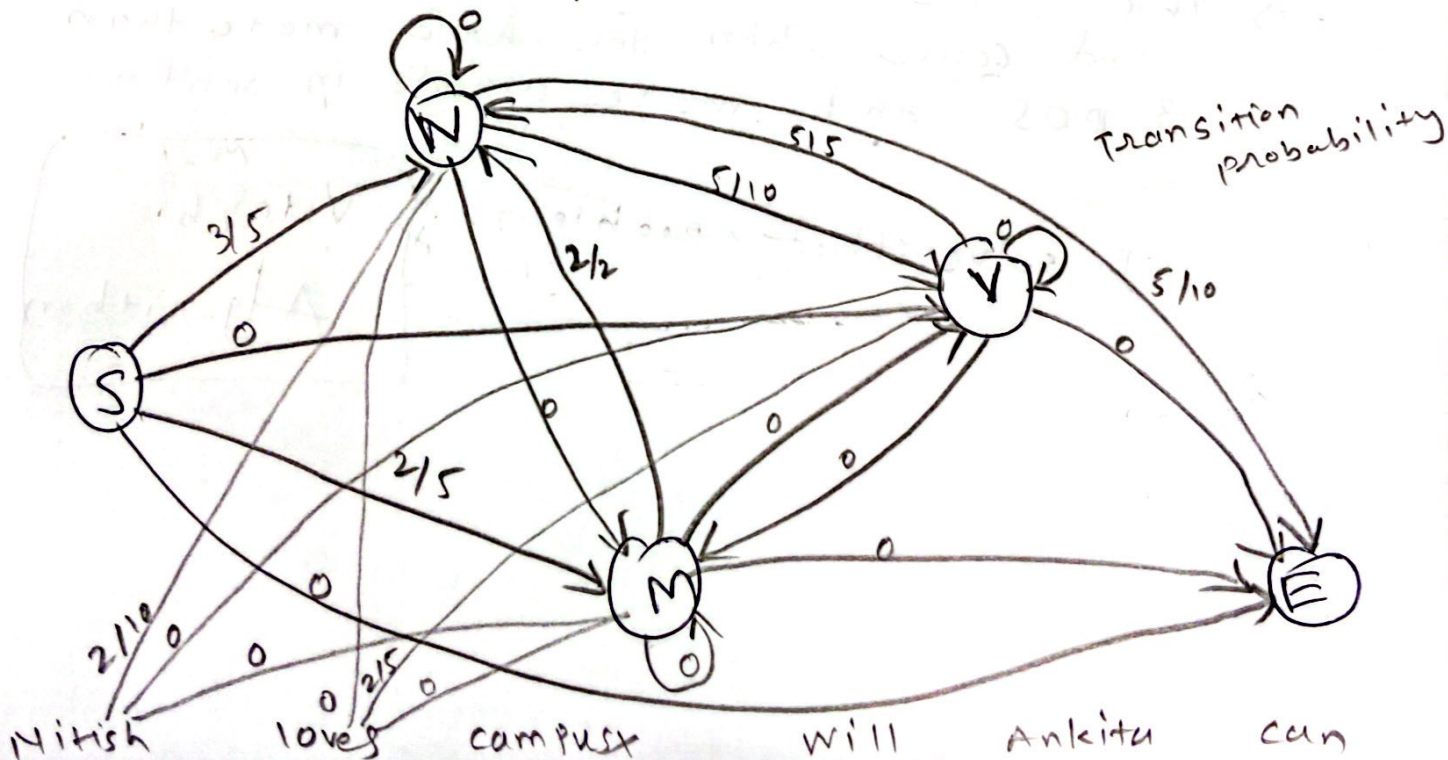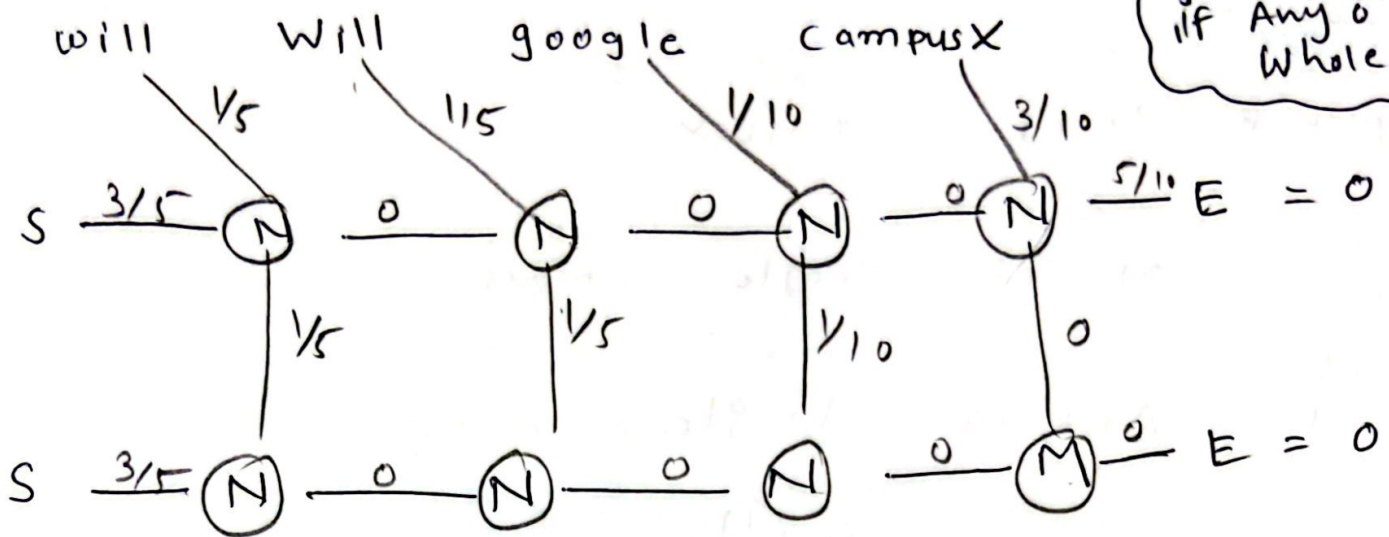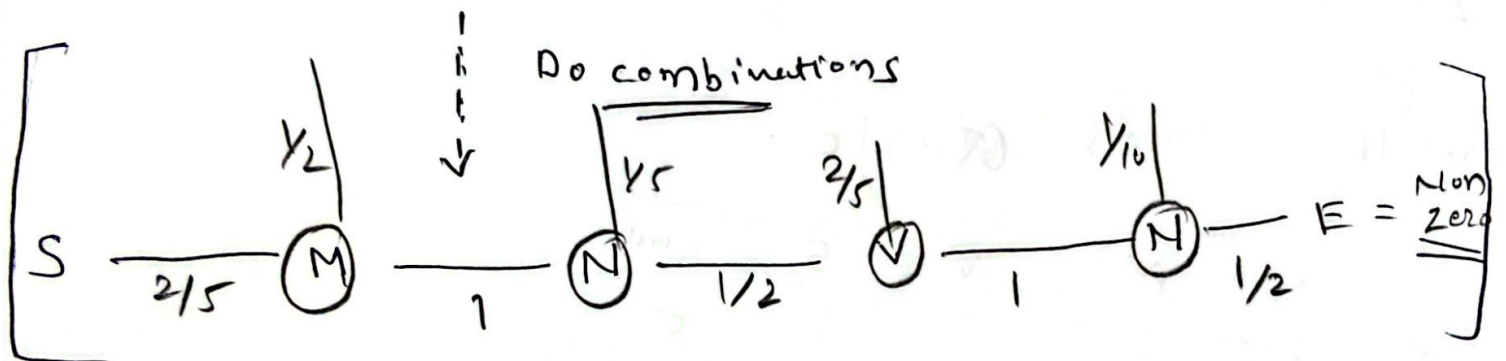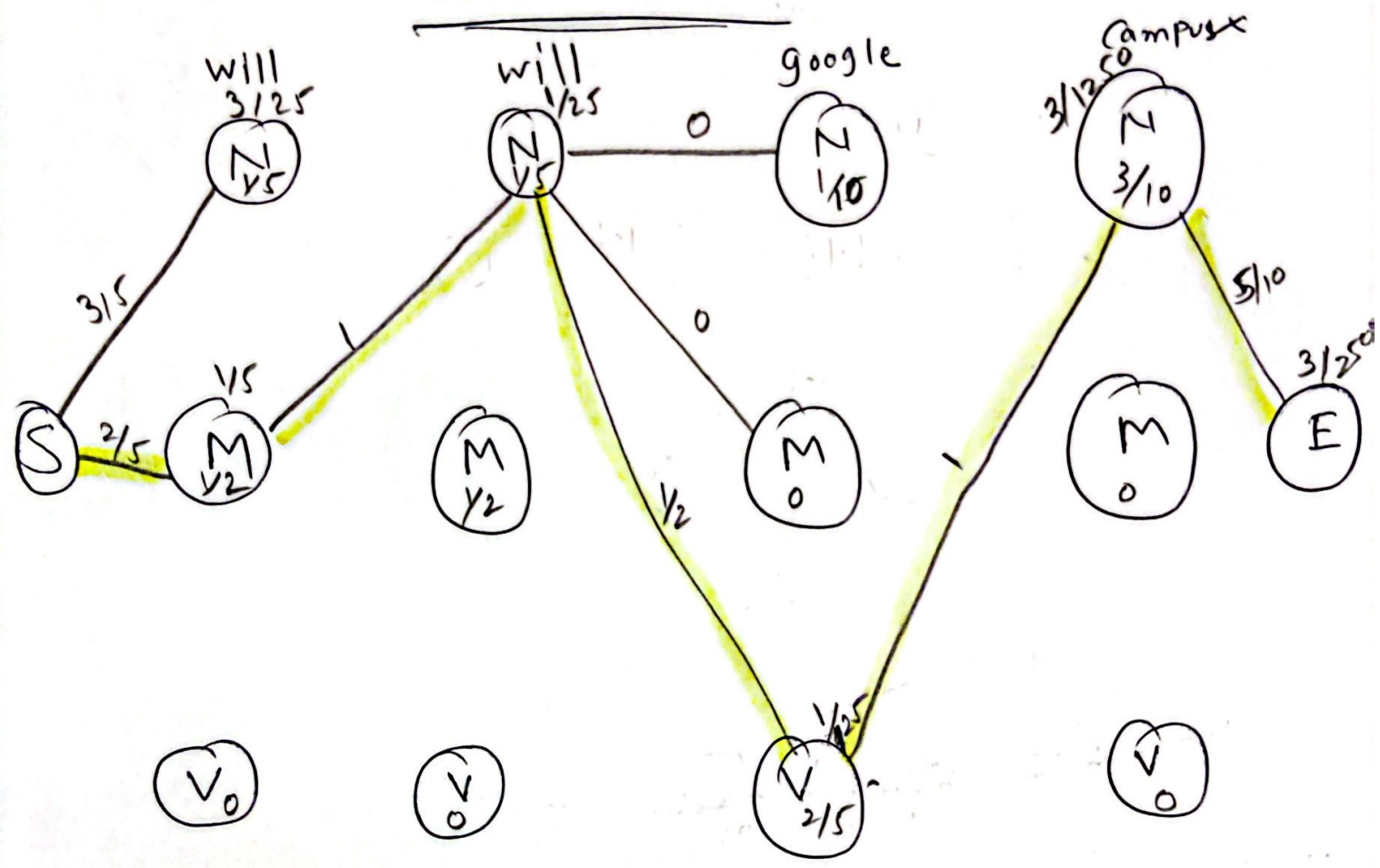
Note: this will have $3^4 = 81$ combinations
since model, verb, Noun = 3
and Number of words in sentence = 4

⟹ This whole combination will be healthy in end cause when you have more than 3 POS and more words in sentence
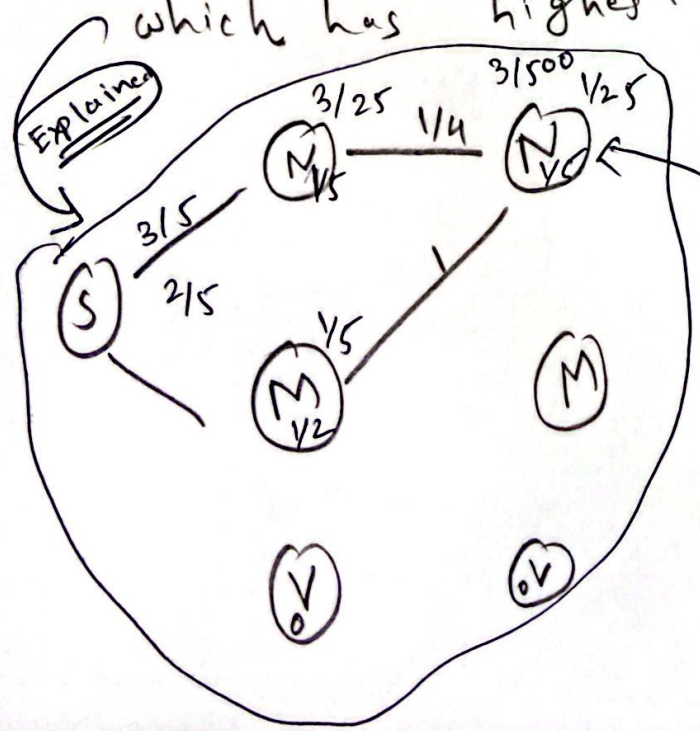
To solve that problem ⟹ Viterbi Algorithm

Viterbi Algorithm

Top diagram labels:

will 3/25 — will 1/25 — google 0 — Campus 3/13 co

N 1/5 — N 1/5 — N 1/10 — N 3/10

3/5 — 1/5 — 0 — 5/10 — 3/25 co

S — 2/5 M 1/2 — M 1/2 — M 0 — M 0 — E

1/2 — 1/25

V 0 — V 0 — V 2/5 — V 0

→ (if 0 don't go for that node)

  ⇒ (multiply every transition & emission probability)

⇒ if you get two Paths then choose path, which has highest probability.

Bottom diagram (Explained):

3/25 — 1/4 — 3/500 1/25

M 1/5 — N 1/6

3/5 — 3/25

S — 2/5 — M 1/2 1/5

M 1/5

V 0 — V 0

for this we get two Paths

$S-N-N = 3/500$

$S-M-N = 1/25$

we will choose this path and remove other path since it has highest probability.