NLP

P1: Grammatical Error Detection / Correction.

'This is my brother's cat

space

Here space might be req. if translating to another language

Token: Single surface form word
My cat is afraid of other cats
cats ≠ cat
→ changing the noun we were referring to (plural noun now)

Suffixation - morphological changes happen at the end (plural affixation)

Derivational morphology: Morphology - semantics change morphology - syntactic"

Type: vocabulary word
- cat is the type for 'cat' and 'cats'

If Token/Type ratio is high means langauge allows agglutination

Agglutinative : keep adding morpheme to determine the meanings but morphemes remain unchanged after unions.

Phoneme : unit of sound that distinguish one word from another.

ladk|a ladkion

stem. suffix

Challenges in tokenization

① Hyphenation

we would deal with the state-of-the-art

noun

whether to replace hyphen or retain it?

commander-in-chief

Scanned by CamScanner

(2) Number

1/2 cup Milk.
1,000 dollars.

Sentencification (Break for sentences)

* Identify '.' correctly.

N-Grams and zipf's law

- Shanon's game: Predict the next word, given (n-1) words

- Jensen's smoothing
- Laplace smoothing

$$P(O_1, O_2 \ldots O_k) = P(O_k | O_{k-1}) P(O_{k-1} | O_{k-2}) \ldots$$
$$= P(O_1) \prod_i P(O_i | O_{i-1})$$

* Entropy (Slides)

* Zipf Law

Date
Page

Laplace Smoothing

The oldest method

$$P_{Lap}(w_1 \ldots w_n) = \frac{C(w_1 \ldots w_n) + 1}{N + B}$$

Lidstone's law

$$P_{Lid}(w_1, \ldots w_n) = \frac{C(w_1 \ldots w_n) + \lambda}{N + B\lambda}$$

Jefferys - Perks law

Set λ as $1/2$

Expectation of λ that maximizes above question.

Held out estimation

N_r = no of n grams with. r freq in 'training' data

Scanned by CamScanner

HMM POS

$q = q_1 q_2 \ldots q_n$ N states

$A = a_{11} \ldots a_{ij} \ldots a_{NN}$

(Transition prob) $\sum_{j=1}^{N} a_{ij} = 1 \; \forall i$

$O = o_1 o_2 \ldots o_T$

$B = b_i(o_t)$ Observation likelihood
(emission probability)
o_t generated by q_i

- $P(q_i | q_1 \ldots q_{i-1}) = P(q_i | q_{i-1})$
- $P(o_i | q_1 \ldots q_i \ldots q_T, o_1 \ldots o_i \ldots o_T) = P(o_i | q_i)$

A matrix (transition probability) $P(t_i | t_{i-1}) = \dfrac{C(t_{i-1}, t_i)}{C(t_{i-1})}$

B (Em prob) $P(w_i | t_i) = \dfrac{C(t_i, w_i)}{C(t_i)}$

$t_1^n \Rightarrow$ given n words, tag sequence (w_1^n)

$$\hat{t}_1^n = \underset{t_1^n}{\text{argmax}} \; P(t_1^n | w_1^n)$$

$$= \text{argmax} \; \frac{P(w_1^n | t_1^n) \times P(t_1^n)}{P(w_1^n)}$$

$\leftarrow$ not req.

$$P(w_1^n \mid t_1^n) \approx \prod_{i=1}^{n} P(w_i \mid t_i)$$

Bigram assumption: $P(t_1^n) \approx \prod_{i=1}^{n} P(t_i \mid t_{i-1})$

$$\therefore t_1^n = \operatorname{argmax} \prod_{i=1}^{n^{\max}} \underbrace{P(w_i \mid t_i)}_{\text{em prob}} \underbrace{P(t_i \mid t_{i-1})}_{\text{trans prob}}$$

Forward Inference

$$\alpha_t(x_t) = P(x_t, y_1, \ldots y_t)$$

$$\alpha_1(x_1) = P(x_1, y_1) = P(x_1) P(y_1 \mid x_1)$$

$$\alpha_2(x_2) = P(x_2, y_1, y_2) = \sum P(x_1, y_1) P(x_2 \mid x_1) P(y_2 \mid x_2)$$
$$= \sum_{x_1} \alpha_1(x_1) P(x_2 \mid x_1) P(y_2 \mid x_2)$$

$$\therefore \alpha_{t+1}(x_{t+1}) = \sum_{x_t} \alpha_t(x_t) P(x_{t+1} \mid x_t) P(y_{t+1} \mid x_{t+1})$$

Backward Inference

$$\beta_t(x_t) = P(y_{t+1}, \ldots y_T \mid x_t)$$

$$P(\{\} \mid x_T) = 1 = \beta_T(x_T)$$

$$\beta_{t-1}(x_{t-1}) = P(y_t \ldots y_T \mid x_{t-1})$$

$$= \sum_{x_t} P(x_t \mid x_{t-1}) P(y_t, y_{t+1} \ldots y_T \mid x_t)$$

$$= \sum_{x_t} P(x_t \mid x_{t-1}) P(y_t \mid x_t) P(y_{t+1} \ldots y_T \mid x_t)$$

$$= \sum P(x_t \mid x_{t-1}) P(y_t \mid x_t) \beta_t(x_t)$$

$$\alpha_t(x_t) \beta_t(x_t) = P(x_t, y_1, \ldots, y_t) P(y_{t+1} \ldots y_T \mid x_t) \ldots$$

$$= P(x_t, Y) \propto P(x_t \mid Y)$$

$$P(x_t, x_{t+1} \mid Y) = \frac{P(x_t, x_{t+1}, (y_1 \ldots y_t), (y_{t+1}), (y_{t+2} \ldots y_T))}{P(Y)}$$

$$= \frac{P(x_t, y_1 \ldots y_t) P(x_{t+1} \mid x_t) P(y_{t+2} \ldots y_T \mid x_{t+1}) P(y_{t+1} \mid x_{t+1})}{\sum_{x_T} P(x_t, Y)}$$

$$= \frac{\alpha_t(x_t) P(x_{t+1} \mid x_t) \beta_{t+1}(x_{t+1}) P(y_{t+1} \mid x_{t+1})}{\sum_{x_T} \alpha_T(x_T)}$$

Saumyak

Brevity ⟷ Ambiguity

1946 – Stored program computer.

Rule based — Limited domains; large corpus, knowledge

ANN methods: Huge domain.

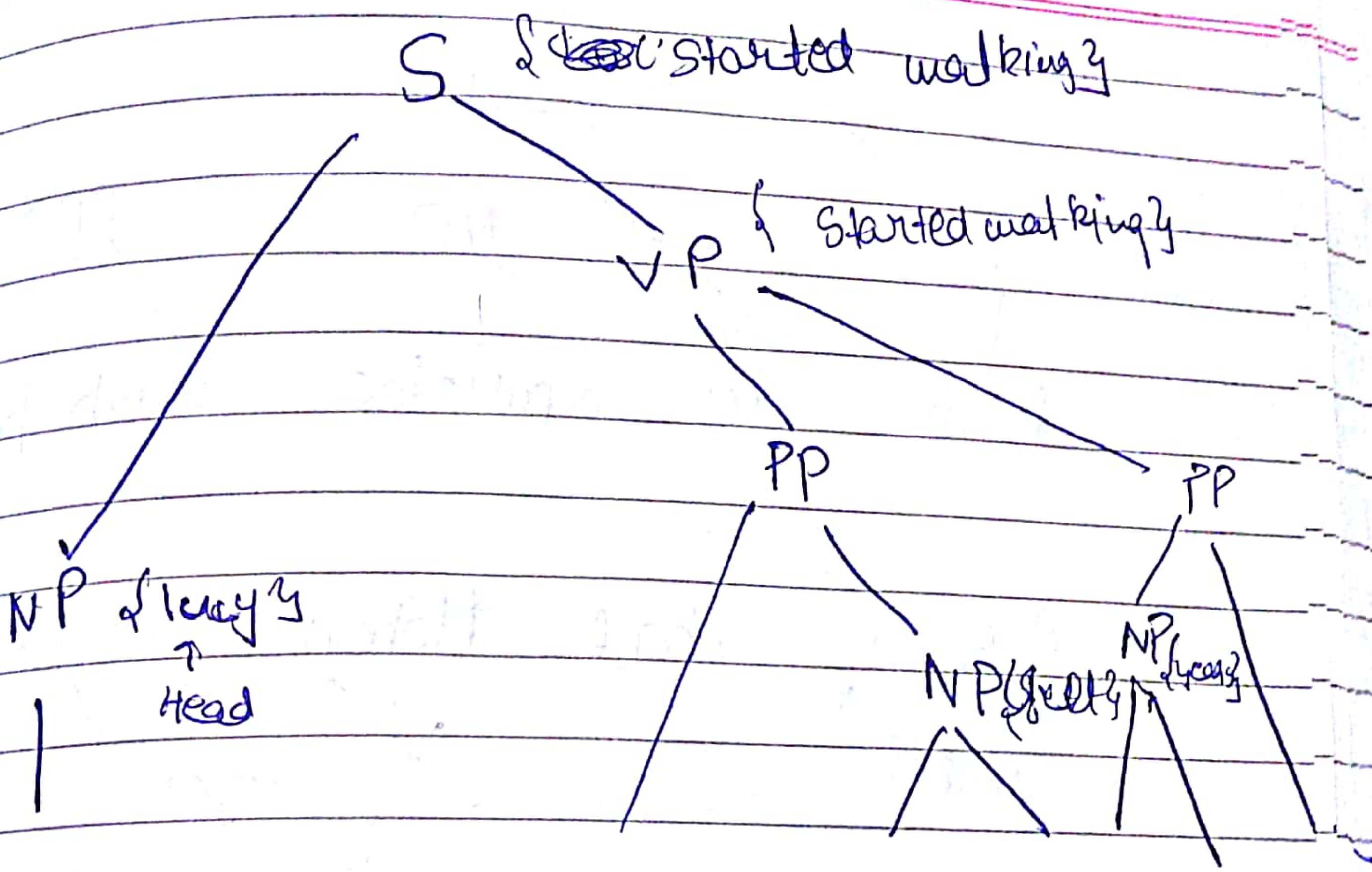
obj — action

Lucy started walking on two feet 5 million years ago.

obj — obj

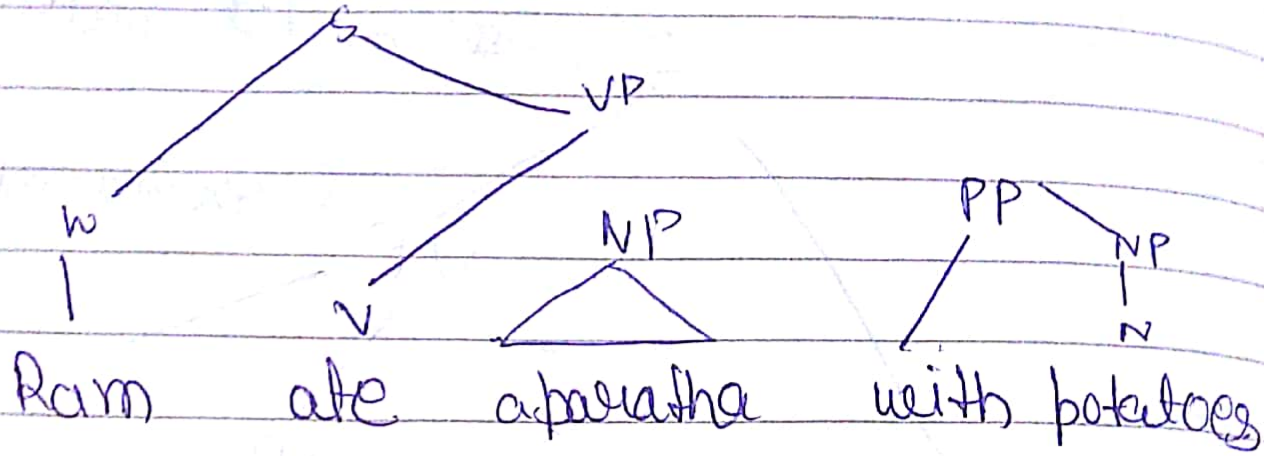
POS → Chunking → parse tree
tagging

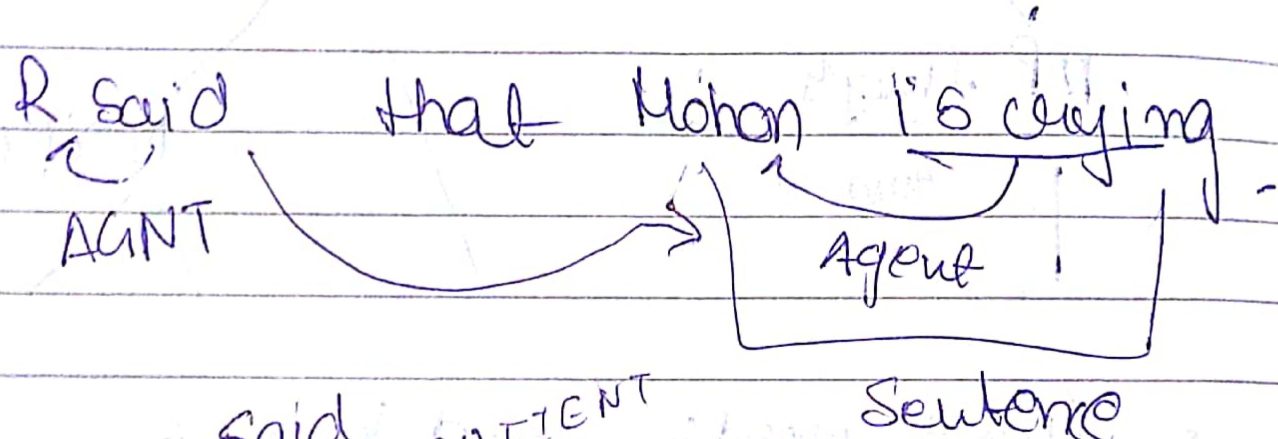
POS → Lucy started walking on two feet
n v v pro n n
Chunking → 5 million years ago
n n n

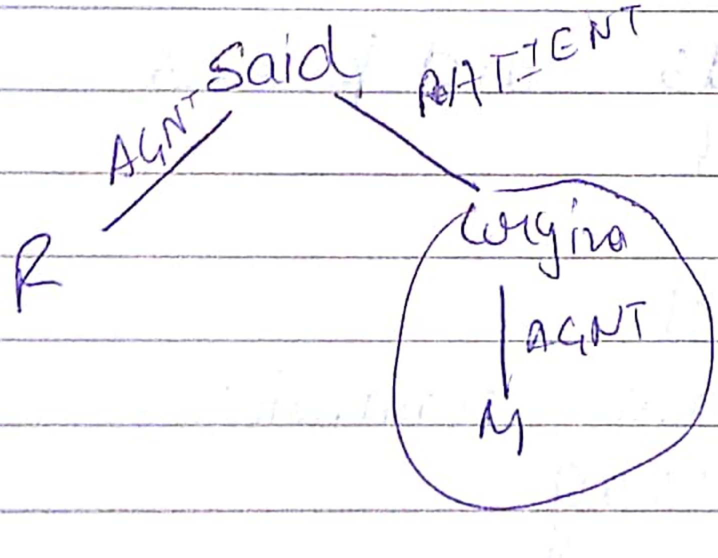


- Group words and phrases.
- Hierarchical

1) Knowledge based
- Rules, grammar, semantics
2) Statisical learning
- Features
3) Neural net based
- Data


S
VP
N
NP
PP
NP
V
N
Ram ate a paratha with potatoes


R said that Mohan is crying
AGNT
Agent
Sentence


said
AGNT
PATIENT
R
crying
AGNT
M

Restrictive clause

S1 That book is lost

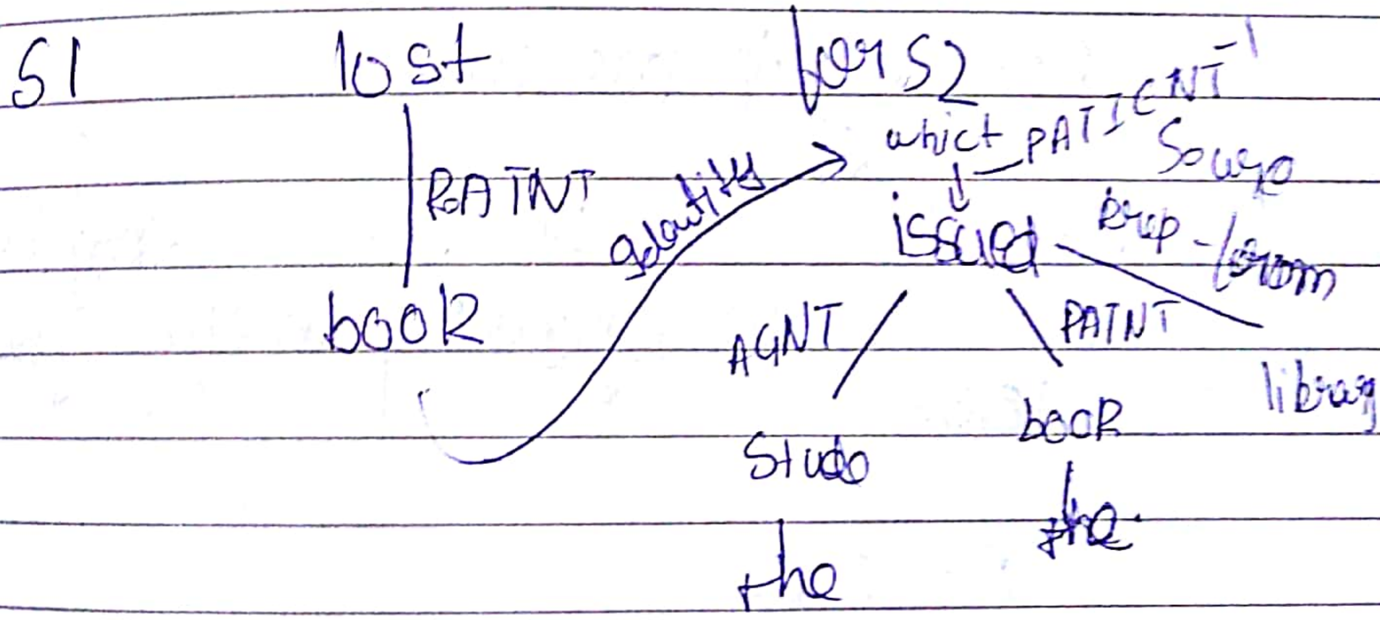
S2 The student issued the book from the liabrary.

to stop from repeating use a pronoun.

The book [ ] is lost

→ co referent

⇒ The book which the student issued from the liabrary is lost.



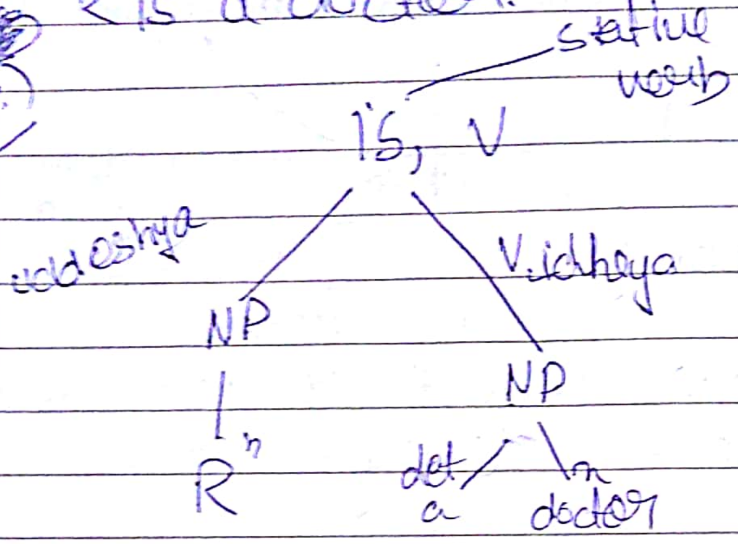
$S' \to NP'\ VP'$ allows for missing NP

$VP' \to [NP^*]\ [NP']\ PP^*$

Samyak

send notes

R is a doctor.



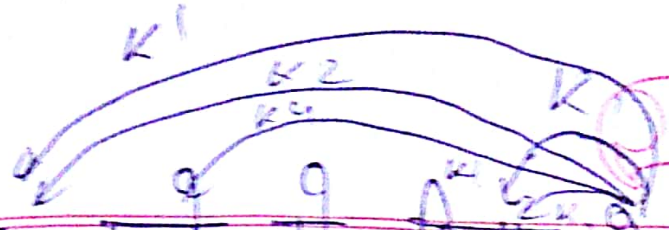
Bottom up parsing

Place the red block, ~~over the~~ besides the yellow cylinder, on the green cube

A B D C

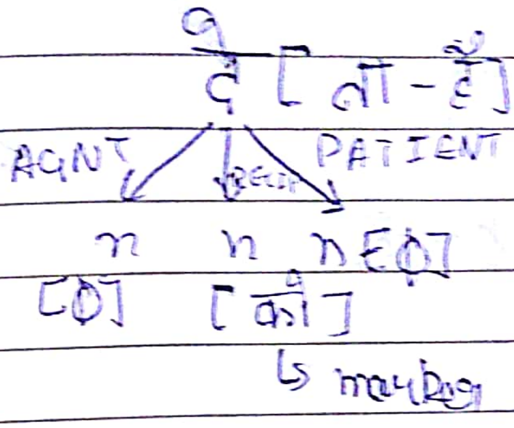
Y. → A B

Examine top of stack (open, rem-sent, rem-rule)
while stack not empty:
→ if top of stack ⇒ ((start), nil, ...) record guess,
If open can be reduced by rule R
then modify top of stack
('', '', R+1) and push (reduced open,
rem-sent, All) else if shift is possible
modify top of stack
else pop

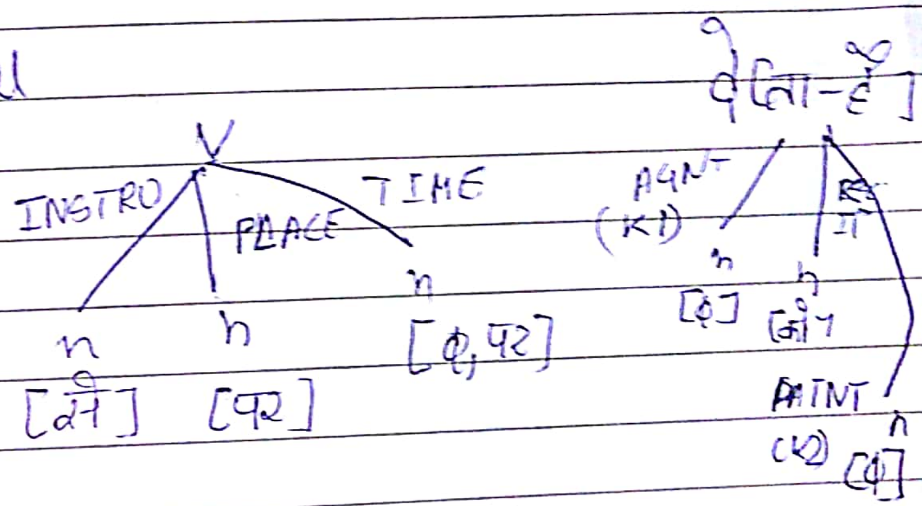


राम बच्चे को किताब देता है

n n postp n v vaux



Optional



Solution graph (parse tree)

C1 No more than one incoming edge
exactly one.

C2 For outgoing edges of a node, there should be exactly one labelled edge corresponding to mandatory edges

C3 At most one ... optional edges.

A solution graph is a subgraph is it satisfies C1, C2, C3

gf दै[धा]

k। $\phi \to$ कै