CS626: Speech, NLP and the Web

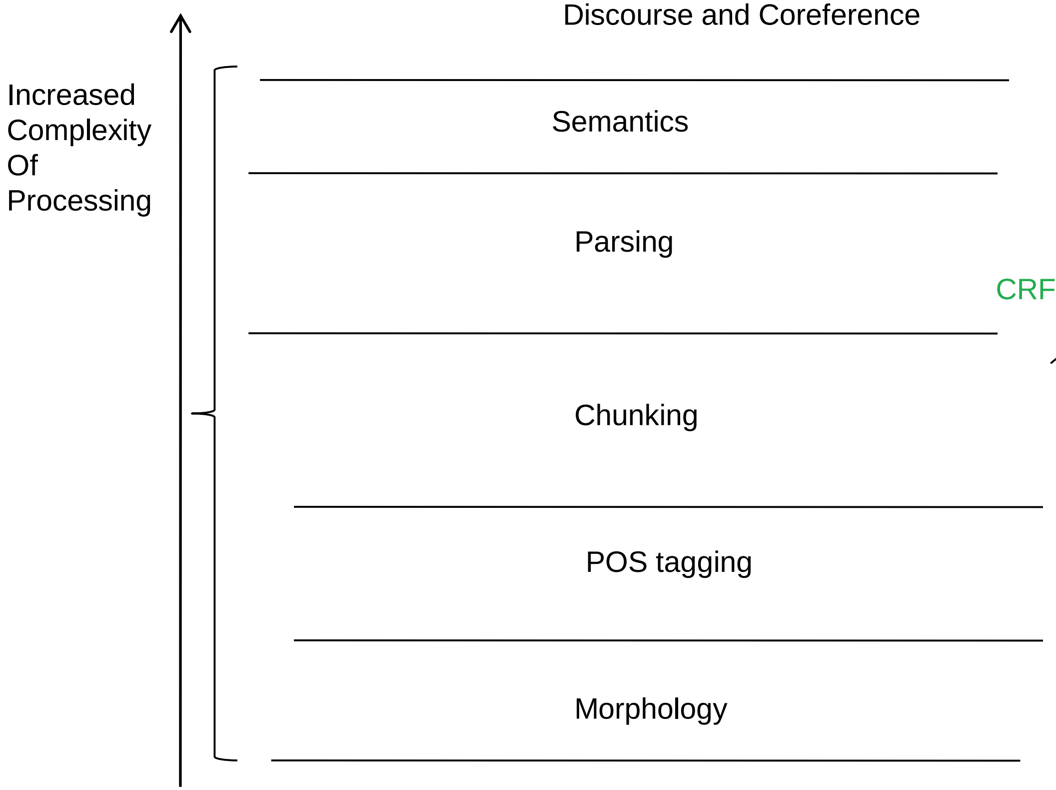
Deep Parsing

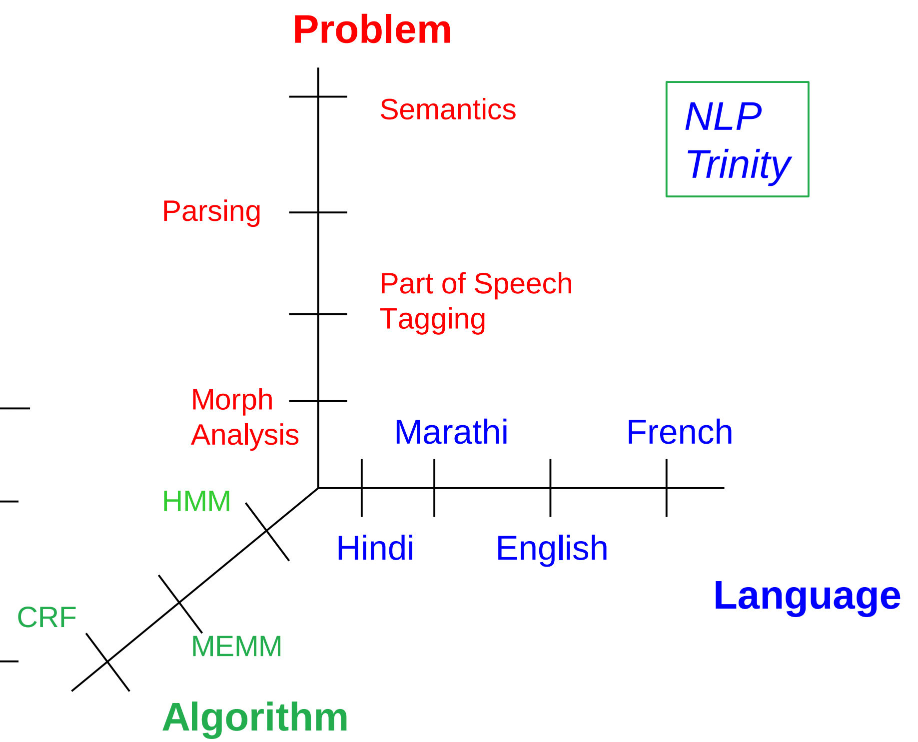
Pushpak Bhattacharyya
Computer Science and Engineering Department
IIT Bombay

Week of 14th September, 2020

Agenda for the week

- Need for parsing
- Two types of parsing: Constituency, Dependency
- Ambiguity in parsing
- Classical Algorithms for parsing
- Probabilistic parsing
- Neural Parsing


Increased
Complexity
Of
Processing
Discourse and Coreference
Semantics
Parsing
Chunking
POS tagging
Morphology


Problem
Semantics
Parsing
Part of Speech
Tagging
Morph
Analysis
NLP
Trinity
Marathi
French
Hindi
English
Language
HMM
CRF
MEMM
Algorithm

Chunking can be ambiguous

- Example: *The red rigid rolling round ball.*
- Correct labels are: *B I I I I I*
- But there can FRAGMENTATION
  - In general “rolling” is a verb
  - So this can potentially start a chunk with ‘B’

Entropy of a sentence

- Raw sentence has the highest entropy
- Entropy decreases as we go up the NLP layers
- Raw sentence > Morphologically processed > POS tagged > chunked > parsed > Semantic role labeled

Parsing the sentence, “The detective listened with a wooden face”

```
(ROOT
  (S
    (NP (DT The) (NN detective))
    (VP (VBD listened)
      (PP (IN with)
        (NP (DT a) (JJ wooden) (NN face))))
    (. .)))
```

```
det(detective-2, The-1)
nsubj(listened-3, detective-2)
root(Root-0, listened-3)
prep(listened-3, with-4)
det(face-7, a-5)
amod(face-7, wooden-6)
pobj(with-4, face-7)
```

Formal definition of sentiment/opinion: parsing built in the definition

- An opinion is a quintuple, $(e_i, a_{ij}, s_{ijkl}, h_k, t_l)$, where
  - e_i is the name of an entity,
  - a_{ij} is an aspect of e_i,
  - s_{ijkl} is the sentiment on aspect a_{ij} of entity e_i,
  - h_k is the opinion holder,
  - and t_l is the time when the opinion is expressed by h_k

- The sentiment s_{ijkl} is
  - positive, negative, or neutral, or
  - expressed with different strength /intensity levels, e.g., 1–5 stars as used by most review sits on the Web

- When an opinion is on the entity itself as a whole, the special aspect GENERAL is used to denote it. Here, e_i and a_{ij} together represent the opinion target.

Example

- “I loved the songs in the movie, though only the cast was liked by my brother”

Example (cntd.)

- Entity: *movie*
- Aspects: *songs, cast*
- Opinion holder: *I, brother*
- Time: *present (I), past (brother)*
- Opinioner-sentiment-aspect: *I-love-song, brother-like-cast*

Two kinds of parse representations: Constituency Vs. Dependency

Dependency Parsing

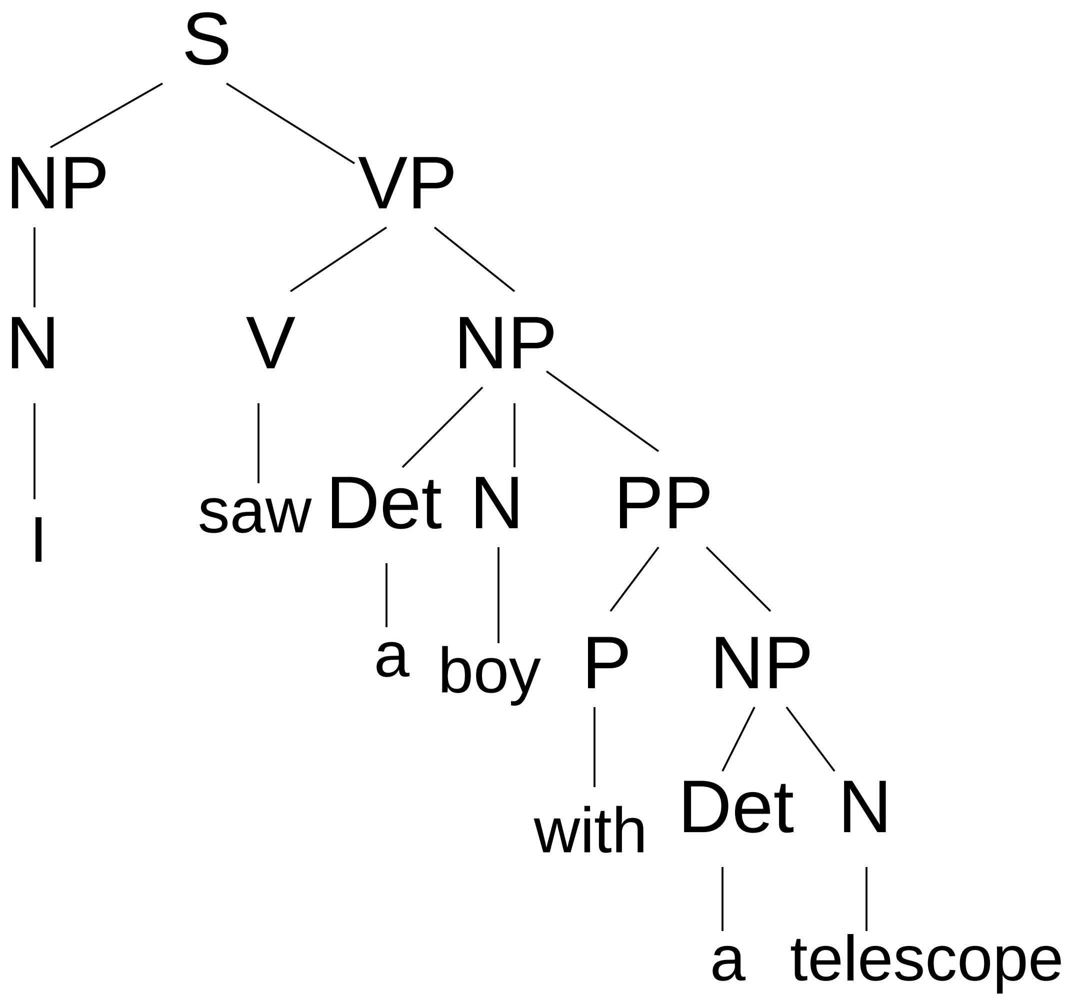
- Dependency approach is suitable for free word-order language

- Example : Hindi
  - राम ने श्याम को देखा (Ram ne Shyam ko dekha)
  - श्याम को राम ने देखा (Shyam ko Ram ne dekha)

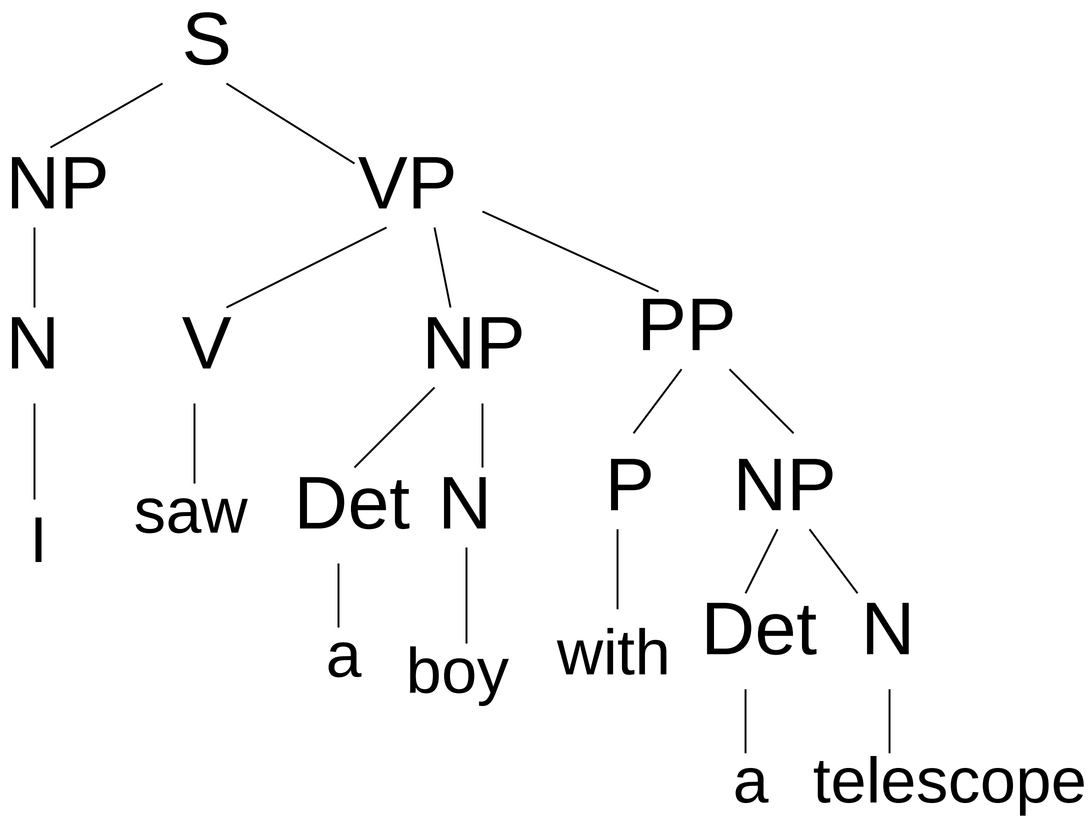
- One step closer to **Semantics**

Parsing Challenge: Structural Ambiguity

Constituency Parse Tree - 1



Constituency Parse Tree -2



Structural ambiguity

- Sentences can be ambiguous
  - Structural ambiguity
  - A sentence can have multiple parse trees.
- Example
  - *I saw a boy with a telescope*
    - Two possible meanings
      - I used the telescope to see the boy
      - I saw the boy who had a telescope
    - Two different constituent parse trees
- The correct meaning is determined by the attachment point of the PP “with a telescope”
  - determined by binding theory

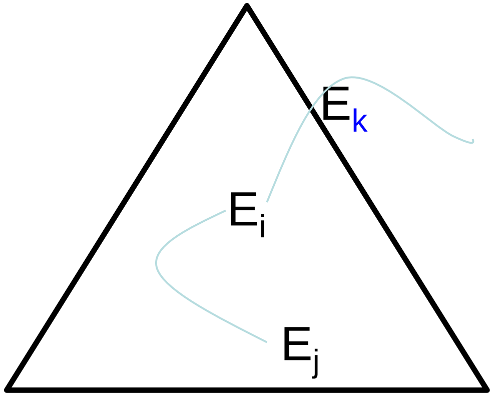
Binding Theory: Key aspects

- Binding theory concerns with
  - Syntactic restrictions on nominal references
  - Eg: The relation between pronoun and its antecedent
- Example
  - "**He** read a book to **himself.**".
    - Himself = reflexive pronoun
    - He = noun
- Three Key aspects
  - Class of nominal (Pronoun, anaphora, non-pronouns etc.)
  - Domain of binding (Local or non-local binding)
  - Structural condition on the syntactic relation between a nominal and its binder.

http://www.sas.rochester.edu/lin/sites/asudeh/pdf/asudeh-dalrymple06-ell2.pdf [Last accessed:15-sept-2020]

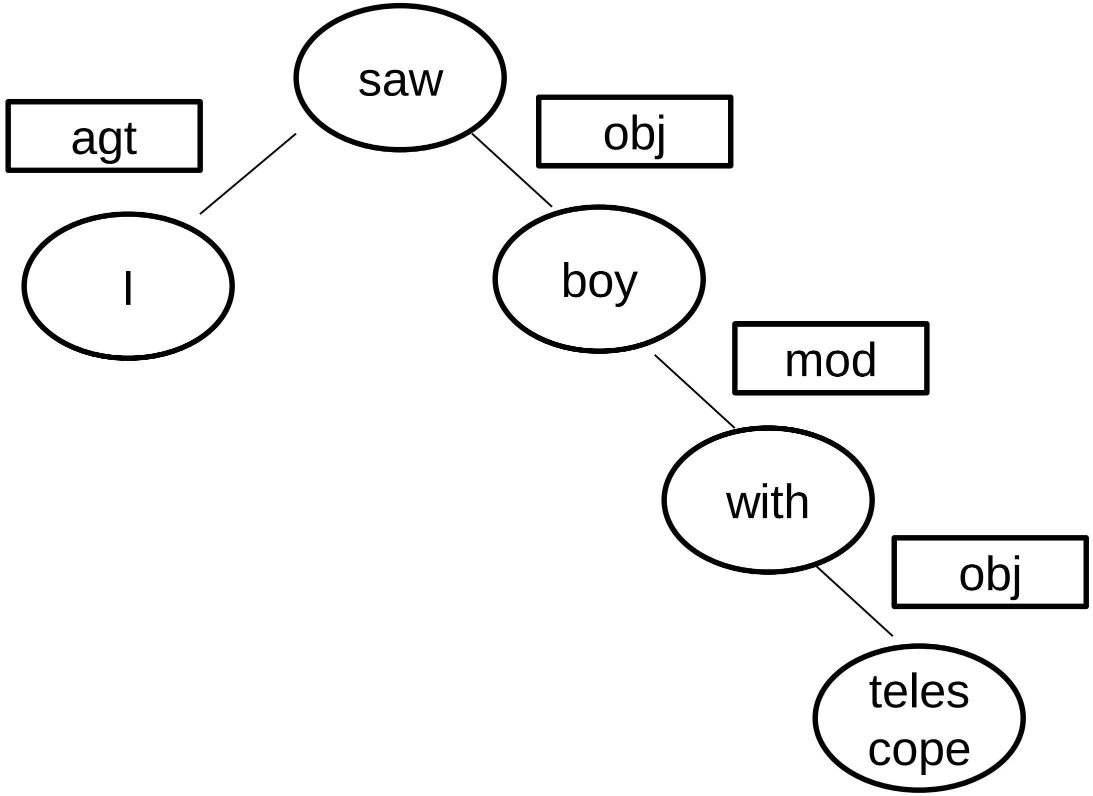
Parse Tree

- Within a sub-tree entities bind together more than they do with entities outside the sub-tree

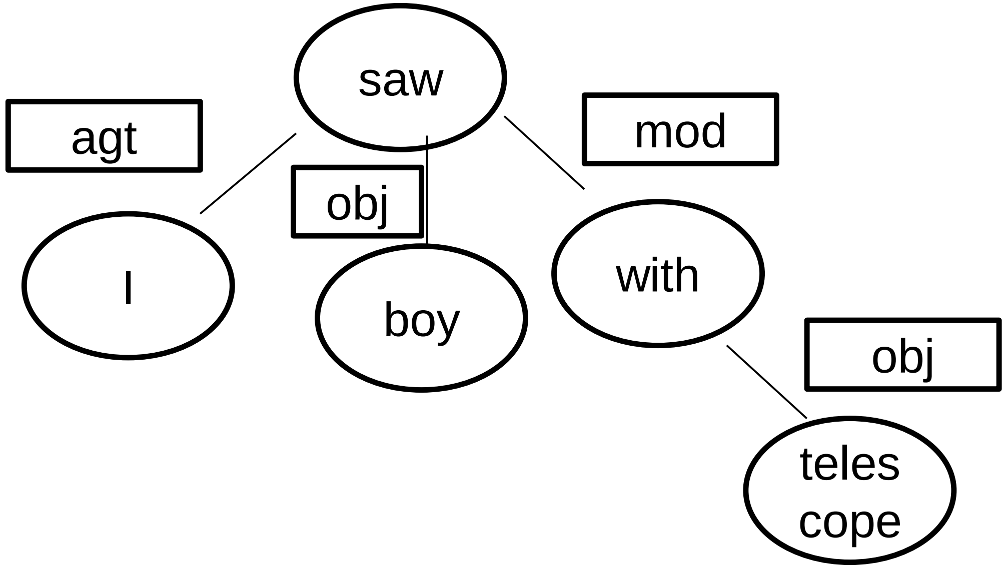


- Strength (E_i, E_j) > Strength (E_i, E_k)

Dependency Parse Tree - 1



Dependency Parse Tree - 2



Verb centric view of Sentence

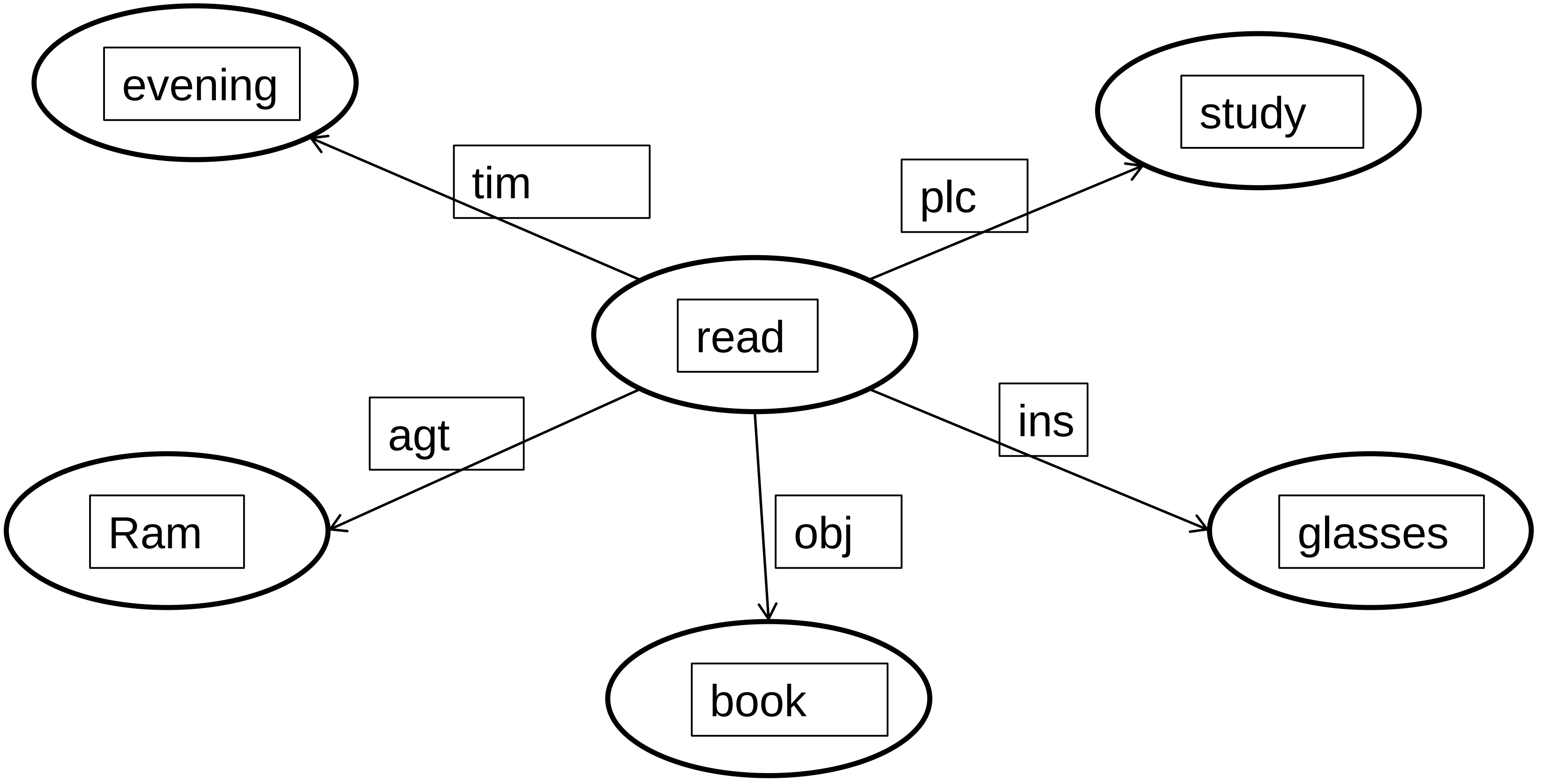
Stage
(Sentence)

Actors
1. Agent (Who)
2. Object (What)
3. Place (Where)
4. Time (When)
5. Instrument (by what)
6. Source (from where)
7. Destination (to where)

Action
(verb)

Missing here are actors that answer the questions how and why

- Ram reads a book with his glasses in the evening in his study.



- The labels on the arcs are semantic roles and the task is Semantic Role Labeling.

Two views of NLP

- Verb centric view
- Noun centric view
- Example
  - Information retrieval requires giving importance to Noun
    - What is the **Captial** of **India?**
  - Question answering requires more attention to verb
- Relationships of the verb with the noun
  - Agent (Who), Object (What), Place (Where), Time (When), Instrument (by What), Source (from Where), Destination (to Where)
  - These are called semantic roles or case roles or Karaka.

Case markers

Karak	**Function**	**Case markers**	**Preposition (English)**	**Post-position (Hindi)**
Karta (कर्ता)	Subject	Nominative		ne (ने)
Karma (कर्म)	Object	Accusative	To	ko (को)
Karan (करण)	Instrument	Instrumental	By/With/Through	se (से)
Sampradan (सम्प्रदान)	Receiver	Dative	To/For	ke liye (के लिए)
Apadan (अपादान)	Separation	Ablative	From	se (separate) (से अलग होने के लिए)
Sambandh (सम्बन्ध)	Possession	Genitive	Of	ka, ke, ki (का, के, की)
Adhikaran (अधिकरण)	Location	Locative	In/On/At/Among	me, par (में, पर)
Sambodhan (सम्बोधन)	Address someone	Vocative		He! (हे!)

Reference: https://openpathshala.com/blog/learn-sanskrit/introduction-to-karak-sanskrit-grammar [last accessed: 14-sept-2020]

Isolating the adjuncts from the arguments

- Example
  - *Ram reads a book with his glasses in the evening in his study.*
    - Argument: *Ram, book*
    - Adjuncts: *glasses, evening, study*
- The function words (determiners, prepositions) links between objects and arguments.
- Function words: They link the content words (the meaning bearing words)

Selectional preference

- Desire and Deservingness (आकांक्षा और योग्यता)
- Verbs has desire and nouns fulfils those desires
- Example
  - *Ram reads a book with his glasses in the evening in his study*.
  - *'evening'* can not have the place relationship and '*study'* can not have the time relationship
  - "*study*" gets selectional preference over "*evening*" when it comes to place relationship

Grammar and Parsing Algorithms

A simplified grammar

- $S \rightarrow NP\ VP$
- $NP \rightarrow DT\ N \mid N$
- $VP \rightarrow V\ ADV \mid V$

Example Sentence

People laugh

1 2 3

These are positions

Lexicon:

People - N, V

Laugh - N, V

This indicate that both Noun and Verb is possible for the word "People"

Top-Down Parsing

	State	Backup State	Action
1.	((S) 1)	-	-
2.	((NP VP)1)	-	-
3a.	((DT N VP)1	((N VP) 1)	-
3b.	((N VP)1)	-	-
4.	((VP)2)	-	Consume “People”
5a.	((V ADV)2)	((V)2)	-
6.	((ADV)3)	((V)2)	Consume “laugh”
5b.	((V)2)	-	-
6.	((.)3)	-	Consume “laugh”

Position of input pointer

Termination Condition : All inputs over. No symbols remaining.

Note: Input symbols can be pushed back.

Discussion for Top-Down Parsing

- This kind of searching is goal driven.
- Gives importance to textual precedence (rule precedence).
- No regard for data, a priori (useless expansions made).

Bottom-Up Parsing

Some conventions:

N_{12}

Represents positions

$S_{1?} \rightarrow NP_{12} \circ VP_{2?}$

End position unknown

Work on the LHS done, while the work on RHS remaining

Bottom-Up Parsing (pictorial representation)

$S \rightarrow NP_{12}\ VP_{23}\ \circ$

People Laugh

1 2 3

N_{12}

V_{12}

$NP_{12} \rightarrow N_{12}\ \circ$

$VP_{12} \rightarrow V_{12}\ \circ$

$S_{1?} \rightarrow NP_{12}\ \circ\ VP_{2?}$

N_{23}

V_{23}

$NP_{23} \rightarrow N_{23}\ \circ$

$VP_{23} \rightarrow V_{23}\ \circ$

Problem with Top-Down Parsing

- Left Recursion
  - Suppose you have A-> AB rule.

    Then we will have the expansion as follows:
    - ((A)K) -> ((AB)K) -> ((ABB)K) ……..

Combining top-down and bottom-up strategies

Top-Down Bottom-Up Chart Parsing

- Combines advantages of top-down & bottom-up parsing.
- Does not work in case of left recursion.
  - *e.g.* – "People laugh"
    - People – noun, verb
    - Laugh – noun, verb
  - Grammar – $S \rightarrow NP\ VP$

    $NP \rightarrow DT\ N \mid N$

    $VP \rightarrow V\ ADV \mid V$

Transitive Closure

People laugh

1 2 3

S →•NP VP
NP →•DT N
NP →•N

NP →N•
S → NP•VP
VP →•V ADV
VP →•V

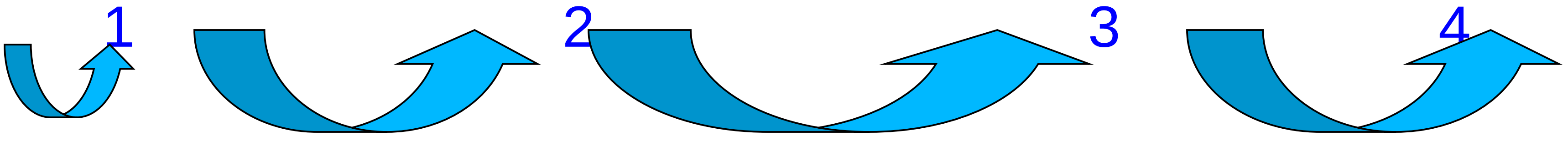
VP → V •
S → NP VP •
success

Arcs in Parsing

- Each arc represents a chart which records
  - Completed work (left of •)
  - Expected work (right of •)

Example

People laugh loudly



S $\rightarrow \bullet$ NP VP
NP $\rightarrow \bullet$ DT N
NP $\rightarrow \bullet$ N

NP $\rightarrow$ N$\bullet$
S $\rightarrow$ NP$\bullet$VP
VP $\rightarrow \bullet$V ADV
VP $\rightarrow \bullet$V

VP $\rightarrow$ V$\bullet$
VP $\rightarrow$ V$\bullet$ADV
S $\rightarrow$ NP VP$\bullet$

VP $\rightarrow$ V ADV$\bullet$
S $\rightarrow$ NP VP$\bullet$

An important parsing algo

Illustrating CYK [Cocke, Younger, Kashmi] Algo

- $S \rightarrow NP\ VP$ 1.0
- $NP \rightarrow DT\ NN$ 0.5
- $NP \rightarrow NNS$ 0.3
- $NP \rightarrow NP\ PP$ 0.2
- $PP \rightarrow P\ NP$ 1.0
- $VP \rightarrow VP\ PP$ 0.6
- $VP \rightarrow VBD\ NP$ 0.4

- $DT \rightarrow$ the 1.0
- $NN \rightarrow$ gunman 0.5
- $NN \rightarrow$ building 0.5
- $VBD \rightarrow$ sprayed 1.0
- $NNS \rightarrow$ bullets 1.0

CYK: Start with (0,1)

0 *The* 1 *gunman* 2 *sprayed* 3 *the* 4 *building* 5 *with* 6 *bullets* 7.

To From	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	DT						
1	-------						
2	-------	---------					
3	-------	---------	--------				
4	--------	---------	--------	---------			
5	--------	---------	--------	---------	---------		
6	--------	---------	--------	---------	---------	---------	

CYK: Keep filling diagonals

0 *The* 1 *gunman* 2 *sprayed* 3 *the* 4 *building* 5 *with* 6 *bullets* 7.

To **From**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	DT						
1	-------	NN					
2	-------	-------- -					
3	-------	-------- -	--------				
4	------- -	-------- -	--------	-------- -			
5	------- -	-------- -	--------	-------- -	-------- -		
6	------- -	-------- -	--------	-------- -	-------- -	-------- -	

CYK: Try getting higher level structures

0 *The* 1 *gunman* 2 *sprayed* 3 *the* 4 *building* 5 *with* 6 *bullets* 7*.*

To From	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	DT	NP					
1	-------	NN					
2	-------	---------					
3	-------	---------	--------				
4	--------	---------	--------	---------			
5	--------	---------	--------	---------	---------		
6	--------	---------	--------	---------	---------	---------	

CYK: Diagonal continues

0 *The* 1 *gunman* 2 *sprayed* 3 *the* 4 *building* 5 *with* 6 *bullets* 7.

To From	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	DT	NP					
1	-------	NN					
2	-------	---------	VBD				
3	-------	---------	--------				
4	--------	---------	--------	---------			
5	--------	---------	--------	---------	---------		
6	--------	---------	--------	---------	---------	---------	

CYK (cont…)

0 *The* 1 *gunman* 2 *sprayed* 3 *the* 4 *building* 5 *with* 6 *bullets* 7.

To From	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	DT	NP	--------				
1	-------	NN	--------				
2	-------	-------- -	VBD				
3	-------	-------- -	--------				
4	------- -	-------- -	--------	-------- -			
5	------- -	-------- -	--------	-------- -	-------- -		
6	------- -	-------- -	--------	-------- -	-------- -	-------- -	

CYK (cont…)

0 *The* **1** *gunman* **2** *sprayed* **3** *the* **4** *building* **5** *with* **6** *bullets* **7**.

To From	1	2	3	4	5	6	7
0	DT	NP	--------				
1	-------	NN	--------				
2	-------	---------	VBD				
3	-------	---------	--------	DT			
4	--------	---------	--------	---------			
5	--------	---------	--------	---------	---------		
6	--------	---------	--------	---------	---------	---------	

CYK (cont…)

0 *The* 1 *gunman* 2 *sprayed* 3 *the* 4 *building* 5 *with* 6 *bullets* 7.

To From	1	2	3	4	5	6	7
0	DT	NP	--------	---------			
1	-------	NN	--------	---------			
2	-------	---------	VBD	---------			
3	-------	---------	--------	DT			
4	--------	---------	--------	---------	NN		
5	--------	---------	--------	---------	---------		
6	--------	---------	--------	---------	---------	---------	

CYK: starts filling the 5th column

0 *The* 1 *gunman* 2 *sprayed* 3 *the* 4 *building* 5 *with* 6 *bullets* 7.

To From	1	2	3	4	5	6	7
0	DT	NP	--------	---------			
1	-------	NN	--------	---------			
2	-------	---------	VBD	---------			
3	-------	---------	--------	DT	NP		
4	--------	---------	--------	---------	NN		
5	--------	---------	--------	---------	---------		
6	--------	---------	--------	---------	---------	---------	

CYK (cont…)

0 *The* 1 *gunman* 2 *sprayed* 3 *the* 4 *building* 5 *with* 6 *bullets* 7.

To From	1	2	3	4	5	6	7
0	DT	NP	--------	---------			
1	-------	NN	--------	---------			
2	-------	---------	VBD	---------	VP		
3	-------	---------	--------	DT	NP		
4	--------	---------	--------	---------	NN		
5	--------	---------	--------	---------	---------		
6	--------	---------	--------	---------	---------	---------	

CYK (cont…)

0 *The* 1 *gunman* 2 *sprayed* 3 *the* 4 *building* 5 *with* 6 *bullets* 7*.*

To From	1	2	3	4	5	6	7
0	DT	NP	--------	-------- -			
1	-------	NN	--------	-------- -	-------- -		
2	-------	-------- -	VBD	-------- -	VP		
3	-------	-------- -	--------	DT	NP		
4	------- -	-------- -	--------	-------- -	NN		
5	------- -	-------- -	--------	-------- -	-------- -		
6	------- -	-------- -	--------	-------- -	-------- -	-------- -	

CYK: S found, but NO termination!

0 *The* **1** *gunman* **2** *sprayed* **3** *the* **4** *building* **5** *with* **6** *bullets* **7**.

To From	1	2	3	4	5	6	7
0	DT	NP	--------	---------	S		
1	-------	NN	--------	---------	---------		
2	-------	---------	VBD	---------	VP		
3	-------	---------	--------	DT	NP		
4	---------	---------	--------	---------	NN		
5	---------	---------	--------	---------	---------		
6	---------	---------	--------	---------	---------	---------	

CYK (cont…)

0 *The* 1 *gunman* 2 *sprayed* 3 *the* 4 *building* 5 *with* 6 *bullets* 7.

To From	1	2	3	4	5	6	7
0	DT	NP	--------	-------- -	S		
1	-------	NN	--------	-------- -	-------- -		
2	-------	-------- -	VBD	-------- -	VP		
3	-------	-------- -	--------	DT	NP		
4	------- -	-------- -	--------	-------- -	NN		
5	------- -	-------- -	--------	-------- -	-------- -	P	
6	------- -	-------- -	--------	-------- -	-------- -	-------- -	

CYK (cont…)

0 *The* 1 *gunman* 2 *sprayed* 3 *the* 4 *building* 5 *with* 6 *bullets* 7.

To From	1	2	3	4	5	6	7
0	DT	NP	--------	---------	S	---------	
1	-------	NN	--------	---------	---------	---------	
2	-------	---------	VBD	---------	VP	---------	
3	-------	---------	--------	DT	NP	---------	
4	--------	---------	--------	---------	NN	---------	
5	--------	---------	--------	---------	---------	P	
6	--------	---------	--------	---------	---------	---------	

CYK: Control moves to last column

0 *The* 1 *gunman* 2 *sprayed* 3 *the* 4 *building* 5 *with* 6 *bullets* 7.

To From	1	2	3	4	5	6	7
0	DT	NP	--------	---------	S	---------	
1	-------	NN	--------	---------	---------	---------	
2	-------	---------	VBD	---------	VP	---------	
3	-------	---------	--------	DT	NP	---------	
4	---------	---------	--------	---------	NN	---------	
5	---------	---------	--------	---------	---------	P	
6	---------	---------	--------	---------	---------	---------	NP NNS

CYK (cont…)

0 *The* 1 *gunman* 2 *sprayed* 3 *the* 4 *building* 5 *with* 6 *bullets* 7.

To From	1	2	3	4	5	6	7
0	DT	NP	--------	---------	S	---------	
1	-------	NN	--------	---------	---------	---------	
2	-------	---------	VBD	---------	VP	---------	
3	-------	---------	--------	DT	NP	---------	
4	--------	---------	--------	---------	NN	---------	
5	--------	---------	--------	---------	---------	P	PP
6	--------	---------	--------	---------	---------	---------	NP NNS

CYK (cont…)

0 *The* **1** *gunman* **2** *sprayed* **3** *the* **4** *building* **5** *with* **6** *bullets* **7**.

To From	1	2	3	4	5	6	7
0	DT	NP	--------	---------	S	---------	
1	-------	NN	--------	---------	---------	---------	
2	-------	---------	VBD	---------	VP	---------	
3	-------	---------	--------	DT	NP	---------	NP
4	--------	---------	--------	---------	NN	---------	---------
5	--------	---------	--------	---------	---------	P	PP
6	--------	---------	--------	---------	---------	---------	NP NNS

CYK (cont…)

0 *The* 1 *gunman* 2 *sprayed* 3 *the* 4 *building* 5 *with* 6 *bullets* 7.

To From	1	2	3	4	5	6	7
0	DT	NP	--------	---------	S	---------	
1	-------	NN	--------	---------	---------	---------	
2	-------	---------	VBD	---------	VP	---------	VP
3	-------	---------	--------	DT	NP	---------	NP
4	---------	---------	--------	---------	NN	---------	---------
5	---------	---------	--------	---------	---------	P	PP
6	---------	---------	--------	---------	---------	---------	NP NNS

CYK: filling the last column

0 *The* 1 *gunman* 2 *sprayed* 3 *the* 4 *building* 5 *with* 6 *bullets* 7.

To From	1	2	3	4	5	6	7
0	DT	NP	--------	-------- -	S	-------- -	
1	-------	NN	--------	-------- -	-------- -	-------- -	-------- -
2	-------	-------- -	VBD	-------- -	VP	-------- -	VP
3	-------	-------- -	--------	DT	NP	-------- -	NP
4	------- -	-------- -	--------	-------- -	NN	-------- -	-------- -
5	------- -	-------- -	--------	-------- -	-------- -	P	PP
6	------- -	-------- -	--------	-------- -	-------- -	-------- -	NP NNS

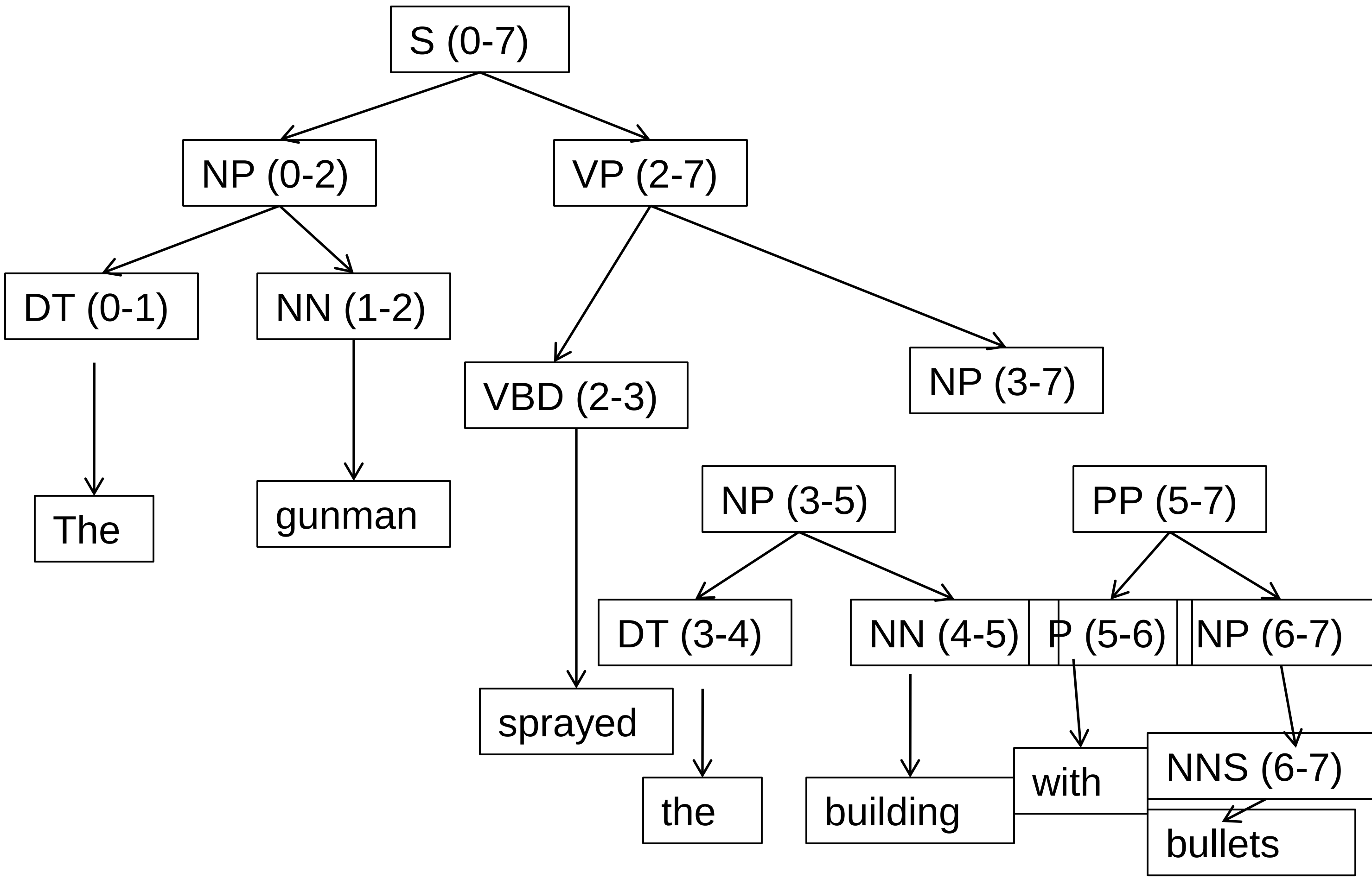
CYK: terminates with S in (0,7)

0 *The* 1 *gunman* 2 *sprayed* 3 *the* 4 *building* 5 *with* 6 *bullets* 7.

To From	1	2	3	4	5	6	7
0	DT	NP	--------	---------	S	---------	S
1	-------	NN	--------	---------	---------	---------	---------
2	-------	---------	VBD	---------	VP	---------	VP
3	-------	---------	--------	DT	NP	---------	NP
4	---------	---------	--------	---------	NN	---------	---------
5	---------	---------	--------	---------	---------	P	PP
6	---------	---------	--------	---------	---------	---------	NP NNS

CYK: Extracting the Parse Tree

- The parse tree is obtained by keeping back pointers.



Probabilistic parsing

Example of Sentence labeling: Parsing

$[_{S1}[_{S}[_{S}[_{VP}[_{VB}\text{Come}][_{NP}[_{NNP}\text{July}]]]]$

$[_{,}\ ,]$

$[_{CC}\ \text{and}]$

$[_{S}\ [_{NP}\ [_{DT}\ \text{the}]\ [_{JJ}\ \text{IIT}]\ [_{NN}\ \text{campus}]]$

$[_{VP}\ [_{AUX}\ \text{is}]$

$[_{ADJP}\ [_{JJ}\ \text{abuzz}]$

$[_{PP}[_{IN}\ \text{with}]$

$[_{NP}[_{ADJP}\ [_{JJ}\ \text{new}]\ [_{CC}\ \text{and}]\ [_{VBG}\ \text{returning}]]$

$[_{NNS}\ \text{students}]]]]]$

$[_{.}\ .]]]$

Noisy Channel Modeling

Source sentence

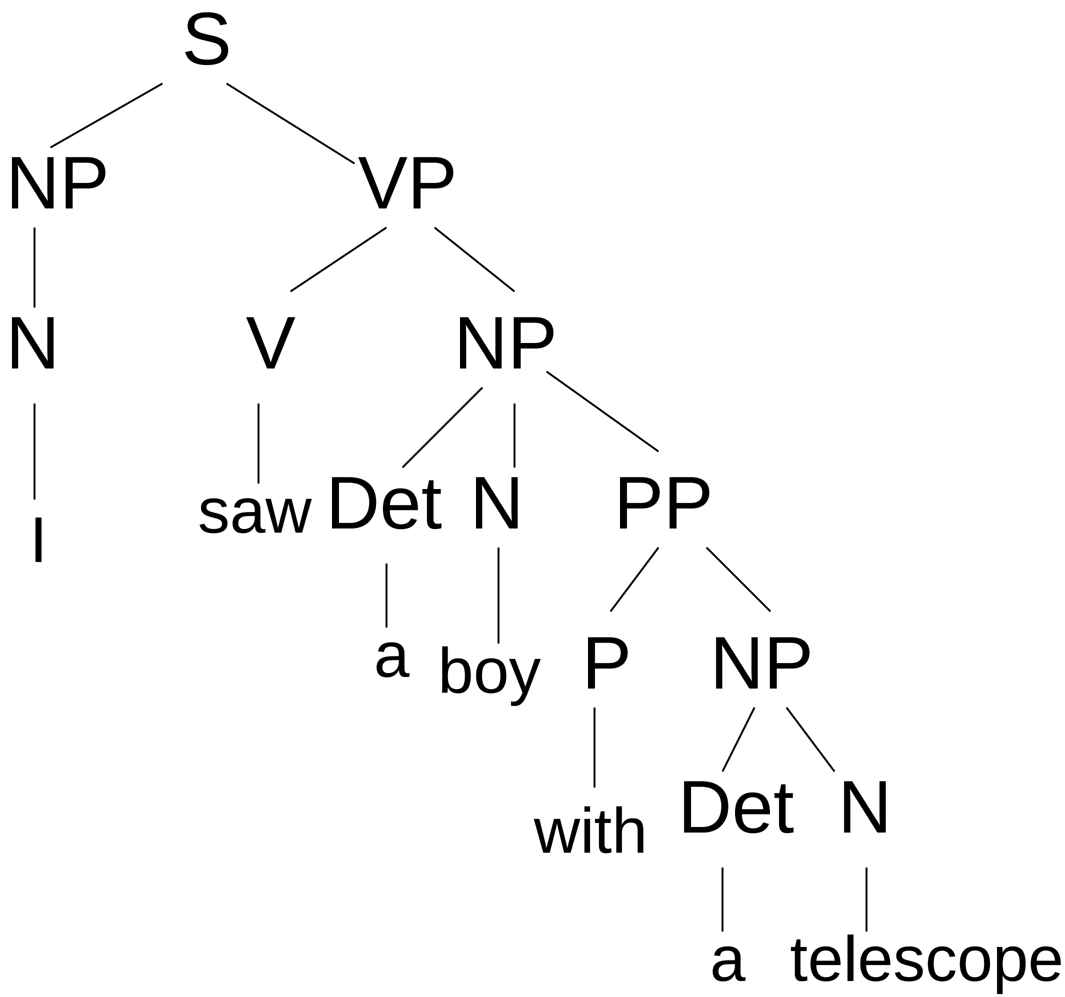
Target parse

$$T^* = \arg\max_{T} [P(T|S)]$$

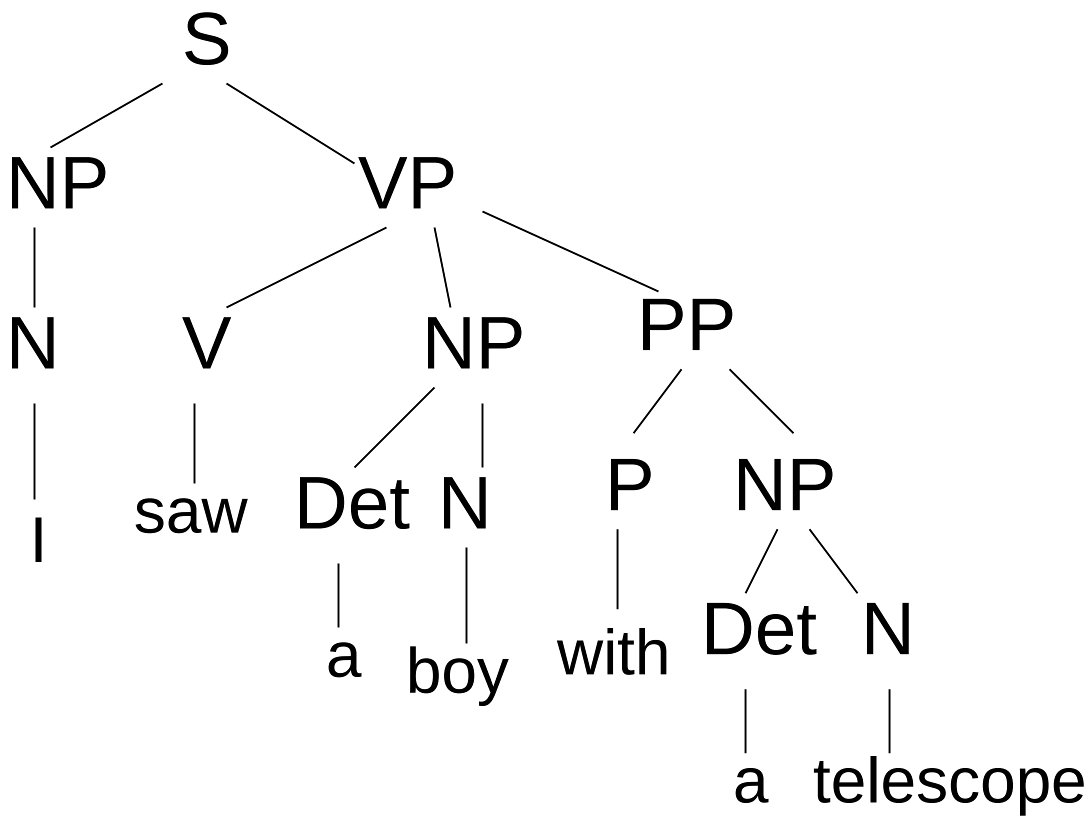
$$= \arg\max_{T} [P(T).P(S|T)]$$

$$= \arg\max_{T} [P(T)]$$, since given the parse the sentence is completely determined and $P(S|T)=1$

I saw a boy with a telescope: Tree - 1



Constituency Parse Tree -2



Formal Definition of PCFG

- A PCFG consists of
  - A set of terminals $\{w_k\}$, k = 1,….,V

    $\{w_k\}$ = { child, teddy, bear, played…}
  - A set of non-terminals $\{N^i\}$, i = 1,…,n

    $\{N_i\}$ = { NP, VP, DT…}
  - A designated start symbol N^1
  - A set of rules $\{N^i \rightarrow \zeta^j\}$, where ζ^j is a sequence of terminals & non-terminals

    NP $\rightarrow$ DT NN
  - A corresponding set of rule probabilities

Rule Probabilities

- Rule probabilities are such that

$$\forall i \ \sum_{\mathrm{i}} \mathrm{P}(\mathrm{N}^{i} \rightarrow \zeta^{j}) = 1$$

E.g., P(NP $\rightarrow$ DT NN) = 0.2

P(NP $\rightarrow$ NN) = 0.5

P(NP $\rightarrow$ NP PP) = 0.3

- P(NP $\rightarrow$ DT NN) = 0.2
  - Means 20 % of the training data parses use the rule NP $\rightarrow$ DT NN

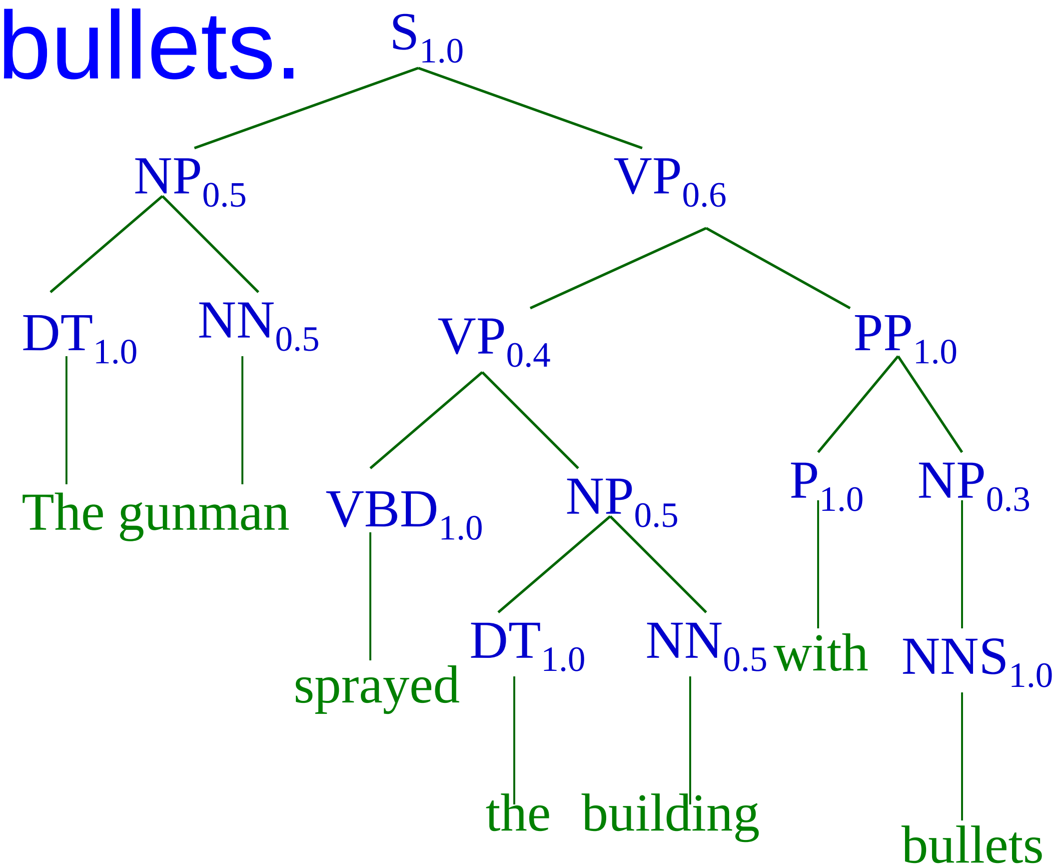
Probabilistic Context Free Grammars

- S → NP VP 1.0
- NP → DT NN 0.5
- NP → NNS 0.3
- NP → NP PP 0.2
- PP → P NP 1.0
- VP → VP PP 0.6
- VP → VBD NP 0.4

- DT → the 1.0
- NN → gunman 0.5
- NN → building 0.5
- VBD → sprayed 1.0
- NNS → bullets 1.0

Example Parse t_1`

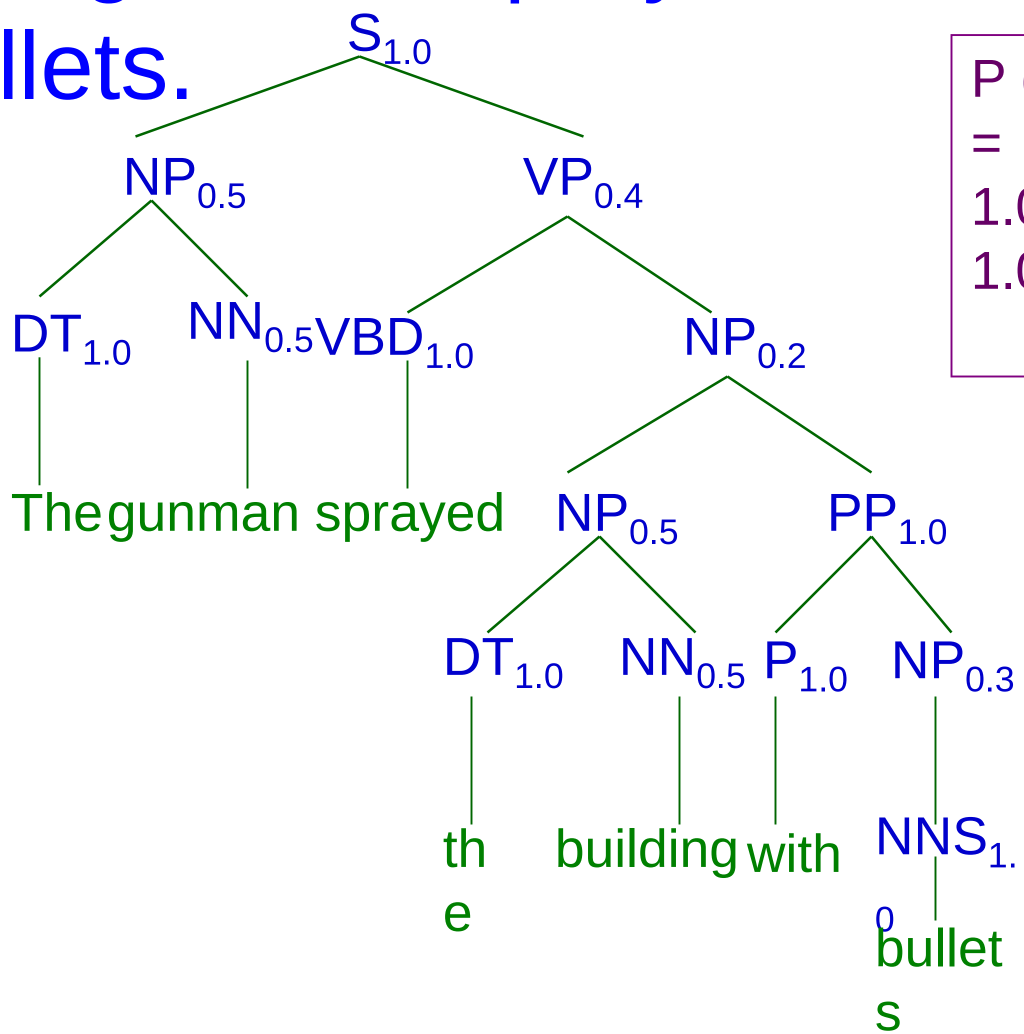
- The gunman sprayed the building with bullets.



$P(t_1) = 1.0 * 0.5 * 1.0 * 0.5 * 0.6 * 0.4 * 1.0 * 0.5 * 1.0 * 0.5 * 1.0 * 1.0 * 0.3 * 1.0 = 0.00225$

Another Parse t_2

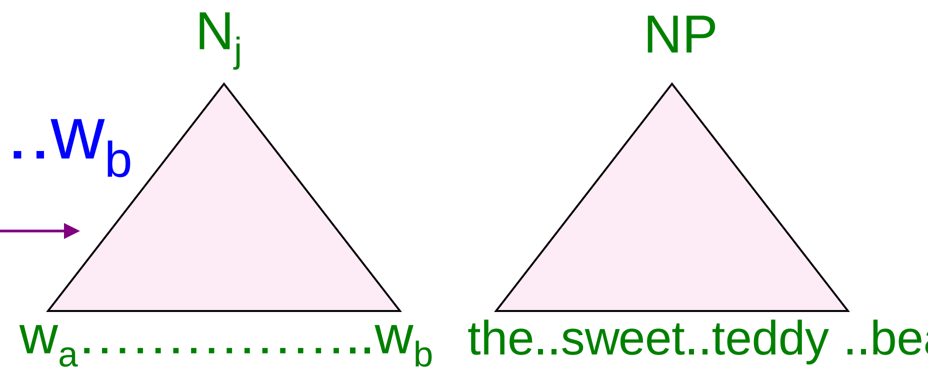
- The gunman sprayed the building with bullets.



P (t_2)
= 1.0 * 0.5 * 1.0 * 0.5 * 0.4 * 1.0 * 0.2 * 0.5 * 1.0 * 0.5 * 1.0 * 1.0 * 0.3 * 1.0
= 0.0015

Probability of a sentence

- Notation :
  - w_{ab} – subsequence $w_a....w_b$
  - N_j dominates $w_a....w_b$ or yield(N_j) = $w_a....w_b$



- Probability of a sentence = $P(w_{1m})$

$$P(w_{1m}) = \sum_t P(w_{1m}, t)$$

Where t is a parse tree of the sentence

$$= \sum_t P(t) P(w_{1m} \mid t)$$

$$= \sum_{t:yield(t)=w_{1m}} P(t) \qquad \because P(w_{1m} \mid t) = 1$$

If t is a parse tree for the sentence w_{1m}, this will be 1 !!

Assumptions of the PCFG model

- Place invariance :

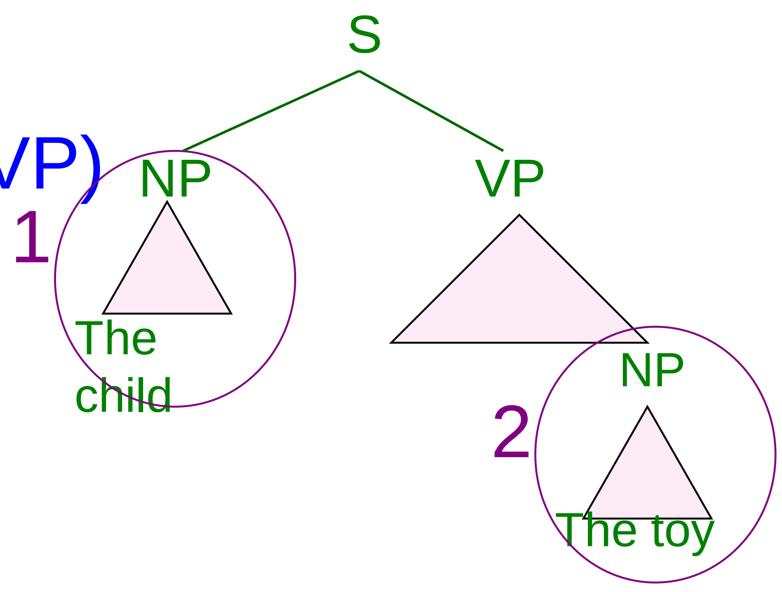
  P(NP → DT NN) is same in locations 1 and 2

- Context-free :

  P(NP → DT NN | anything outside "The child")
  = P(NP → DT NN)

- Ancestor free : At 2,

  P(NP → DT NN|its ancestor is VP)
  = P(NP →DT NN)

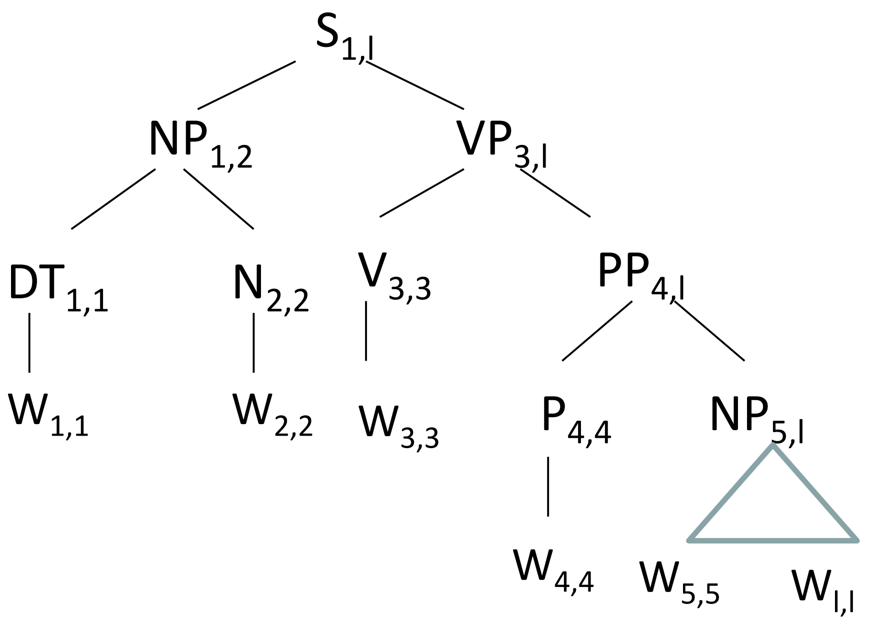


Probability of a parse tree

- *Domination* :We say N_j dominates from k to l, symbolized as $N^{j}_{k,l}$, if $W_{k,l}$ is derived from N_j
- P (tree |sentence) = P (tree | $S_{1,l}$)

  where $S_{1,l}$ means that the start symbol S dominates the word sequence $W_{1,l}$
- P (t |s) approximately equals joint probability of constituent non-terminals dominating the sentence fragments (next slide)

Probability of a parse tree (cont.)



$P(t|s) = P(t \mid S_{1,l})$

$= P(NP_{1,2}, DT_{1,1}, w_{1,1},$
$N_{2,2}, w_{2,2}$
$VP_{3,l}, V_{3,3}, w_{3,3}$
$PP_{4,l}, P_{4,4}, w_{4,4}\ NP_{5,l}, w_{5\ldots l} \mid S_{1,l})$

$= P(NP_{1,2}, VP_{3,l} \mid S_{1,l}) * P(DT_{1,1}, N_{2,2} \mid NP_{1,2}) *$
$P(w_{1,1} \mid DT_{1,1}) * P(w_{2,2} \mid N_{2,2}) * P(V_{3,3}, PP_{4,l} \mid VP_{3,l}) *$
$P(w_{3,3} \mid V_{3,3}) * P(P_{4,4}, NP_{5,l} \mid PP_{4,l}) * P(w_{4,4} \mid P_{4,4}) *$
$P(w_{5\ldots l} \mid NP_{5,l})$

(Using Chain Rule, Context Freeness and Ancestor Freeness)

Why probability in Parsing

Why probability in parsing?

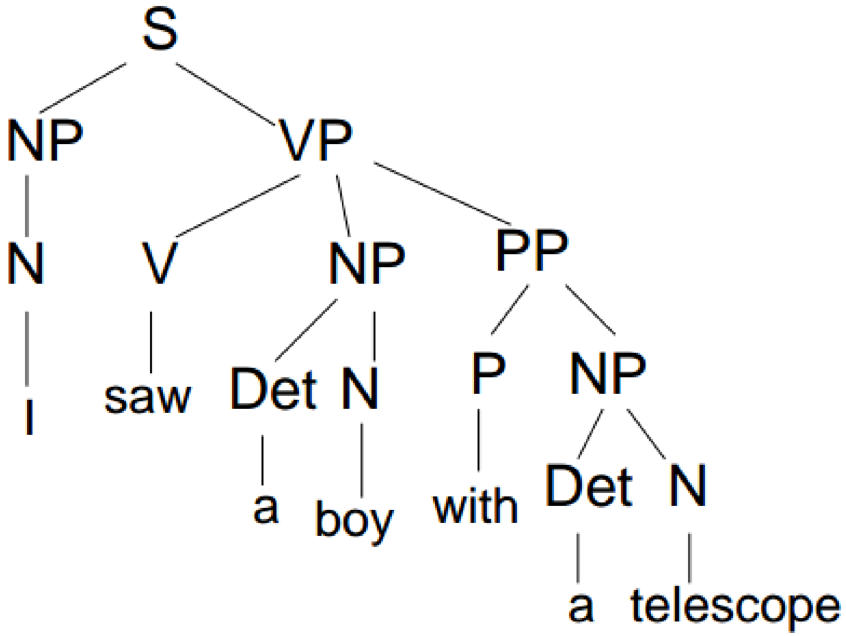
- What is randomness in tree?
  - At every position of the sentence there is a potential ambiguity with respect to whatever phrase structure can be built till and from that point
  - This leads to ambiguity in the parse tree
  - The root of a subtree covering a segment of the sentence is said to **dominate** that segment
  - The ambiguity in deciding **domination** leads to randomness
- Example:

In the earthquake **old** men and women were taken to safe locations.

Domination

- A sentence is dominated by the symbol S through domination of segments by phrases
- Examples
  - The capital of a country dominates the whole country.
  - The capital of a state dominates the whole state.
  - The district headquarter dominates the district.
  - IIT Bombay is dominated by the administration of IIT Bombay.
  - Administration dominates Heads of Depts
  - The department is dominated by head of the department.

Domination: Example



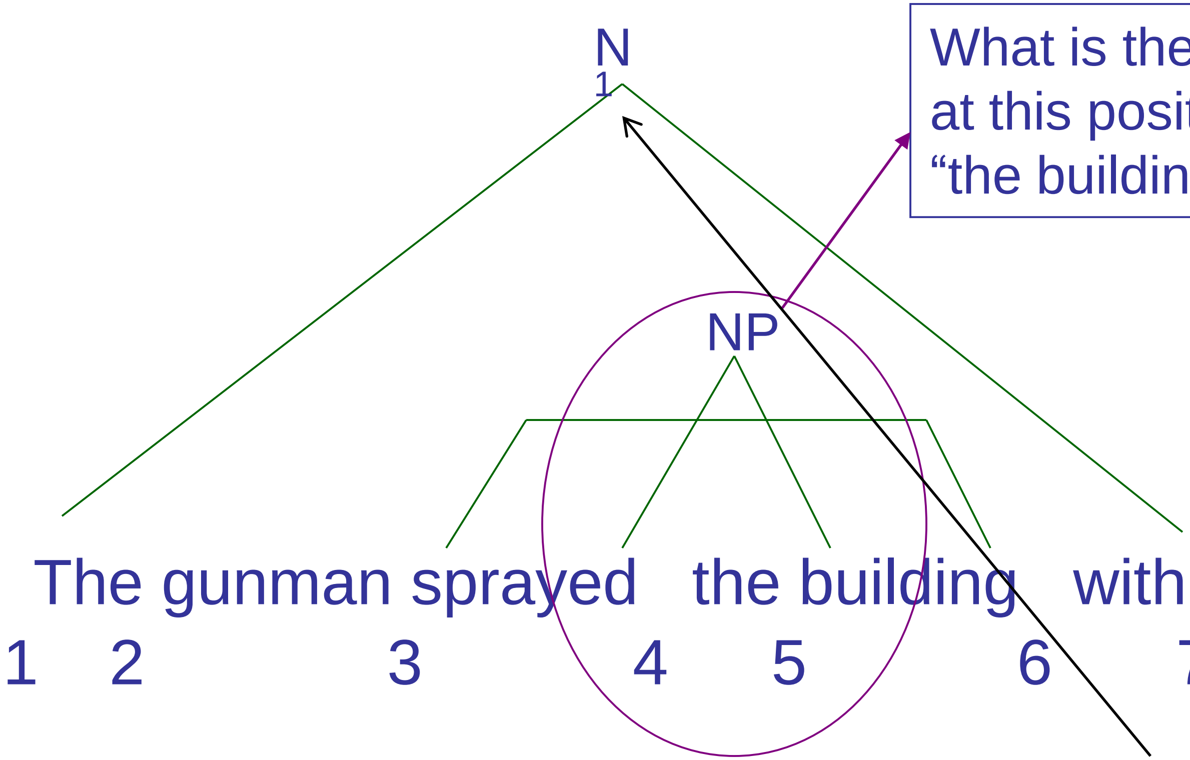
I saw a boy with a telescope

Meaning: I used the telescope to see the boy

- Dominations
  - NP dominates “a telescope”
  - VP dominates “saw a boy with a telescope
  - S dominates the whole sentence

- Domination is composed of many sub-domination.

Interesting Probabilities



What is the probability of having a NP at this position such that it will derive "the building" ? - $\beta_{NP}(4,5)$

Inside Probabilities

Outside Probabilities

What is the probability of starting from N^1 and deriving "The gunman sprayed", a NP and "with bullets" ? - $\alpha_{NP}(4,5)$

Parse tree for the given sentence using probabilistic CYK parsing

$_0$ **The** $_1$ **gunman** $_2$ **sprayed** $_3$ **the** $_4$ **building** $_5$ **with** $_6$ **bullets** $_7$

- Two parse trees are possible because the sentence has attachment ambiguity .

- Total 16 multiplications are required to make both the parse trees using probabilistic CYK.

- Number of multiplications is less in comparison to a probabilistic parsing which prepares the two parse trees independently with 28 multiplication.

	The 1	gunman 2	Sprayed 3	the 4	Building 5	with 6	Bullets 7
0	β_{DT} (0-1) =1.0	β_{NP} (0-2) =0.25					β_{S}(0-7) =0.006
1		β_{NN} (1-2) =0.5					
2			β_{VBD}(2-3) =1.0		β_{VP} (2-5) =0.1		β_{VP}(2-7) =0.024
3				β_{DT}(3-4) =1.0	β_{NP} *(3-5) =0.25*		β_{NP}(3-7) =0.015
4					β_{NN} (4-5) =0.5		
5						β_{P}(5-6) =1.0	β_{PP}(5-7) =0.3
6							$\beta_{NP/NNS}$(6-7) =1.0

Calculation of values for each non terminal occuring in the CYK table

β_{DT} (0-1) =1.0   (From Grammar rules)

β_{NN} (1-2) =0.5   (From Grammar rules)

β_{NP} (0-2) = P(the gunman | $NP_{0\text{-}2}$, G)
= P(NP->DT NN)* β_{DT} (0-1) * β_{NN} (1-2)
= 0.5 * 1.0 * 0.5
=0.25

β_{VBD}(2-3) =1.0   (From Grammar rules)

β_{DT}(3-4) =1.0   (From Grammar rules)

β_{NN} (4-5)   =0.5   (From Grammar rules)

β_{NP} (3-5) = P(the building | $NP_{3\text{-}5}$, G)
= P(NP->DT NN)* β_{DT} (3-4) * β_{NN} (4-5)
= 0.5 * 1.0 * 0.5
=0.25

$$\beta_{VP}(2\text{-}5) = P(VP\text{->}VBD\ NP) * \beta_{VBD}(2\text{-}3) * \beta_{NN}(3\text{-}5)$$
$$= 0.4 * 1 * 0.25$$
$$= 0.1$$

$$\beta_{P}(5\text{-}6) = 1.0 \text{ (From Grammar rules)}$$

$$\beta_{NP/NNS}(6\text{-}7) = 1.0 \text{ (From Grammar rules)}$$

$$\beta_{PP}(5\text{-}7) = P(PP\text{->}P\ NP) * \beta_{P}(5\text{-}6) * \beta_{NP/NNS}(6\text{-}7)$$
$$= 1.0 * 1.0 * 0.3$$
$$= 0.3$$

$$\beta_{NP}(3\text{-}7) = P(NP\text{->}NP\ PP) * \beta_{NP}(3\text{-}5) * \beta_{PP}(5\text{-}7)$$
$$= 0.2 * 0.25 * 0.3$$
$$= 0.015$$

$$\beta_{VP}(2\text{-}7) = (P(VP\text{->}VBD\ NP) * \beta_{VBD}(2\text{-}3) * \beta_{NP}(3\text{-}7) + P(VP\text{->}VP\ PP) * \beta_{VP}(2\text{-}5) * \beta_{PP}(5\text{-}7))$$
$$= 0.4 * 1 * 0.015 + 0.6 * 0.1 * 0.3$$
$$= 0.024$$

$$\boldsymbol{\beta_{S}(0\text{-}7) = P(S\text{->}NP\ VP) * \beta_{NP}(0\text{-}2) * \beta_{VP}(2\text{-}7)}$$
$$\boldsymbol{= 1 * 0.25 * 0.024}$$
$$\boldsymbol{= 0.006}$$

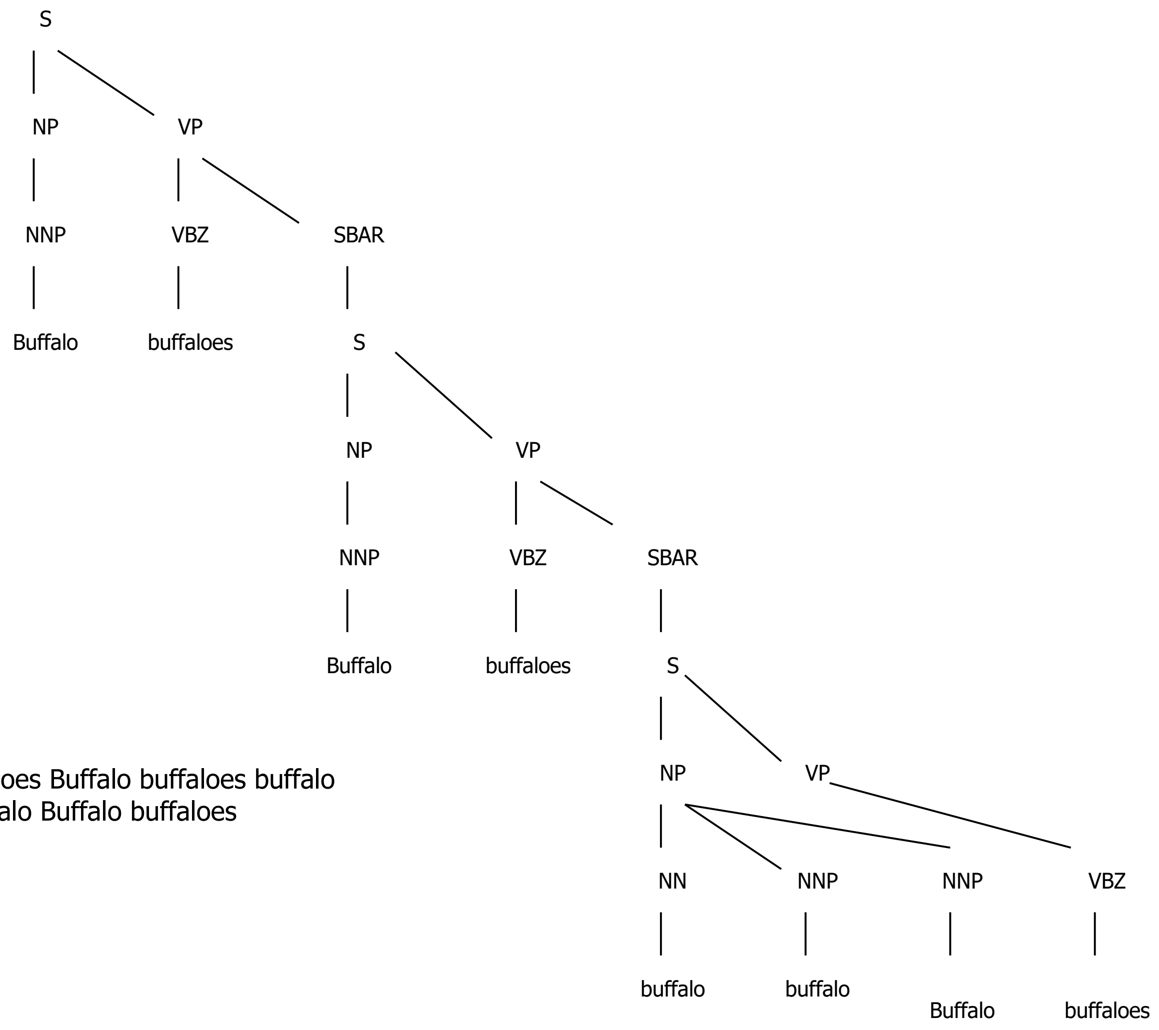
A very difficult parsing situation!

Repeated Word handling

Sentence on Buffaloes!

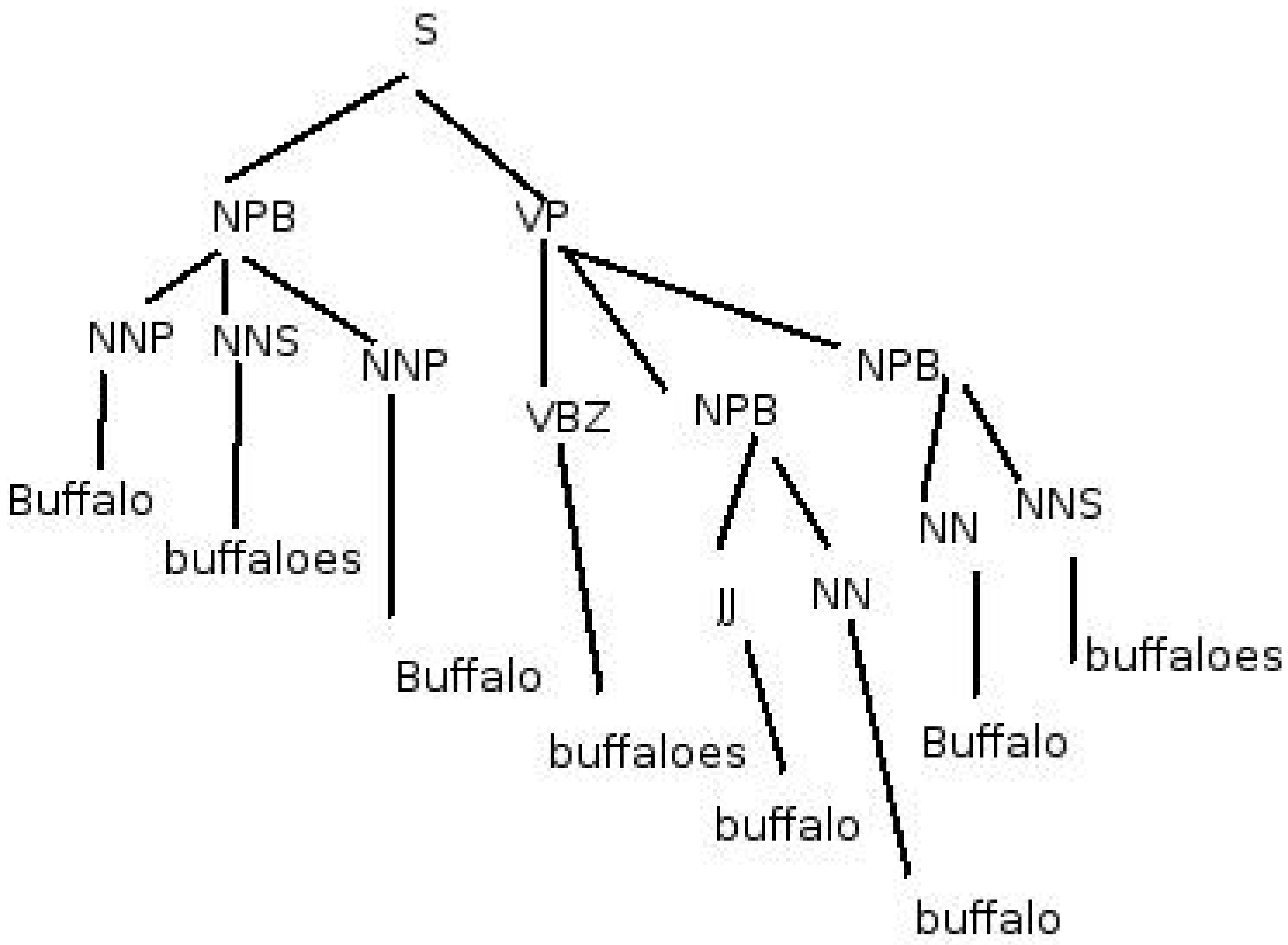
Buffaloe buffaloes Buffaloe buffaloes buffaloe buffaloe Buffaloe buffaloes

Charniak

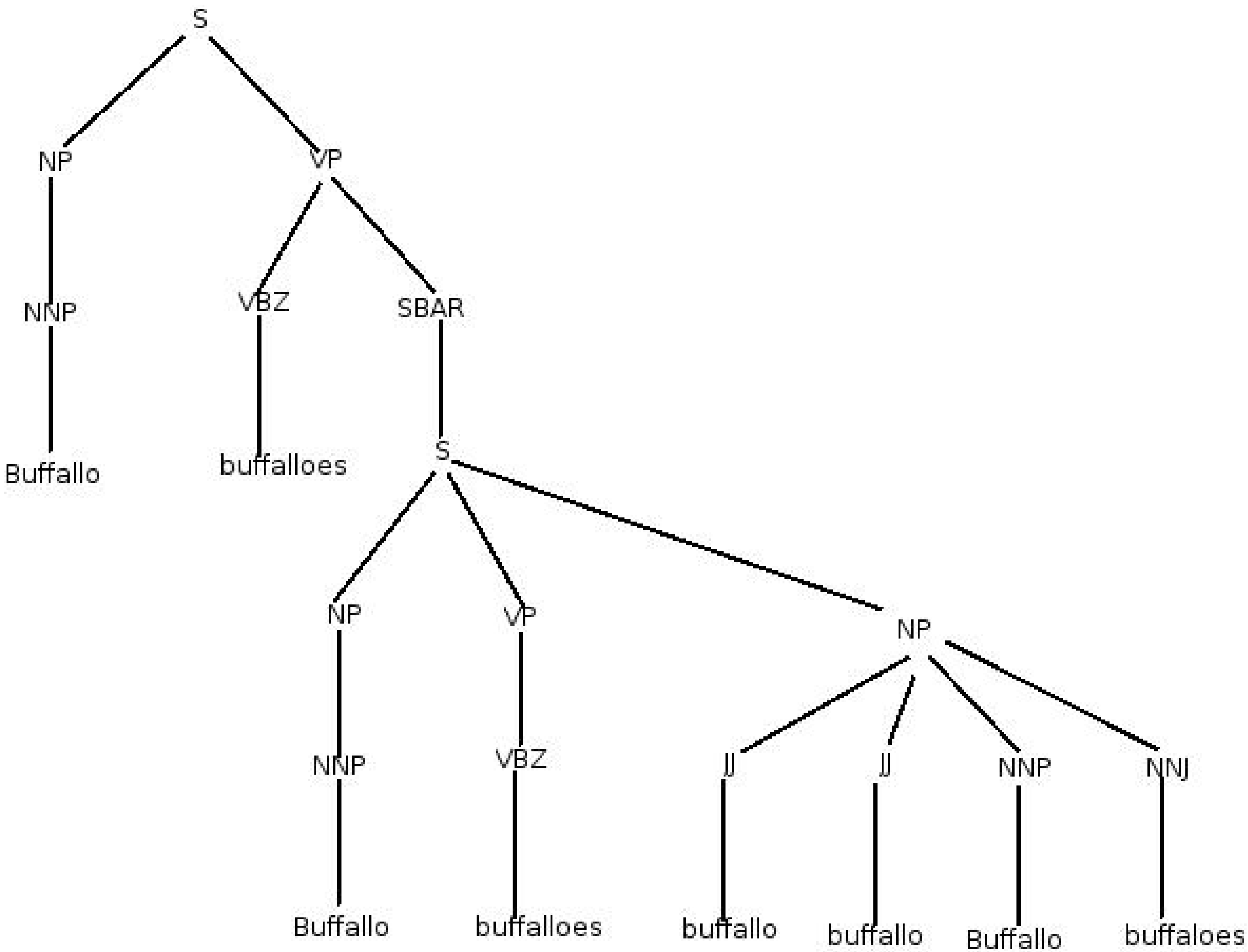


Buffalo buffaloes Buffalo buffaloes buffalo buffalo Buffalo buffaloes

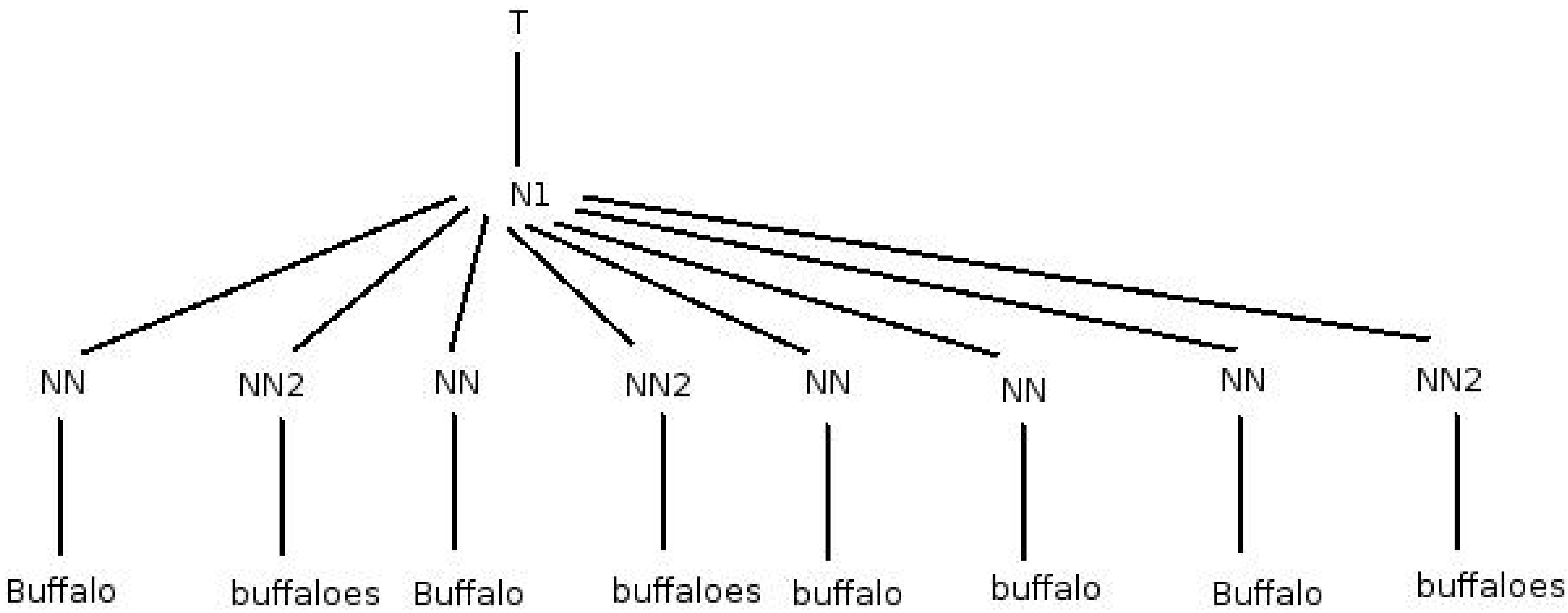
Collins


S
NPB
VP
NNP
NNS
NNP
VBZ
NPB
NPB
Buffalo
buffaloes
Buffalo
buffaloes
JJ
NN
NN
NNS
buffalo
buffalo
Buffalo
buffaloes

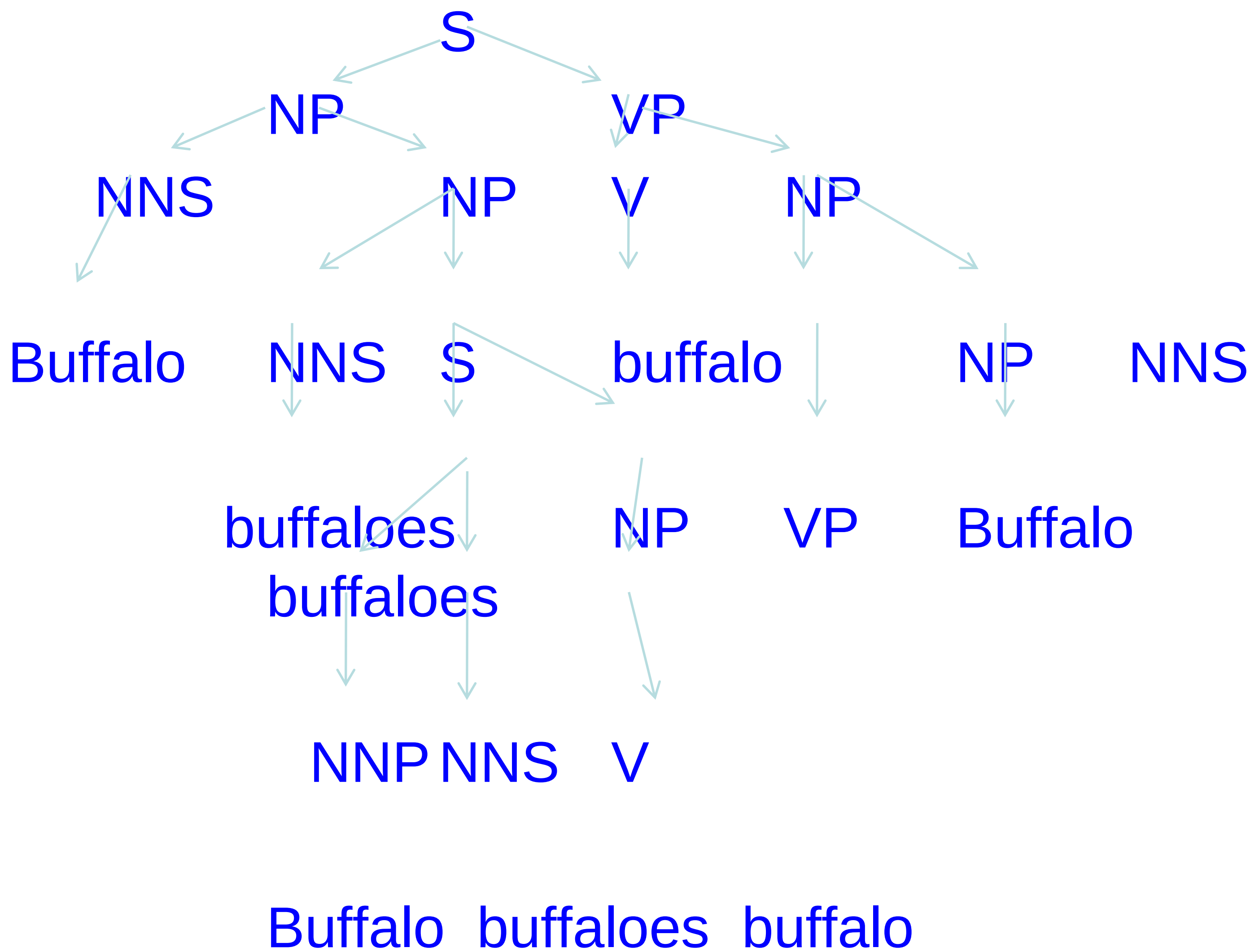
Stanford


S
NP
VP
NNP
VBZ
SBAR
Buffallo
buffalloes
S
NP
VP
NP
NNP
VBZ
JJ
JJ
NNP
NNJ
Buffallo
buffalloes
buffallo
buffallo
Buffallo
buffaloes

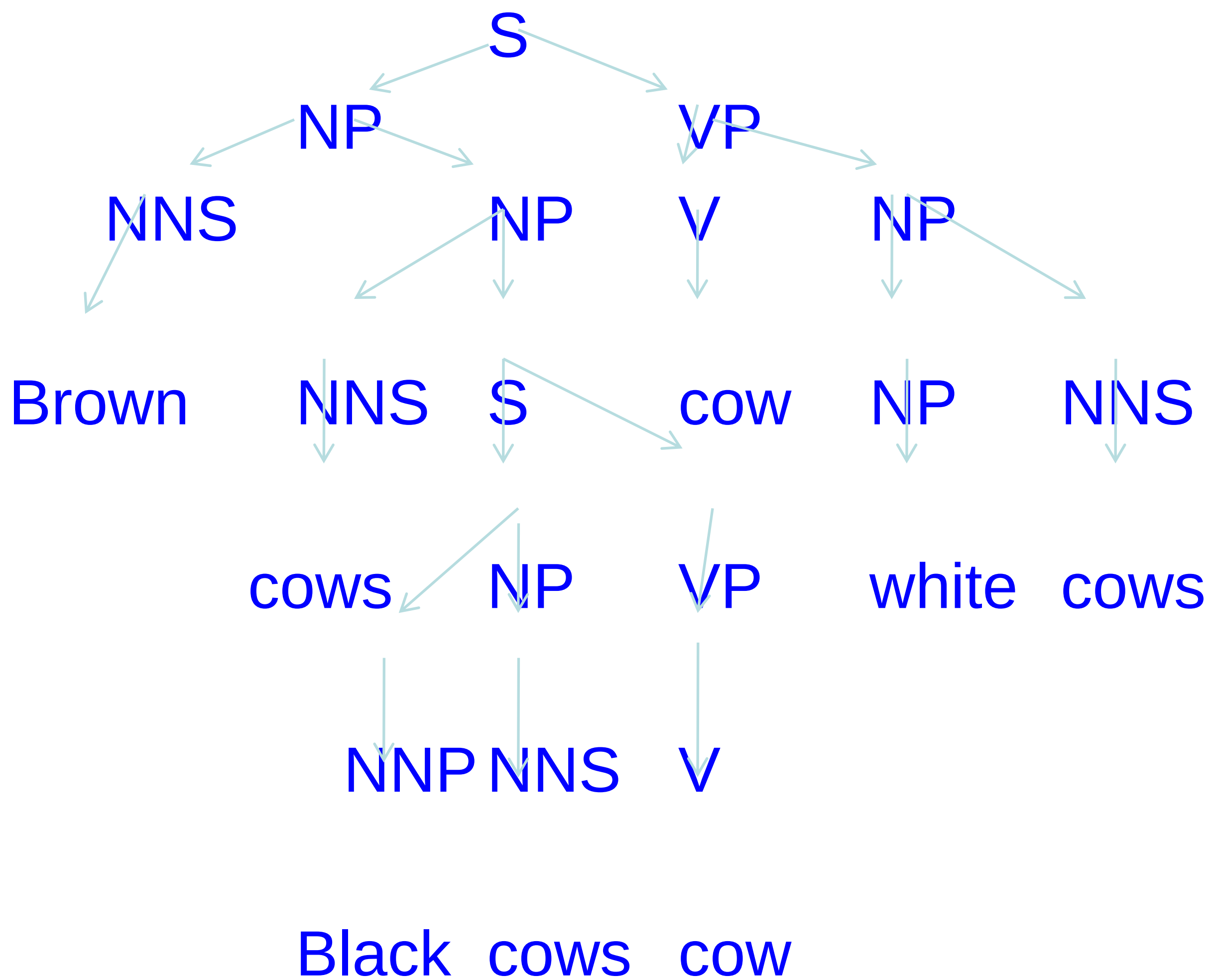
RASP


T
N1
NN
NN2
NN
NN2
NN
NN
NN
NN2
Buffalo
buffaloes
Buffalo
buffaloes
buffalo
buffalo
Buffalo
buffaloes

Correct parse



Another sentence of same structure

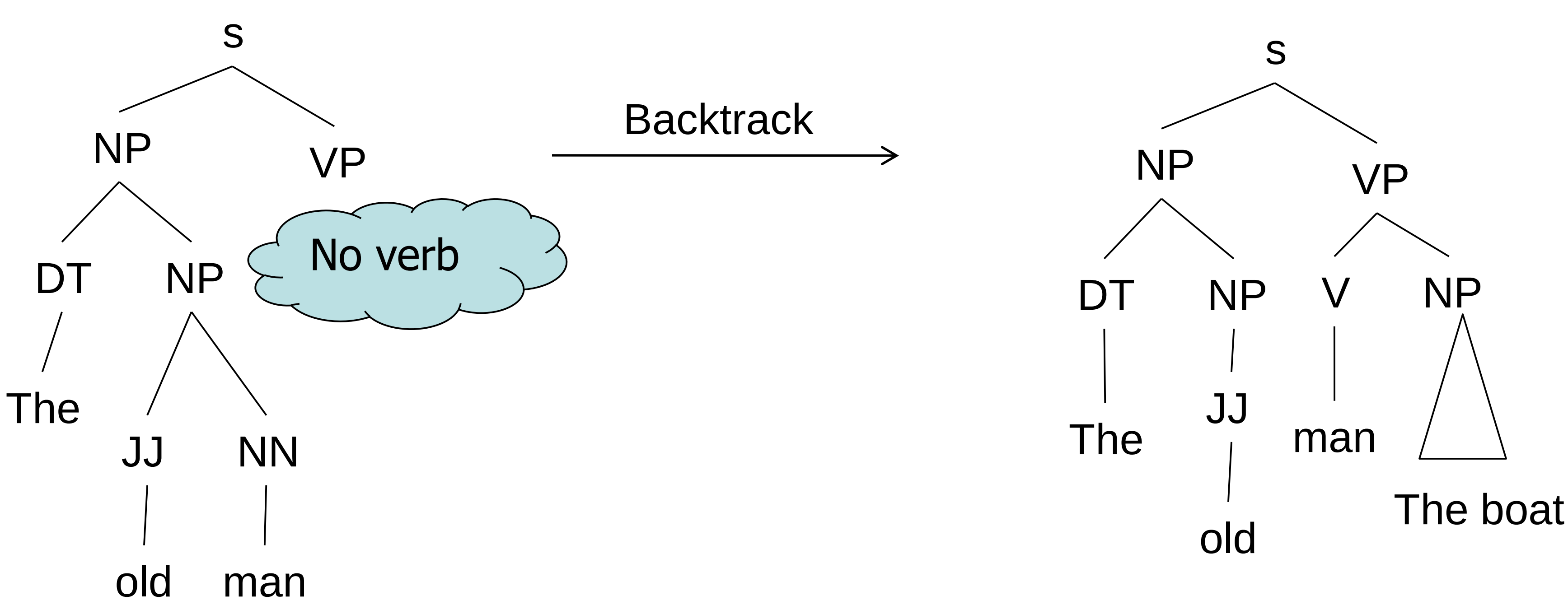


Observation

- Collins and Charniak come close to producing the correct parse.
- RASP tags all the words as nouns.

Another phenomenon: Garden pathing

e.g. The old man the boat.



Another example: The horse raced past the garden fell.

Project Suggestions

Principle

- How do I choose a project?
  - Beauty: e.g. Music
  - Utility: e.g. Chatbot
  - Both: e.g., chatbot that understands and produces humour

Poetry

Translation, question answer

Well known poem

- Jack and Jill
- went up the hill
- to fetch a pail of water
- Jack fell down
- And broke his crown
- And Jill came tumbling after

Google Hindi

- जैक और जिल
- पहाड़ी के ऊपर चला गया
- पानी का एक पाउच लाने के लिए |
- जैक नीचे गिर गया
- और उसका मुकुट तोड़ दिया
- और जिल बाद में लड़खड़ाते हुए आया ||

Bing Hindi

- जैक और जिल
- पहाड़ी पर चला गया
- पानी की एक पैल लाने के लिए |
- जैक नीचे गिर गया
- और उसका मुकुट तोड़ दिया
- और जिल के बाद tumbling आया ||

With Rhythm

- जैक और जिल
- गया चोटी पर चल
- लाने को बाल्टी भर पानी |
- जैक गया गिर
- उसका फट गया सिर
- और जिल आई लुढ़कती हुई पीछे ||

Question Answering on Poetry

- Who went with Jack? *Jill*
- Who went with Jill? *Jack*
- Who got injured? *Jack*
- Why did they go up the hill?- *to fetch water*
- What did they carry? *a pail*
- Did they get water?- *do not know*
- Did Jill slide down?- *No*
- Did Jill too get injured?- *likely*

Tweets

Translation, Formalization, Normalization

- *“got 2 go” → got to go*
- *“hey what’s up” → Hello how are you?*
- *“hey! what’s up” → are! Kyaa chal rahaa?*

Humour

Uses Text Ambiguity

“Officer: There is heavy firing

Minister: which sector?

Officer: IT”

“I filed a lawsuit against the airport due to my luggage being lost, but I lost the case”

“haldi is healthy”

Social n/w related

Topics

- Automatic population of LinkedIn profile from the home page, e.g., publications
  - Want a module that when run does the automatic populating
- Translating FB posts into multiple languages
- Linking Instagram with FB
  - Automatically captioning Instagram picture with FB post
- Aggregating search results from multiple social n/w
  - FB, LinkedIn, Instagram

Information Retrieval and Information Extraction Related

Topics

- Summarization
- Combined summarization and sentiment (multitask learning)
  - Oh no!, forgot to send the letter!
  - Information content: did not send the letter
  - Pragmatic content: disappointment indicated by “oh, no!”, that summarization will drop
- Tune search: retrieving song name, lyrics, composer, movie name etc. from the tune (needs feature extraction from the sound)

ASR-NLU-TTS

Wholly or partically

Topics

- Spoken signal conversion to text

Explainability

Topics

- Interpretability of word embeddings
  - Which component represents what
  - ‘dog’, ‘cat’, ‘tiger’ vectors: which component(s) represent ‘carnivorous’?
- Extracting rules from trained neural nets
- Shadowing a neural net with decision tree or support vector machine
  - Decision tree uncovers rules, SVM uncovers features