Parsing, Contd

Prof. Sameer Singh

CS 295: STATISTICAL NLP

WINTER 2017

February 9, 2017

Based on slides from Dan Jurafsky, Noah Smith, Slav Petrov, and everyone else they copied from.

Outline

Syntactic Parsing: CKY Algorithm

Probabilistic CFGs

Dependency Parsing

Outline

Syntactic Parsing: CKY Algorithm

Probabilistic CFGs

Dependency Parsing

Review of Syntactic Parsing

Context-free Grammar

G

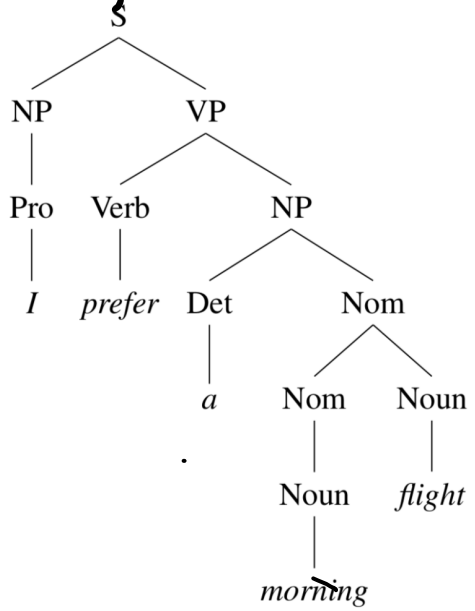
T - {man, book, cry ...}

N - {S, NP, VP, Noun, Verbs, PP...}

R - {S → NP VP, Noun → man, VP → Verb PP..}

Parse Tree

X, G → Tree



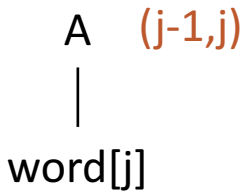
Dynamic Programming

T[i,j] = Set of all valid non-terminals for the constituent span (i,j)

Base case

Rule: $A \rightarrow word[j]$

A should be in T[j-1,j]

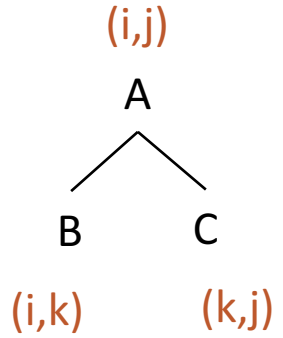


Recursion

Rule: $A \rightarrow B\ C$

T[0,n]

S



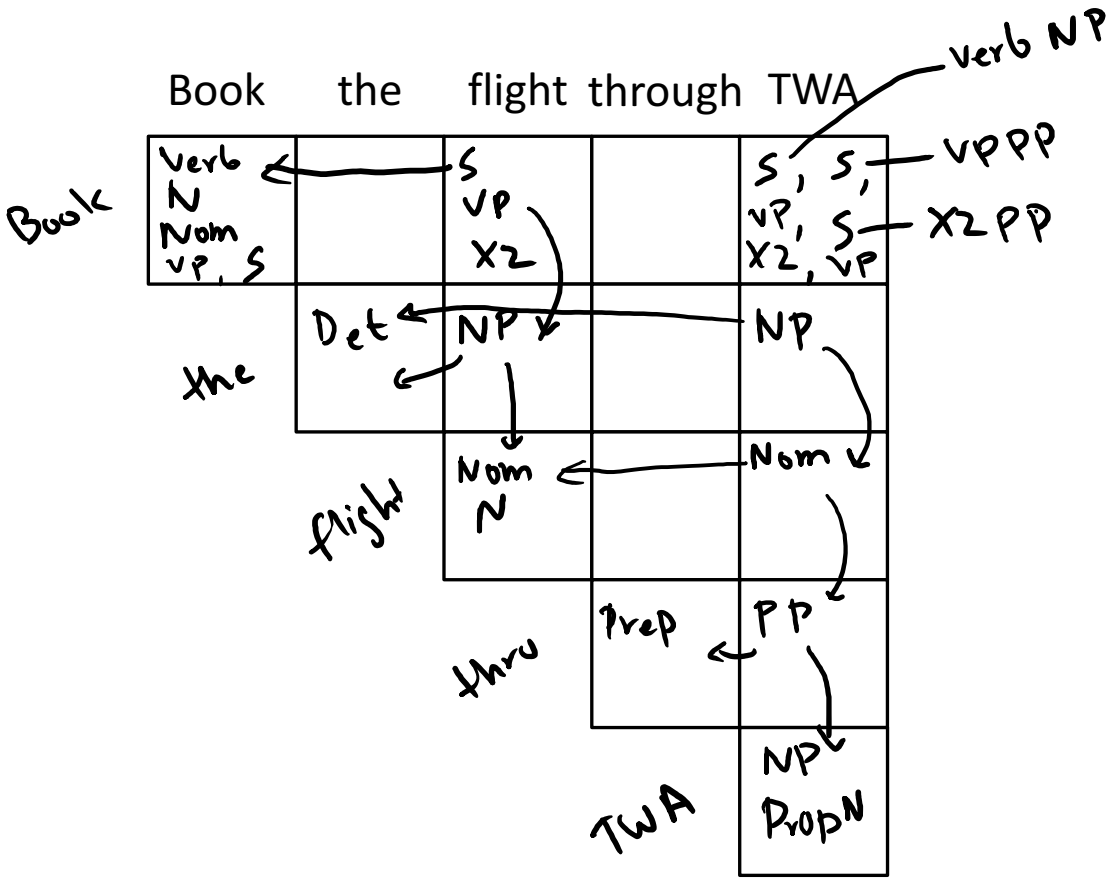
If you find a k such that

B is in T[i,k], and

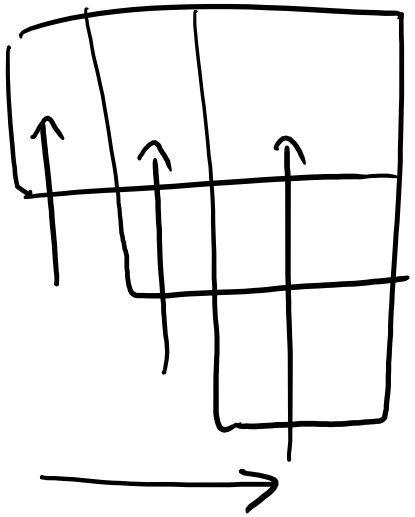
C is in T[k,j], then A should be in T[i,j]

CKY Algorithm

$S \rightarrow NP\ VP$
$S \rightarrow X1\ VP$
$X1 \rightarrow Aux\ NP$
$S \rightarrow book \mid include \mid prefer$
$S \rightarrow Verb\ NP$
$S \rightarrow X2\ PP$
$S \rightarrow Verb\ PP$
$S \rightarrow VP\ PP$
$NP \rightarrow I \mid she \mid me$
$NP \rightarrow TWA \mid Houston$
$NP \rightarrow Det\ Nominal$
$Nominal \rightarrow book \mid flight \mid meal \mid money$
$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow book \mid include \mid prefer$
$VP \rightarrow Verb\ NP$
$VP \rightarrow X2\ PP$
$X2 \rightarrow Verb\ NP$
$VP \rightarrow Verb\ PP$
$VP \rightarrow VP\ PP$
$PP \rightarrow Preposition\ NP$



CKY Algorithm



function CKY-PARSE(*words, grammar*) **returns** *table*

for $j \leftarrow$ **from** 1 **to** LENGTH(*words*) **do**
  for all $\{A \mid A \rightarrow words[j] \in grammar\}$
    $table[j-1, j] \leftarrow table[j-1, j] \cup A$
  for $i \leftarrow$ **from** $j-2$ **downto** 0 **do**
    for $k \leftarrow i+1$ **to** $j-1$ **do**
      for all $\{A \mid A \rightarrow BC \in grammar$ **and** $B \in table[i,k]$ **and** $C \in table[k,j]\}$
        $table[i,j] \leftarrow table[i,j] \cup A$

n

n

n

|R|

CKY Algorithm: Complexity

|N|: Number of non-terminals

|R|: Number of rules

n: Number of tokens in the sentence

Memory

$O(n^2 |N|)$

Time

$O(|R| n^3)$

Outline

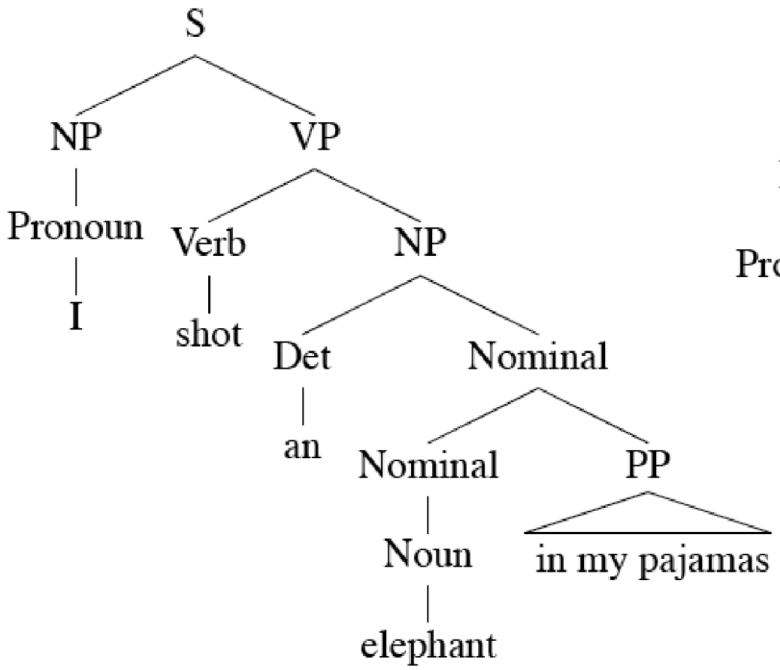
Syntactic Parsing: CKY Algorithm

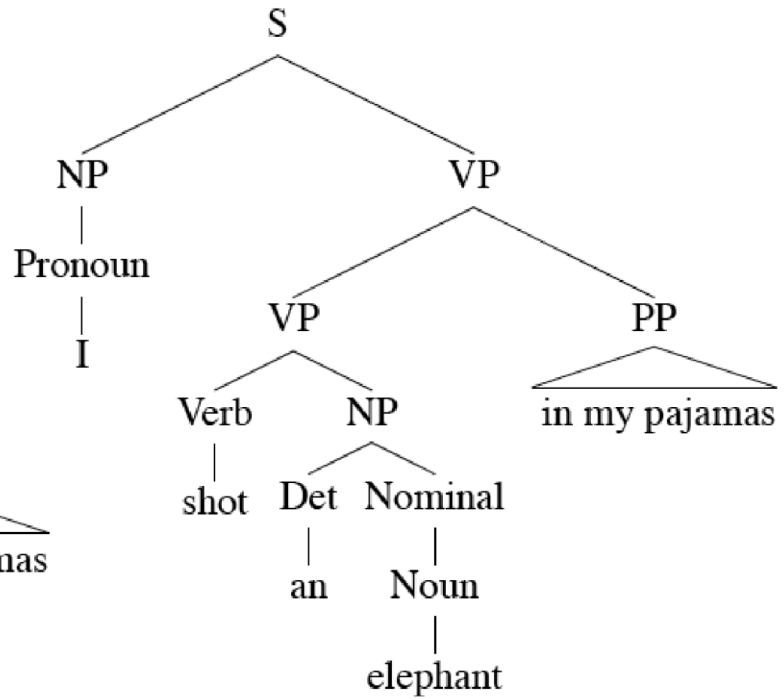
Probabilistic CFGs

Dependency Parsing

Ambiguity: Which parse?

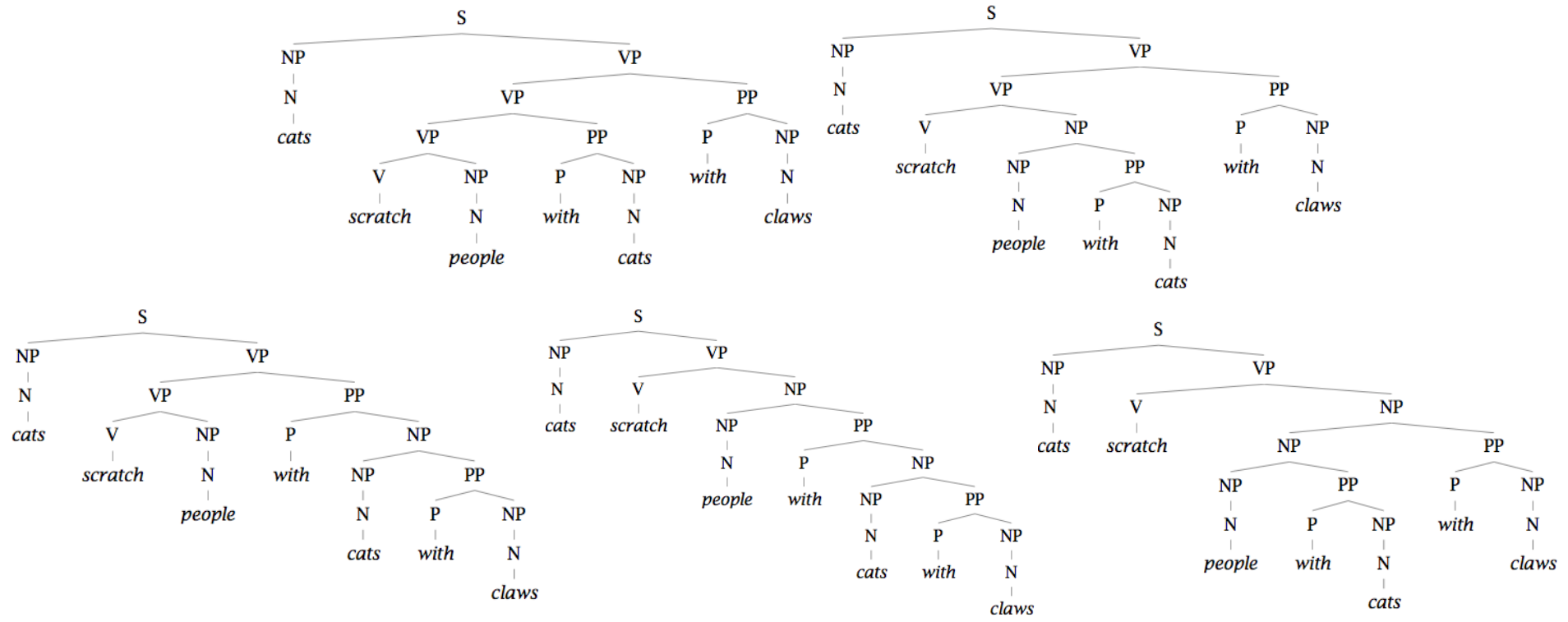
I shot an elephant in my pajamas.





Finding the *Best* Parse Tree

Cats scratch people with cats with claws.



Probabilistic CFGs

Same as a regular context-free grammar:

- Terminal, non-terminals, and rules
- Additionally, attach a probability to each rule!

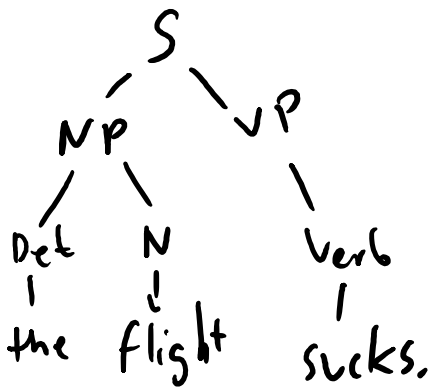
Rule: $A \rightarrow B\ C$

Probability: P($A \rightarrow B\ C \mid A$)

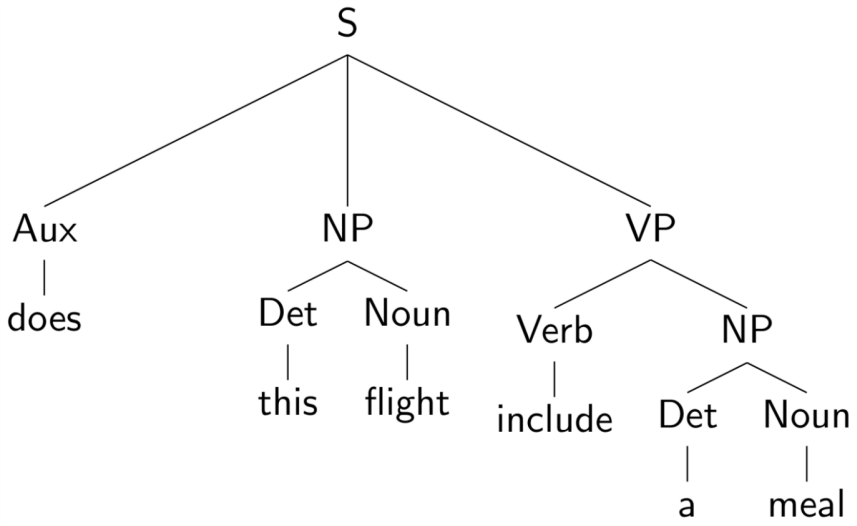
Compute the probability of a parse tree:

$$P(t) = P(\text{the} \mid \text{Det})\, P(\text{flight} \mid \text{N}) \ldots$$

$$\ldots P(\text{NP VP} \mid \text{S})$$



Example of a PCFG



$p(\text{Aux NP VP} \mid S)$

$p(\text{does} \mid \text{Aux})$

$p(\text{Det N} \mid \text{NP})$

$p(\text{Verb NP} \mid \text{VP})$

$\vdots$

Estimating the probabilities

$$p(\alpha \to \beta \mid \alpha) = \frac{\#\, \alpha \to \beta \text{ appears}}{\#\, \alpha \text{ appears}}$$

The Parsing Problem

Given sentence **x** and grammar **G**,

Recognition

Is sentence **x** in the grammar? If so, prove it.

"Proof" is a deduction, valid parse tree.

Parsing

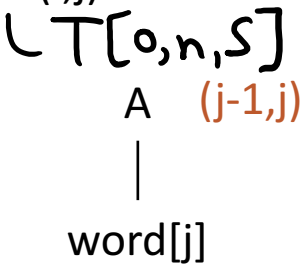
Show one or more derivations for **x** in **G**.

$$\underset{t \in \mathcal{T}_x}{\operatorname{argmax}}\, p(t \mid x)$$

Even with small grammars, grows exponentially!

Probabilistic CKY Algorithm

T[i,j,A] = Probability of the best parse with root A for the span (i,j)



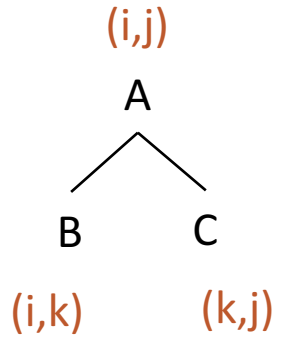
Base case

Rule: P($A \rightarrow word[j]$)

T[j-1,j,A] = P(word[j] | A)

Recursion

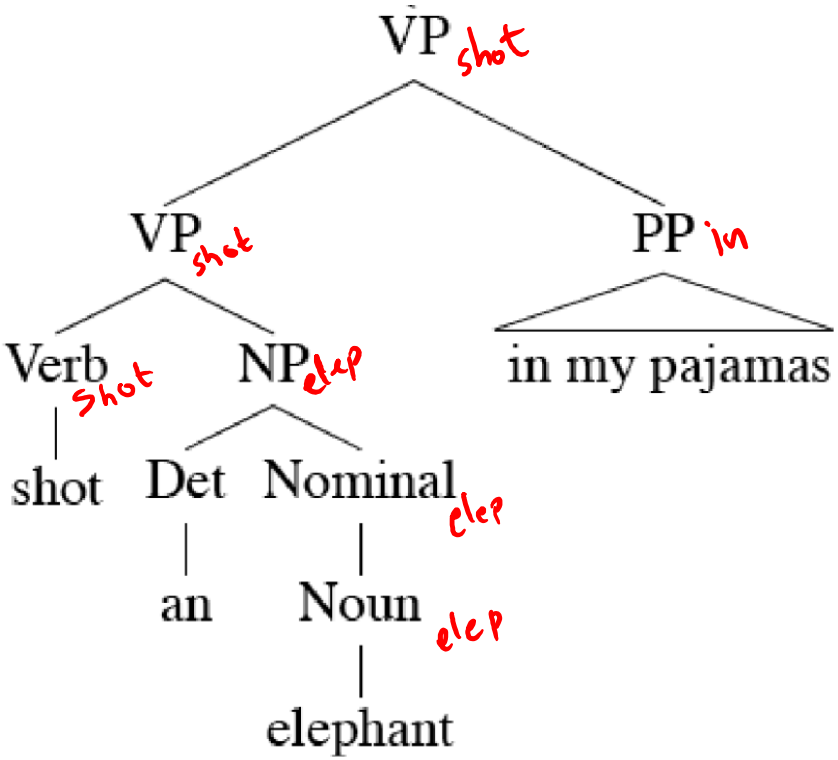
Rule: P($A \rightarrow B\ C$)



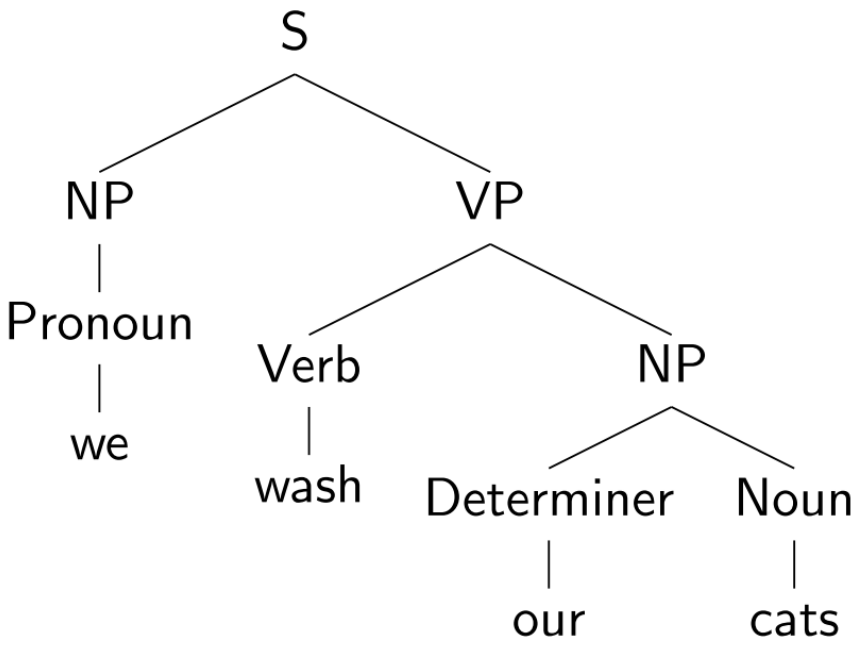
Try every position k, and every non-terminal pair:

$$T[i,j,A] = \max_{k} P(B\ C \mid A)\ T[i,k,B]\ T[k,j,C]$$

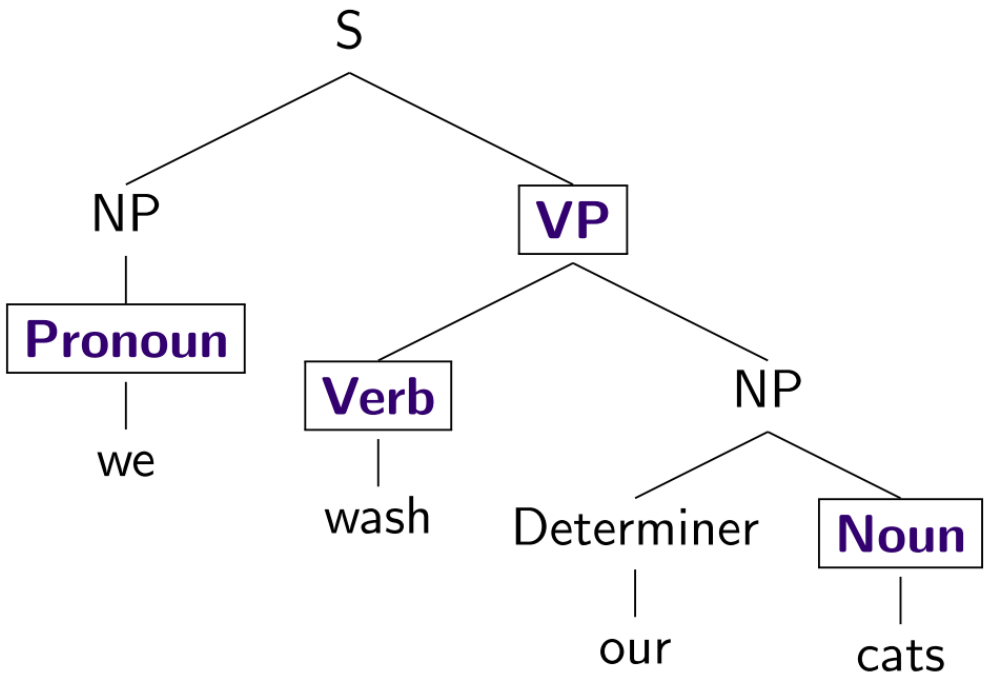
Lexicalized PCFGs



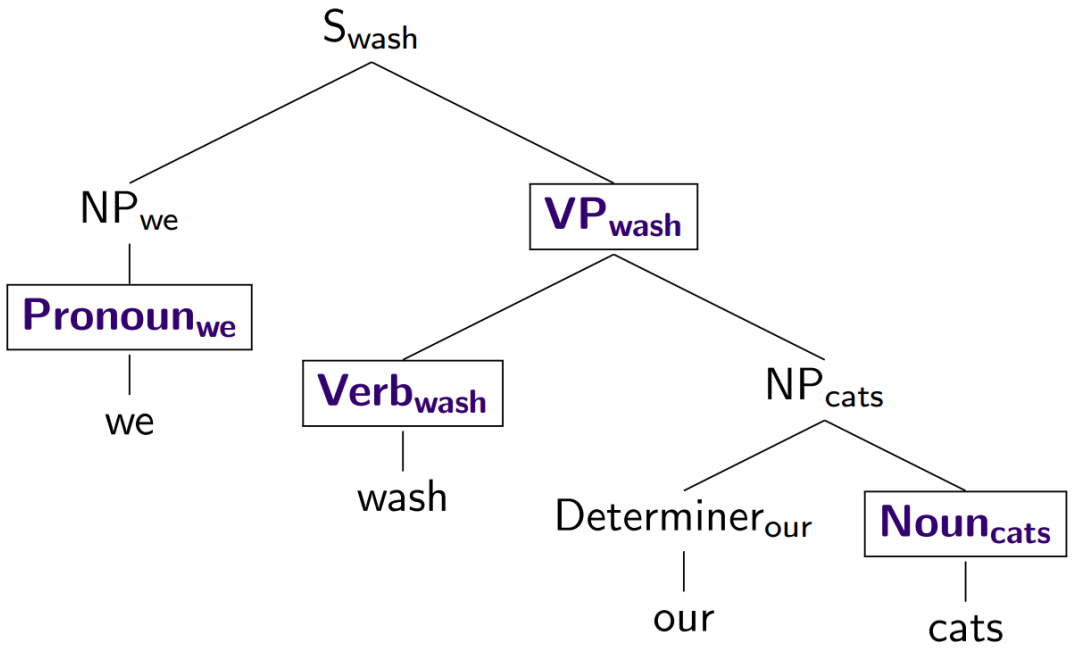
Lexicalizing a CFG



Lexicalizing a CFG



Lexicalizing a CFG



Outline

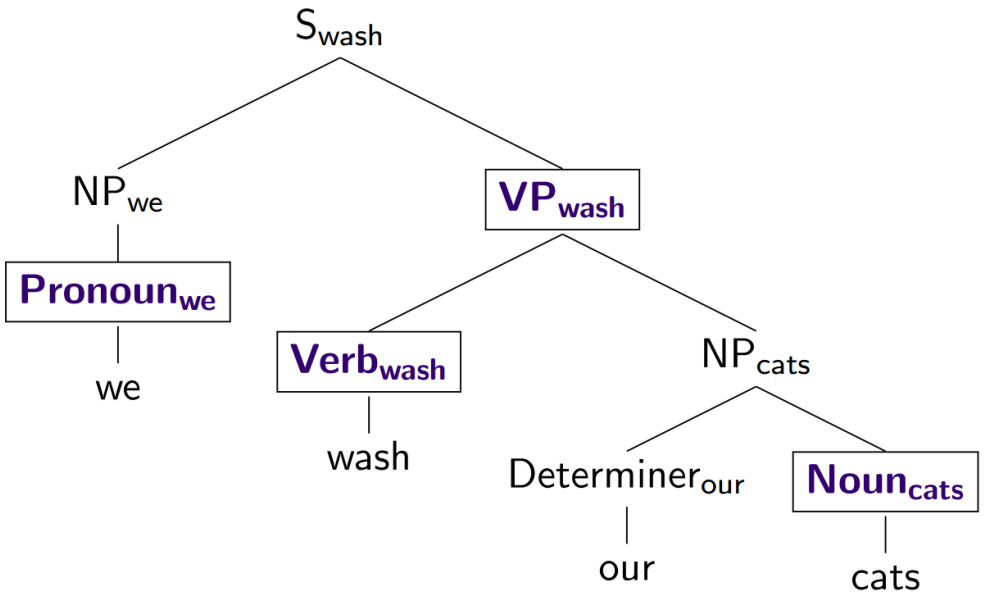
Syntactic Parsing: CKY Algorithm

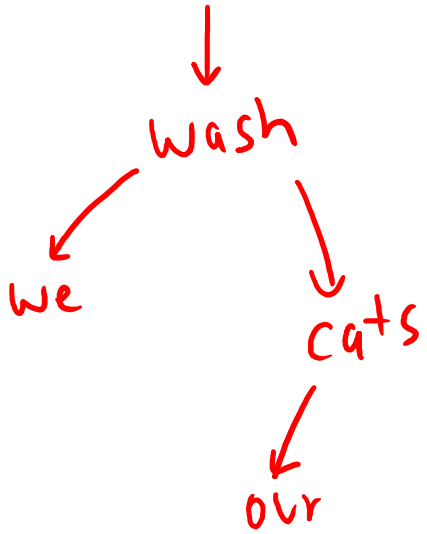
Probabilistic CFGs

Dependency Parsing

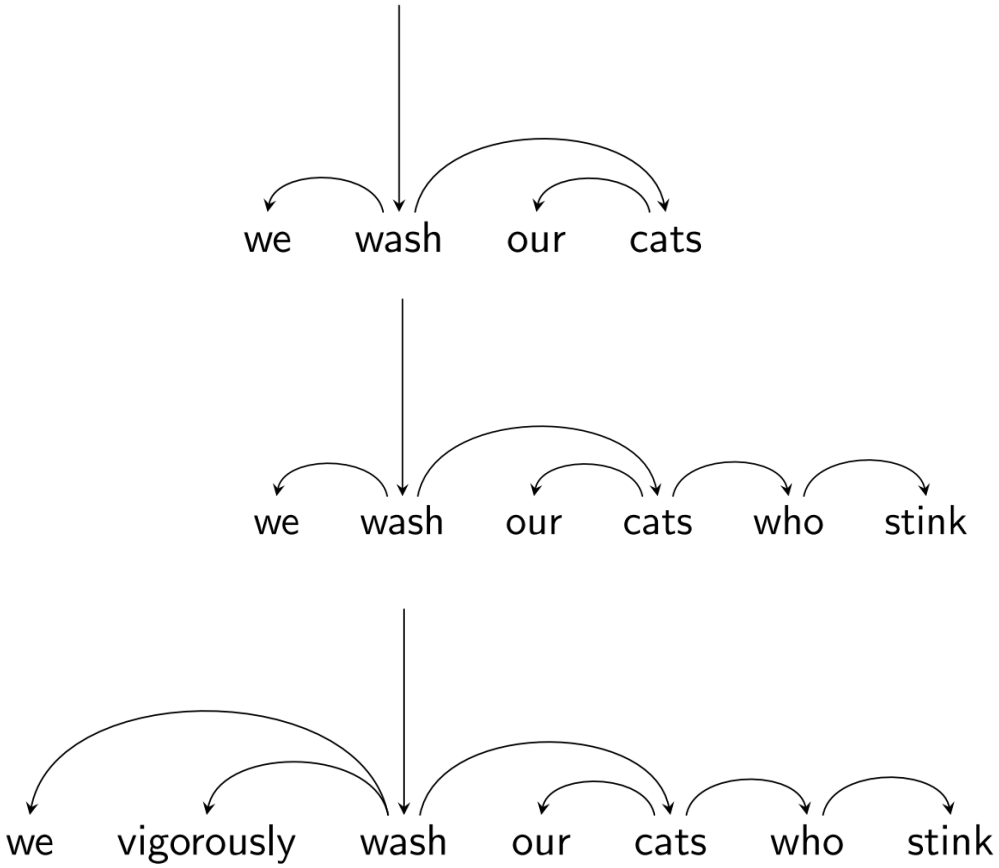
Dependencies

Represent only the syntactic dependencies…

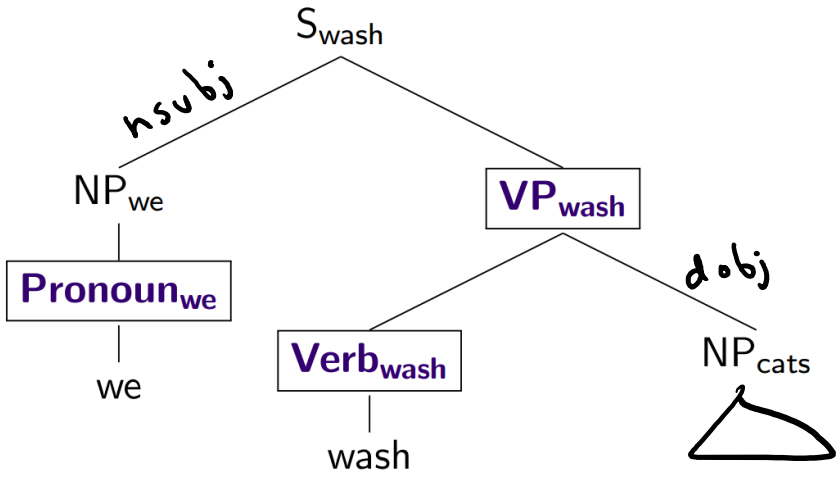


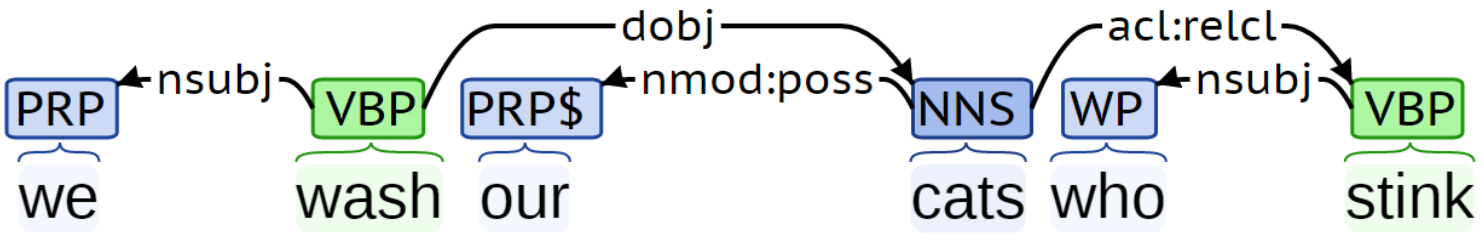


Nested Structure = Subtrees



Dependency Labels

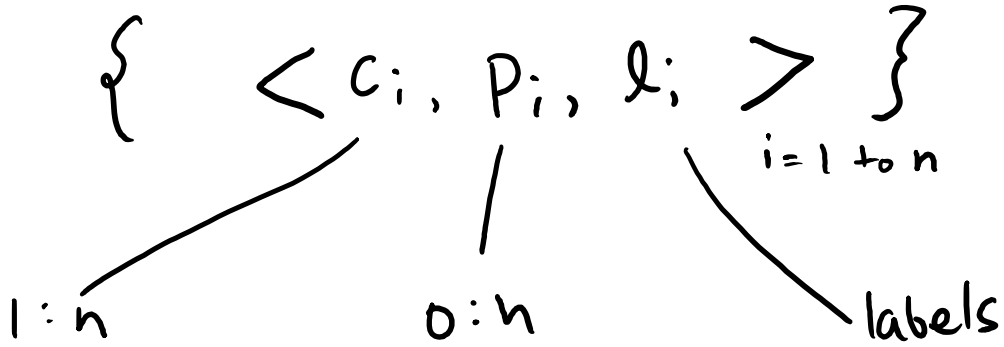




Dependency Labels

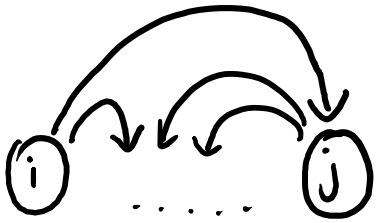
Clausal Argument Relations	Description
NSUBJ	Nominal subject
DOBJ	Direct object
IOBJ	Indirect object
CCOMP	Clausal complement
XCOMP	Open clausal complement
Nominal Modifier Relations	**Description**
NMOD	Nominal modifier
AMOD	Adjectival modifier
NUMMOD	Numeric modifier
APPOS	Appositional modifier
DET	Determiner
CASE	Prepositions, postpositions and other case markers
Other Notable Relations	**Description**
CONJ	Conjunct
CC	Coordinating conjunction

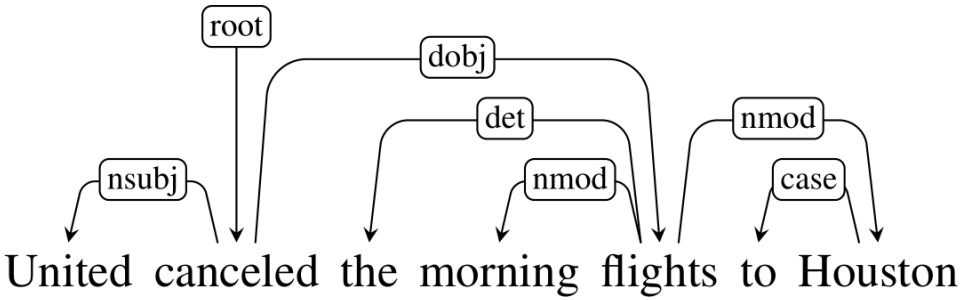
Dependency Trees

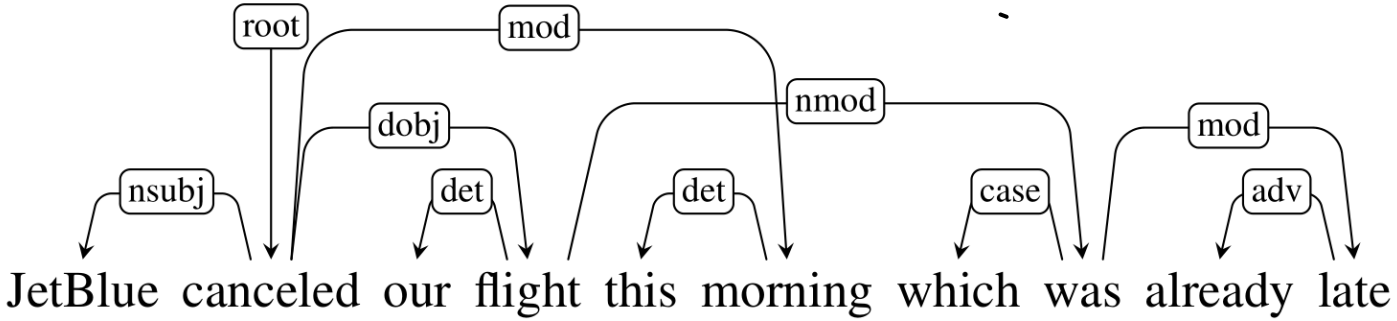


- single-head
- connected
- acyclic
- projective vs non-proj

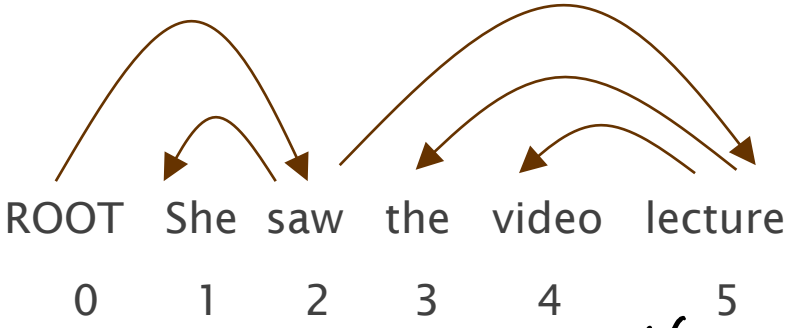
Projective vs Non-projective





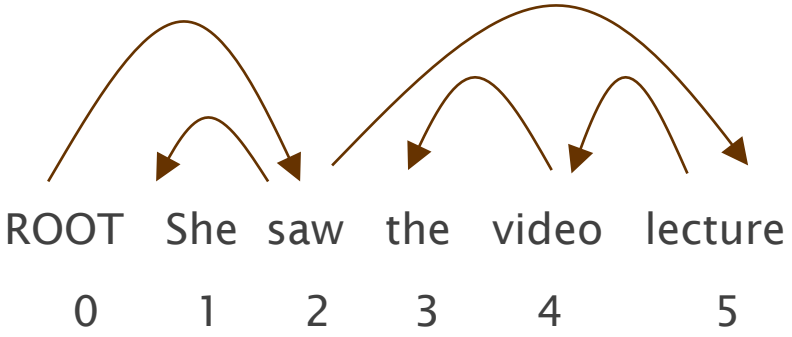


Evaluating Dependency Parses



80% UAS

Gold			
1	2	She	nsubj
2	0	saw	root
3	5	the	det
4	5	video	nn
5	2	lecture	dobj



LAS 40%

Parsed			
1	2	She	nsubj ✓
2	0	saw	root ✓
3	4	the	det ✗
4	5	video	nsubj ✗
5	2	lecture	ccomp ✗

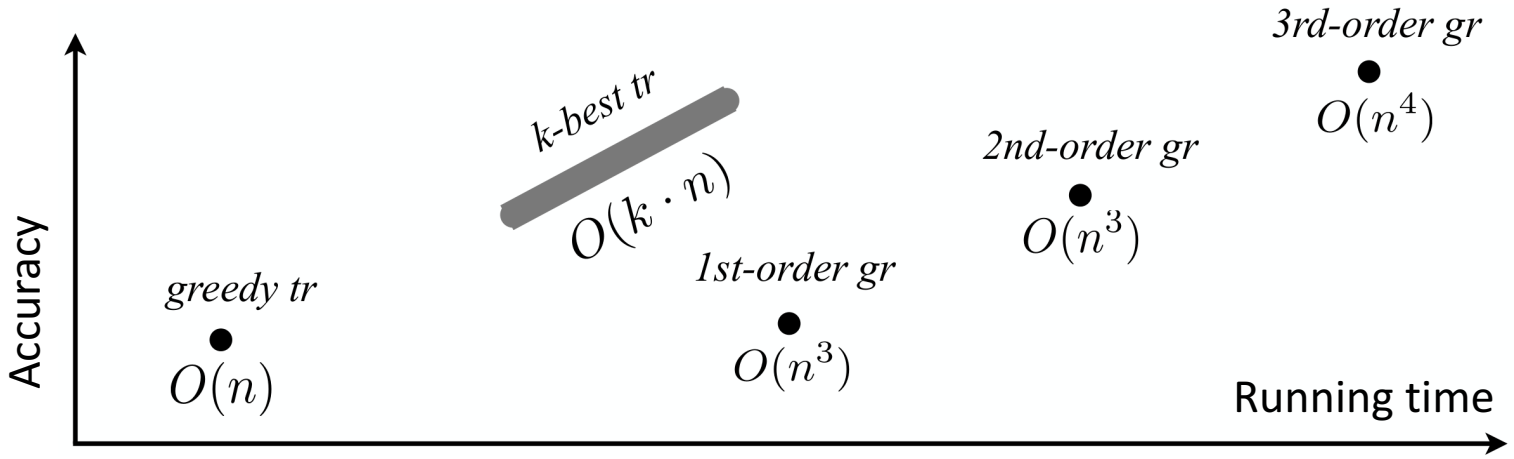
Parsing Algorithms

Transition-based

- Fast, greedy, linear-time
- Trained for greedy search
- Features decide what to do next
- Beam search, i.e. *k*-best

Graph-based

- Slower, exhaustive algorithms
- Dynamic programming, inference
- Features used to score whole trees



Transition-based Parsing

Step	Stack	Word List	Action	Relation Added
0	[root]	[book, me, the, morning, flight]	SHIFT	

Transition-based Parsing

Step	Stack	Word List	Action	Relation Added
0	[root]	[book, me, the, morning, flight]	SHIFT	
1	[root, book]	[me, the, morning, flight]	SHIFT	

Transition-based Parsing

Step	Stack	Word List	Action	Relation Added
0	[root]	[book, me, the, morning, flight]	SHIFT	
1	[root, book]	[me, the, morning, flight]	SHIFT	
2	[root, book, me]	[the, morning, flight]	RIGHTARC	(book → me)

Transition-based Parsing

Step	Stack	Word List	Action	Relation Added
0	[root]	[book, me, the, morning, flight]	SHIFT	
1	[root, book]	[me, the, morning, flight]	SHIFT	
2	[root, book, me]	[the, morning, flight]	RIGHTARC	(book → me)
3	[root, book]	[the, morning, flight]	SHIFT	
4	[root, book, the]	[morning, flight]	SHIFT	
5	[root, book, the, morning]	[flight]	SHIFT	

Transition-based Parsing

Step	Stack	Word List	Action	Relation Added
0	[root]	[book, me, the, morning, flight]	SHIFT	
1	[root, book]	[me, the, morning, flight]	SHIFT	
2	[root, book, me]	[the, morning, flight]	RIGHTARC	(book → me)
3	[root, book]	[the, morning, flight]	SHIFT	
4	[root, book, the]	[morning, flight]	SHIFT	
5	[root, book, the, morning]	[flight]	SHIFT	
6	[root, book, the, morning, flight]	[]	LEFTARC	(morning ← flight)
7	[root, book, the, flight]	[]	LEFTARC	(the ← flight)

Transition-based Parsing

Step	Stack	Word List	Action	Relation Added
0	[root]	[book, me, the, morning, flight]	SHIFT	
1	[root, book]	[me, the, morning, flight]	SHIFT	
2	[root, book, me]	[the, morning, flight]	RIGHTARC	(book → me)
3	[root, book]	[the, morning, flight]	SHIFT	
4	[root, book, the]	[morning, flight]	SHIFT	
5	[root, book, the, morning]	[flight]	SHIFT	
6	[root, book, the, morning, flight]	[]	LEFTARC	(morning ← flight)
7	[root, book, the, flight]	[]	LEFTARC	(the ← flight)
8	[root, book, flight]	[]	RIGHTARC	(book → flight)
9	[root, book]	[]	RIGHTARC	(root → book)
10	[root]	[]	Done	

Graph-based Parsing

$$\operatorname*{argmax}_{t \in T_x} \text{score}(t)$$

- 1st order / arc-factored
  $$\sum_e \theta \cdot \phi(p_e, c_e, l_e)$$
  - Proj: Dynamic P
  - NProj: Maximum Spanning Tree
- 2nd $\sum_{e_1, e_2} \text{score}(e_1, e_2)$
- 3rd $\sum_{e_1, e_2, e_3} \text{score}(e_1, e_2, e_3)$

Eisner Algorithm

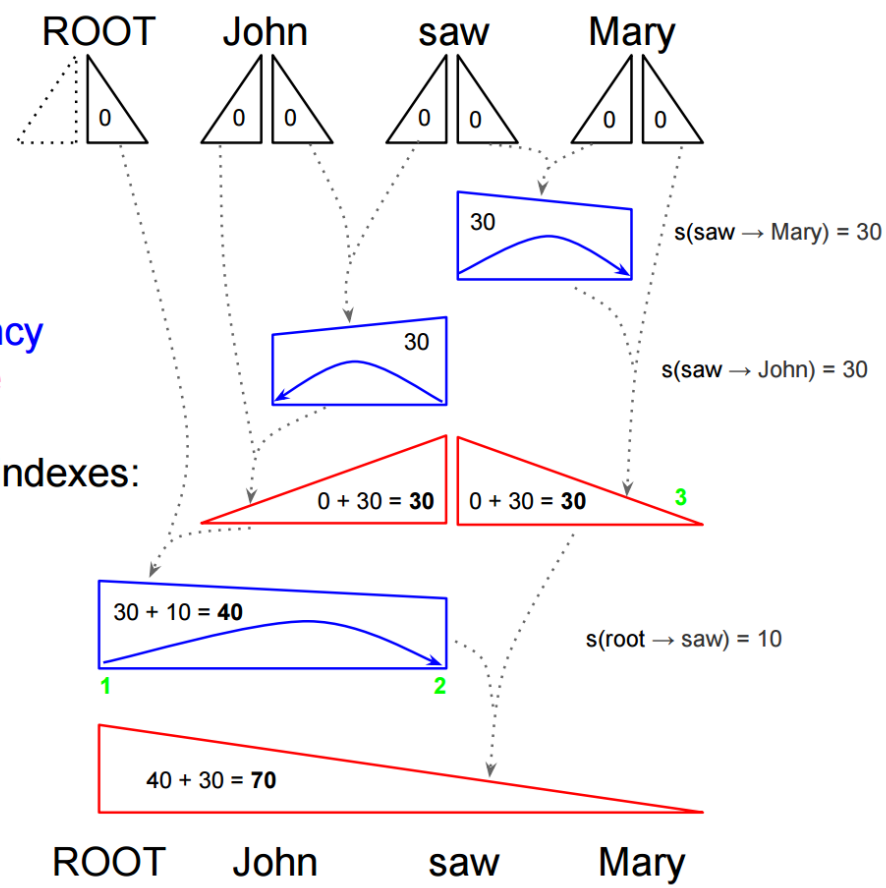
Eisner algorithm

[Eisner 1996]

Chart items either:
1) Create a new dependency
2) Absorb left/right subtree

Each chart item store two indexes:
1) left boundary
2) right boundary

All operations require
3 indexes: $\mathbf{O(n^3)}$



Upcoming...

Homework

- Homework 2 is due on Monday
- Grades for Homework 1 will be released today.

Project

- Status report due in ~2 weeks: February 21, 2017
- Instructions coming soon
- Only 5 pages

Summaries

- Paper summaries: February 17, February 28, March 14
- Only 1 page each