Probabilistic Context Free Grammars

Lecture #14

Computational Linguistics
CMPSCI 591N, Spring 2006

Andrew McCallum

(including slides from Jason Eisner)

Ambiguity in Parsing

- Time flies like an arrow.
- Fruit flies like a banana.

- I saw the man with the telescope.

How to solve this combinatorial explosion of ambiguity?

1. First try parsing without any weird rules, throwing them in only if needed.
2. Better: every rule has a weight.
   A tree's weight is total weight of all its rules.
   Pick the overall "lightest" parse of sentence.
3. Can we pick the weights automatically?
   We'll get to this later …

CYK Parser

Input: A string of words, grammar in CNF
Output: yes/no
Data structure: n x n table

- rows labeled 0 to n-1, columns 1 to n
- cell (i,j) lists constituents spanning i,j

For each i from 1 to n

- Add to (i-1,i) all Nonterminals that could produce the word at (i-1,i)

time 1 flies 2 like 3 an 4 arrow 5

	1	2	3	4	5
0	NP 3 Vst 3				
1		NP 4 VP 4			
2			P 2 V 5		
3				Det 1	
4					N 8

1 S $\rightarrow$ NP VP
6 S $\rightarrow$ Vst NP
2 S $\rightarrow$ S PP
1 VP $\rightarrow$ V NP
2 VP $\rightarrow$ VP PP
1 NP $\rightarrow$ Det N
2 NP $\rightarrow$ NP PP
3 NP $\rightarrow$ NP NP
0 PP $\rightarrow$ P NP

CYK Parser

For **width** from 2 to n

For **start** from 0 to n-width

Define **end** to be **start+width**

For **mid** from **start+1** to **end-1**

For every constituent in (**start, mid**)

For every constituent in (**mid,end**)

For all ways of combining them (if any)

Add the resulting constituent to (**start,end**).

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3				
1		NP 4 VP 4			
2			P 2 V 5		
3				Det 1	
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP
2 VP → VP PP
1 NP → Det N
2 NP → NP PP
3 NP → NP NP
0 PP → P NP

time 1 flies 2 like 3 an 4 arrow 5

	1	2	3	4	5
0	NP 3 Vst 3	NP 10			
1		NP 4 VP 4			
2			P 2 V 5		
3				Det 1	
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP
2 VP → VP PP
1 NP → Det N
2 NP → NP PP
3 NP → NP NP
0 PP → P NP

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8			
1		NP 4 VP 4			
2			P 2 V 5		
3				Det 1	
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP
2 VP → VP PP
1 NP → Det N
2 NP → NP PP
3 NP → NP NP
0 PP → P NP

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8 S 13			
1		NP 4 VP 4			
2			P 2 V 5		
3				Det 1	
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP
2 VP → VP PP
1 NP → Det N
2 NP → NP PP
3 NP → NP NP
0 PP → P NP

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8 S 13			
1		NP 4 VP 4			
2			P 2 V 5		
3				Det 1	
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP
2 VP → VP PP
1 NP → Det N
2 NP → NP PP
3 NP → NP NP
0 PP → P NP

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8 S 13			
1		NP 4 VP 4			
2			P 2 V 5		
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP

2 VP → VP PP
1 NP → Det N

2 NP → NP PP
3 NP → NP NP
0 PP → P NP

time 1 flies 2 like 3 an 4 arrow 5

	1	2	3	4	5
0	NP 3 Vst 3	NP 10 S 8 S 13			
1		NP 4 VP 4			
2			P 2 V 5		
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP
2 VP → VP PP
1 NP → Det N
2 NP → NP PP
3 NP → NP NP
0 PP → P NP

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8 S 13			
1		NP 4 VP 4			
2			P 2 V 5		PP 12
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP

2 VP → VP PP
1 NP → Det N

2 NP → NP PP
3 NP → NP NP
0 PP → P NP

time 1 flies 2 like 3 an 4 arrow 5

	1	2	3	4	5
0	NP 3 Vst 3	NP 10 S 8 S 13			
1		NP 4 VP 4			
2			P 2 V 5		PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP

2 VP → VP PP
1 NP → Det N

2 NP → NP PP
3 NP → NP NP
0 PP → P NP

time 1 flies 2 like 3 an 4 arrow 5

0	NP 3 Vst 3	NP 10 S 8 S 13			
1		NP 4 VP 4			
2			P 2 V 5		PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP
2 VP → VP PP
1 NP → Det N
2 NP → NP PP
3 NP → NP NP
0 PP → P NP

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8 S 13			
1		NP 4 VP 4			NP 18
2			P 2 V 5		PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP
2 VP → VP PP
1 NP → Det N
2 NP → NP PP
3 NP → NP NP
0 PP → P NP

time 1 flies 2 like 3 an 4 arrow 5

0	NP 3 Vst 3	NP 10 S 8 S 13			
1		NP 4 VP 4			NP 18 S 21
2			P 2 V 5		PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP
2 VP → VP PP
1 NP → Det N
2 NP → NP PP
3 NP → NP NP
0 PP → P NP

time 1 flies 2 like 3 an 4 arrow 5

	1	2	3	4	5
0	NP 3 Vst 3	NP 10 S 8 S 13			
1		NP 4 VP 4			NP 18 S 21 VP 18
2			P 2 V 5		PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP

2 VP → VP PP
1 NP → Det N

2 NP → NP PP
3 NP → NP NP
0 PP → P NP

time 1 flies 2 like 3 an 4 arrow 5

	1	2	3	4	5
0	NP 3 Vst 3	NP 10 S 8 S 13			
1		NP 4 VP 4			NP 18 S 21 VP 18
2			P 2 V 5		PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP
2 VP → VP PP
1 NP → Det N
2 NP → NP PP
3 NP → NP NP
0 PP → P NP

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8 S 13			NP 24
1		NP 4 VP 4			NP 18 S 21 VP 18
2			P 2 V 5		PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP

2 VP → VP PP
1 NP → Det N

2 NP → NP PP
3 NP → NP NP
0 PP → P NP

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8 S 13			NP 24 S 22
1		NP 4 VP 4			NP 18 S 21 VP 18
2			P 2 V 5		PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP

2 VP → VP PP
1 NP → Det N

2 NP → NP PP
3 NP → NP NP
0 PP → P NP

time 1 flies 2 like 3 an 4 arrow 5

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8 S 13			NP 24 S 22 S 27
1		NP 4 VP 4			NP 18 S 21 VP 18
2			P 2 V 5		PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP
2 VP → VP PP
1 NP → Det N
2 NP → NP PP
3 NP → NP NP
0 PP → P NP

time 1 flies 2 like 3 an 4 arrow 5

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8 S 13			NP 24 S 22 S 27
1		NP 4 VP 4			NP 18 S 21 VP 18
2			P 2 V 5		PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP

2 VP → VP PP
1 NP → Det N

2 NP → NP PP
3 NP → NP NP
0 PP → P NP

time 1 flies 2 like 3 an 4 arrow 5

	time (1)	flies (2)	like (3)	an (4)	arrow (5)
0	NP 3 Vst 3	NP 10 S 8 S 13			NP 24 S 22 S 27 NP 24
1		NP 4 VP 4			NP 18 S 21 VP 18
2			P 2 V 5		PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP
2 VP → VP PP
1 NP → Det N
2 NP → NP PP
3 NP → NP NP
0 PP → P NP

time 1 flies 2 like 3 an 4 arrow 5

	1	2	3	4	5
0	NP 3 Vst 3	NP 10 S 8 S 13			NP 24 S 22 S 27 NP 24 S 27
1		NP 4 VP 4			NP 18 S 21 VP 18
2			P 2 V 5		PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP

2 VP → VP PP
1 NP → Det N

2 NP → NP PP
3 NP → NP NP
0 PP → P NP

time 1 flies 2 like 3 an 4 arrow 5

	1	2	3	4	5
0	NP 3 Vst 3	NP 10 S 8 S 13			NP 24 S 22 S 27 NP 24 S 27 S 22
1		NP 4 VP 4			NP 18 S 21 VP 18
2			P 2 V 5		PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP

2 VP → VP PP
1 NP → Det N

2 NP → NP PP
3 NP → NP NP
0 PP → P NP

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8 S 13			NP 24 S 22 S 27 NP 24 S 27 S 22 S 27
1		NP 4 VP 4			NP 18 S 21 VP 18
2			P 2 V 5		PP 12 VP 16
3				Det 1	NP 10
4					N 8

1	S	→ NP VP
6	S	→ Vst NP
2	S	→ S PP
1	VP	→ V NP
2	VP	→ VP PP
1	NP	→ Det N
2	NP	→ NP PP
3	NP	→ NP NP
0	PP	→ P NP

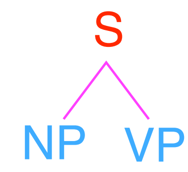
S

Follow backpointers ...

time 1 flies 2 like 3 an 4 arrow 5

0	NP 3 Vst 3	NP 10 S 8 S 13			NP 24 S 22 S 27 NP 24 S 27 S 22 S 27
1		NP 4 VP 4			NP 18 S 21 VP 18
2			P 2 V 5		PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP
2 VP → VP PP
1 NP → Det N
2 NP → NP PP
3 NP → NP NP
0 PP → P NP

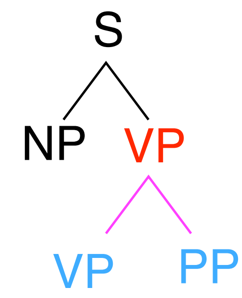


	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8 S 13			NP 24 S 22 S 27 NP 24 S 27 S 22 S 27
1		NP 4 VP 4			NP 18 S 21 VP 18
2			P 2 V 5		PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP
2 VP → VP PP
1 NP → Det N
2 NP → NP PP
3 NP → NP NP
0 PP → P NP

time 1 flies 2 like 3 an 4 arrow 5

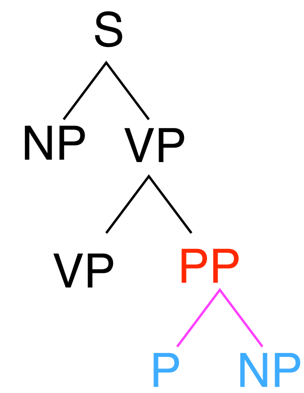
	1	2	3	4	5
0	NP 3 Vst 3	NP 10 S 8 S 13			NP 24 S 22 S 27 NP 24 S 27 S 22 S 27
1		NP 4 VP 4			NP 18 S 21 VP 18
2			P 2 V 5		PP 12 VP 16
3				Det 1	NP 10
4					N 8



1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP
2 VP → VP PP
1 NP → Det N
2 NP → NP PP
3 NP → NP NP
0 PP → P NP

time 1 flies 2 like 3 an 4 arrow 5

	1	2	3	4	5
0	NP 3 Vst 3	NP 10 S 8 S 13			NP 24 S 22 S 27 NP 24 S 27 S 22 S 27
1		NP 4 VP 4			NP 18 S 21 VP 18
2			P 2 V 5		PP 12 VP 16
3				Det 1	NP 10
4					N 8



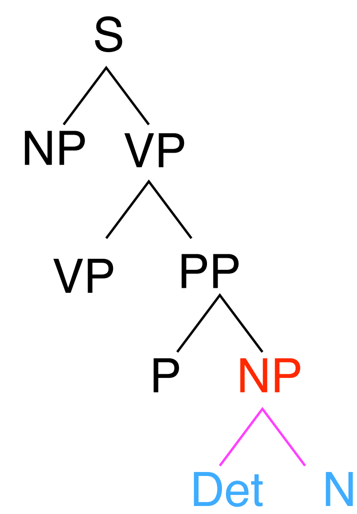
1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP

2 VP → VP PP
1 NP → Det N

2 NP → NP PP
3 NP → NP NP

0 PP → P NP

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8 S 13			NP 24 S 22 S 27 NP 24 S 27 S 22 S 27
1		NP 4 VP 4			NP 18 S 21 VP 18
2			P 2 V 5		PP 12 VP 16
3				Det 1	NP 10
4					N 8



1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP
2 VP → VP PP
1 NP → Det N
2 NP → NP PP
3 NP → NP NP
0 PP → P NP

Which entries do we need?

time 1 flies 2 like 3 an 4 arrow 5

0	NP 3 Vst 3	NP 10 S 8 S 13			NP 24 S 22 S 27 NP 24 S 27 S 22 S 27
1		NP 4 VP 4			NP 18 S 21 VP 18
2			P 2 V 5		PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP
2 VP → VP PP
1 NP → Det N
2 NP → NP PP
3 NP → NP NP
0 PP → P NP

Which entries do we need?

time 1 flies 2 like 3 an 4 arrow 5

0	NP 3 Vst 3	NP 10 S 8 S 13			NP 24 S 22 S 27 NP 24 S 27 S 22 S 27
1		NP 4 VP 4			NP 18 S 21 VP 18
2			P 2 V 5		PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP
2 VP → VP PP
1 NP → Det N
2 NP → NP PP
3 NP → NP NP
0 PP → P NP

Not worth keeping ...

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8 S 13			NP 24 S 22 S 27 NP 24 S 27 S 22 S 27
1		NP 4 VP 4			NP 18 S 21 VP 18
2			P 2 V 5		PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP
2 VP → VP PP
1 NP → Det N
2 NP → NP PP
3 NP → NP NP
0 PP → P NP

… since it just breeds worse options

time 1 flies 2 like 3 an 4 arrow 5

0	NP 3 Vst 3	NP 10 S 8 S 13			NP 24 S 22 S 27 NP 24 S 27 S 22 S 27
1		NP 4 VP 4			NP 18 S 21 VP 18
2			P 2 V 5		PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP
2 VP → VP PP
1 NP → Det N
2 NP → NP PP
3 NP → NP NP
0 PP → P NP

Keep only best-in-class!

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8 S 13	inferior stock		NP 24 S 22 S 27 NP 24 S 27 S 22 S 27
1		NP 4 VP 4			NP 18 S 21 VP 18
2			P 2 V 5		PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP
2 VP → VP PP
1 NP → Det N
2 NP → NP PP
3 NP → NP NP
0 PP → P NP

Keep only best-in-class!

(and backpointers so you can recover parse)

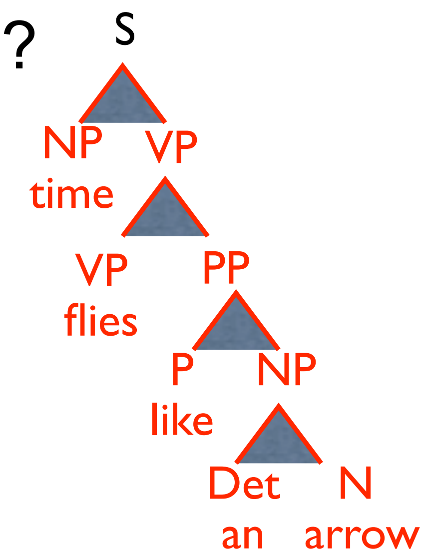
time 1 flies 2 like 3 an 4 arrow 5

	NP 3 Vst 3	NP 10 S 8			NP 24 S 22
1		NP 4 VP 4			NP 18 S 21 VP 18
2			P 2 V 5		PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP
2 VP → VP PP
1 NP → Det N
2 NP → NP PP
3 NP → NP NP
0 PP → P NP

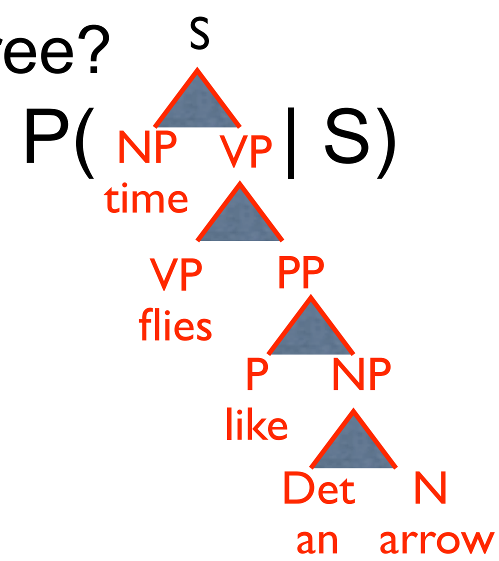
Probabilistic Trees

- Instead of lightest weight tree, take highest probability tree
- Given any tree, your assignment generator would have some probability of producing it!
- Just like using n-grams to choose among strings …
- What is the probability of this tree?



Probabilistic Trees

- Instead of lightest weight tree, take highest probability tree
- Given any tree, your assignment generator would have some probability of producing it!
- Just like using n-grams to choose among strings …
- What is the probability of this tree?



- You rolled a lot

of independent dice...

Chain rule: One word at a time

p(time flies like an arrow)

= p(time)

* p(flies | time)

* p(like | time flies)

* p(an | time flies like)

* p(arrow | time flies like an)

Chain rule + backoff (to get trigram model)

p(time flies like an arrow)
= p(time)
* p(flies | time)
* p(like | time flies)
* p(an | time flies like)
* p(arrow | time flies like an)

Chain rule – written differently

p(time flies like an arrow)

= p(time)

* p(time flies | time)

* p(time flies like | time flies)

* p(time flies like an | time flies like)

* p(time flies like an arrow | time flies like an)

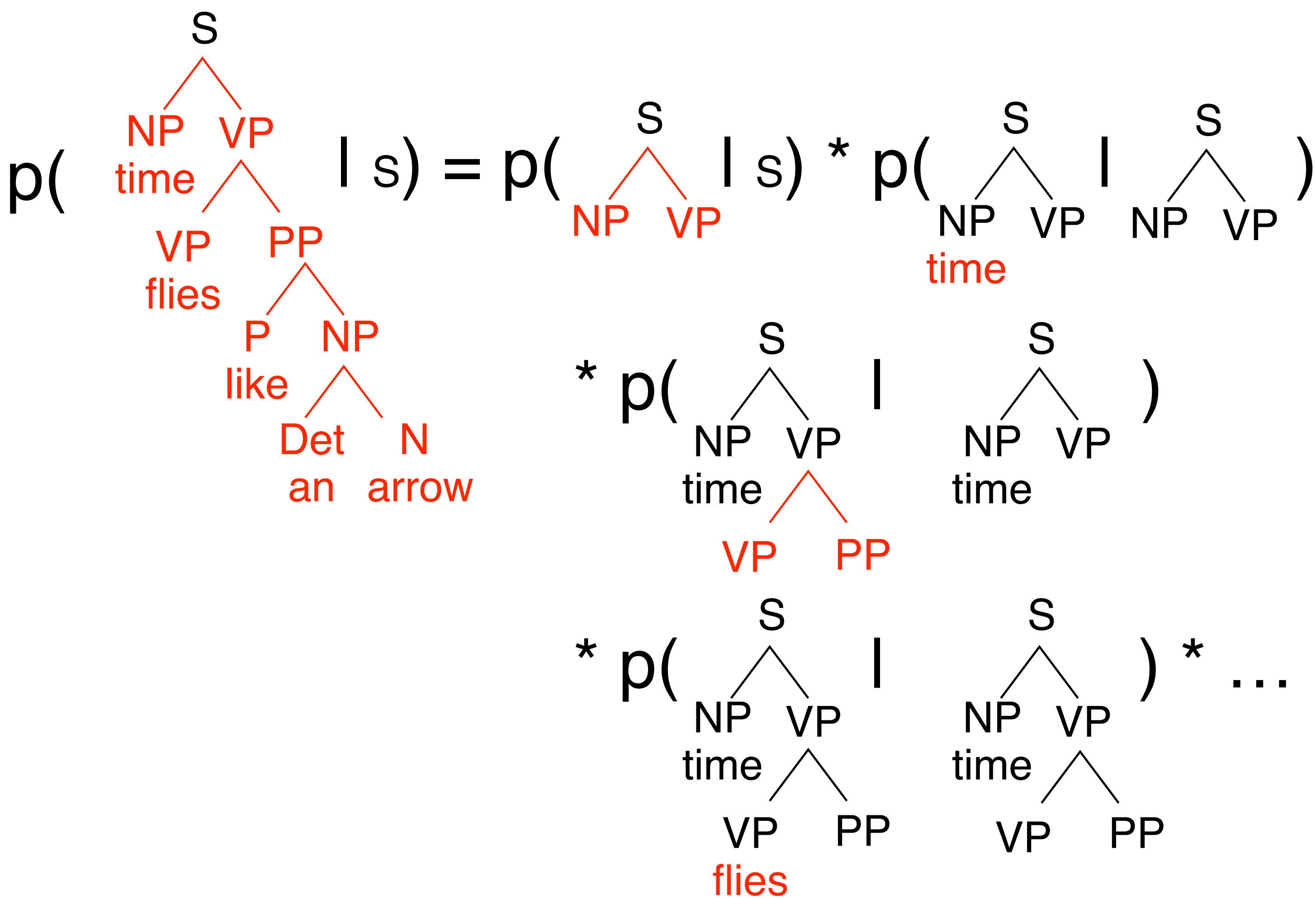
Proof: p(x,y | x) = p(x | x) * p(y | x, x) = 1 * p(y | x)

Chain rule + backoff

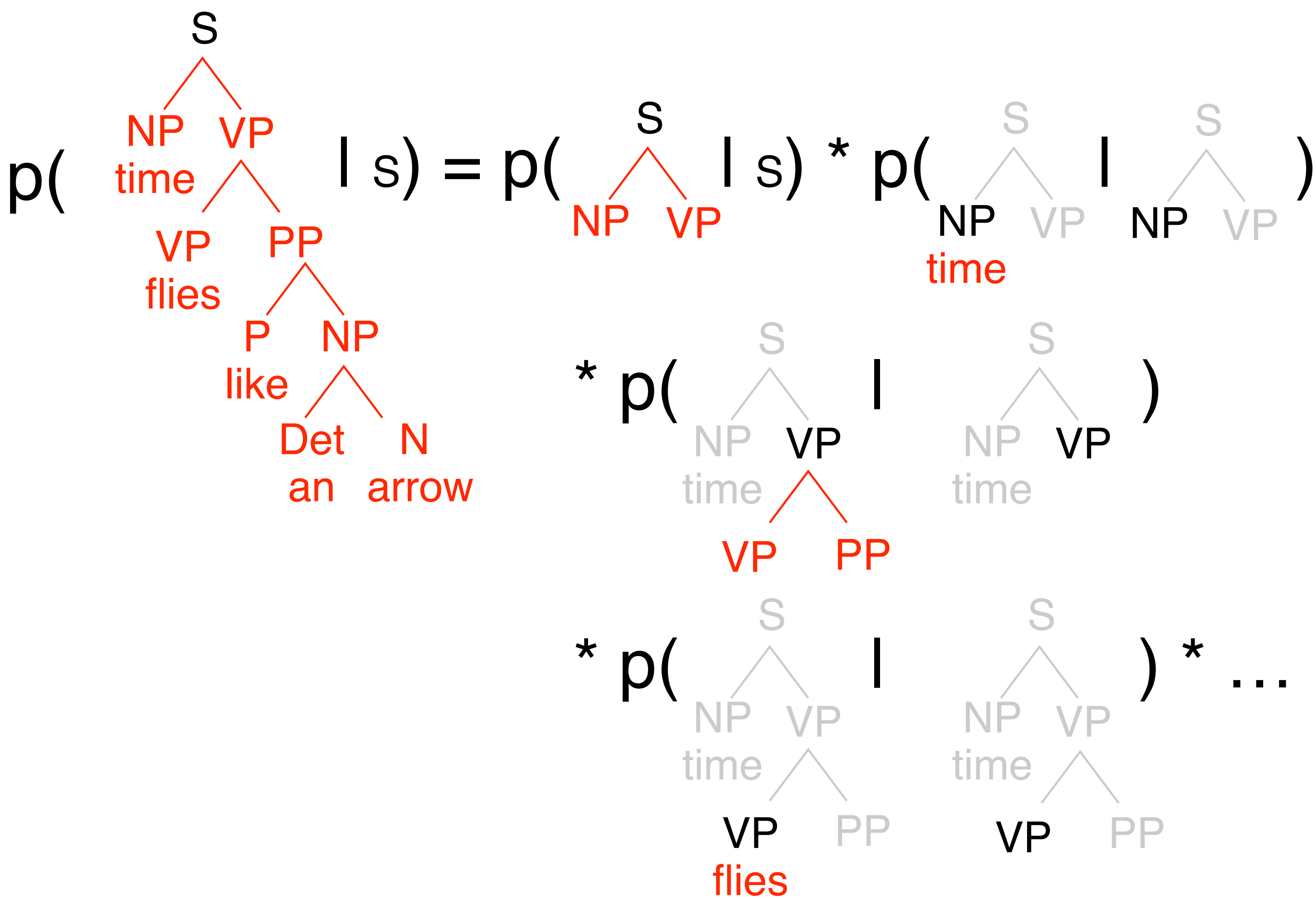
p(time flies like an arrow)
  = p(time)
  * p(time flies | time)
  * p(time flies like | time flies)
  * p(time flies like an | time flies like)
  * p(time flies like an arrow | time flies like an)

Proof: p(x,y | x) = p(x | x) * p(y | x, x) = 1 * p(y | x)

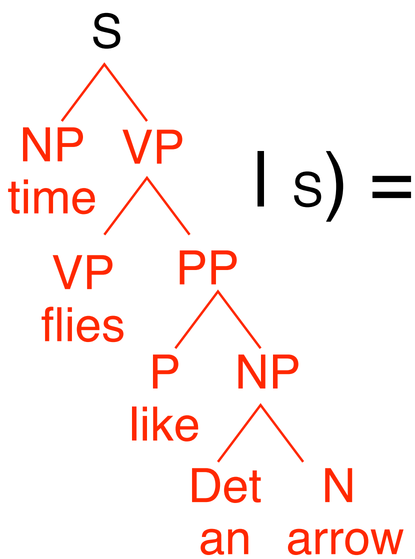
Chain rule: One node at a time



Chain rule + backoff



Simplified notation

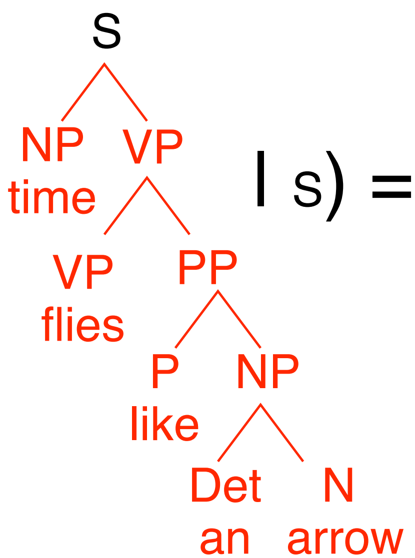
p(

$| \text{S}) = p(\text{S} \rightarrow \text{NP VP} \mid \text{S}) * p(\text{NP} \rightarrow \text{flies} \mid \text{NP})$

$* \; p(\text{VP} \rightarrow \text{VP NP} \mid \text{VP})$

$* \; p(\text{VP} \rightarrow \text{flies} \mid \text{VP}) * \ldots$

Already have a CKY alg for weights …

w(

| S) = w(S → NP VP) + w(NP → flies | NP)

+ w(VP → VP NP)

+ w(VP → flies) + …

Just let w(X → Y Z) = -log p(X → Y Z | X)

Then lightest tree has highest prob

time 1 flies 2 like 3 an 4 arrow 5

	time	flies	like	an	arrow
0	NP 3 Vst 3	NP 10 S 8 S 13			NP 24 S 22 S 27 NP 24 S 27 S 22 S 27
1		NP 4 VP 4			NP 18 S 21 VP 18
2			P 2 V 5		PP 12 VP 16
3				Det 1	NP 10
4					N 8

2^{-8}

multiply to get 2^{-22}

2^{-12}

2^{-2}

1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP
2 VP → VP PP
1 NP → Det N
2 NP → NP PP
3 NP → NP NP
0 PP → P NP

Need only best-in-class to get best parse

time 1 flies 2 like 3 an 4 arrow 5

	1	2	3	4	5
0	NP 3 Vst 3	NP 10 S 8 S 13			NP 24 S 22 S 27 NP 24 S 27 S 22 S 27
1		NP 4 VP 4			NP 18 S 21 VP 18
2			P 2 V 5		PP 12 VP 16
3				Det 1	NP 10
4					N 8

2^{-13}

2^{-8}

multiply to get 2^{-22}

2^{-12}

2^{-2}

1 S → NP VP
6 S → Vst NP
2 S → S PP
1 VP → V NP
2 VP → VP PP
1 NP → Det N
2 NP → NP PP
3 NP → NP NP
0 PP → P NP

Why probabilities not weights?

- We just saw probabilities are really just a special case of weights …
- … *but* we can estimate them from training data by counting and smoothing! Use all of our lovely probability theory machinery!

Probabilistic Context Free Grammars

A PCFG G consists of the usual parts of a CFG

- A set of terminals, $\{w^k\}, k = 1, \ldots, V$
- A set of nonterminals, $\{N^i\}, i = 1, \ldots, n$
- A designated start symbol, N^1
- A set of rules, $\{N^i \rightarrow \zeta^j\}$, (where ζ^j is a sequence of terminals and nonterminals)

and

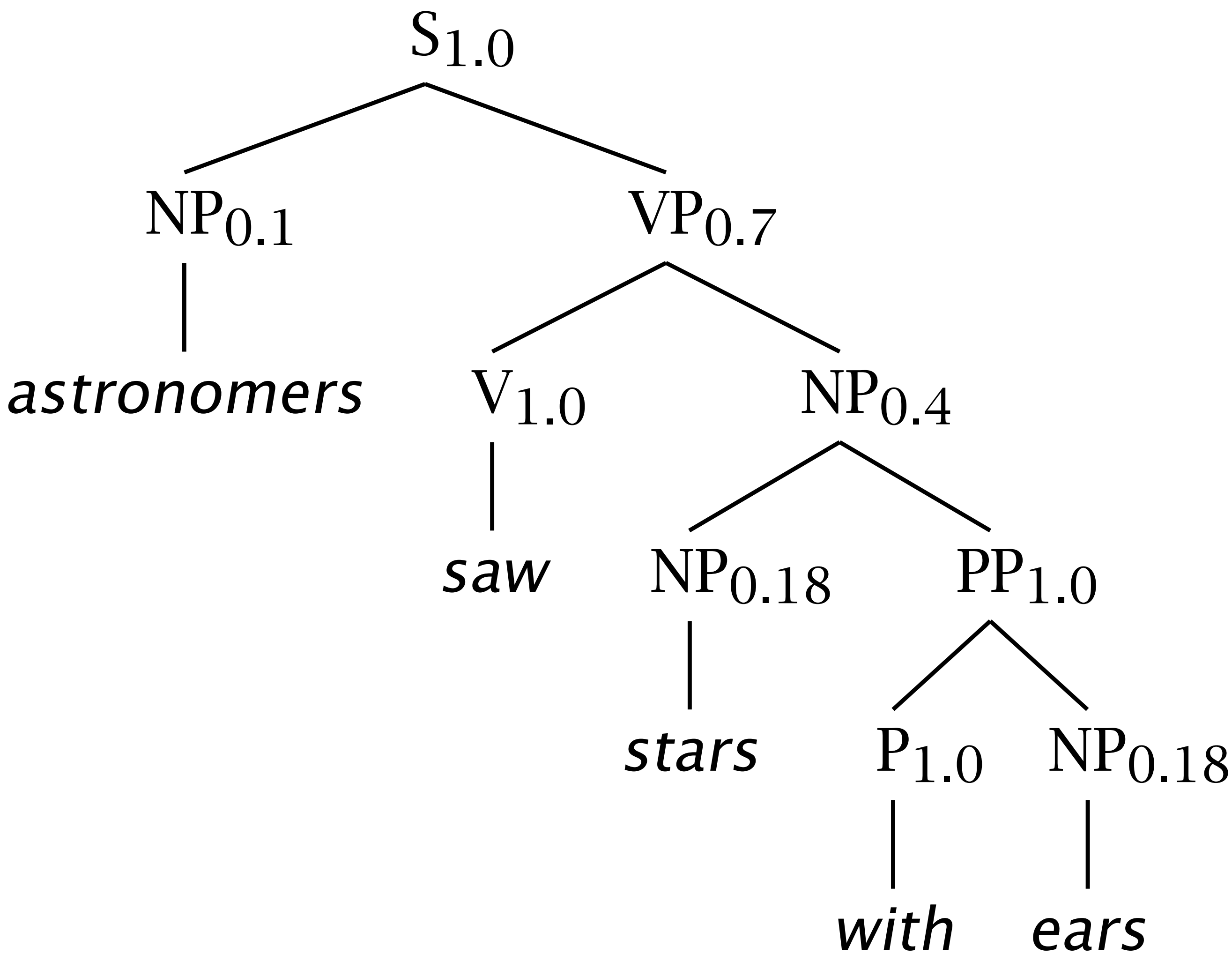
- A corresponding set of probabilities on rules such that:

$$\forall i \quad \sum_j P(N^i \rightarrow \zeta^j) = 1$$

A simple PCFG (in CNF)

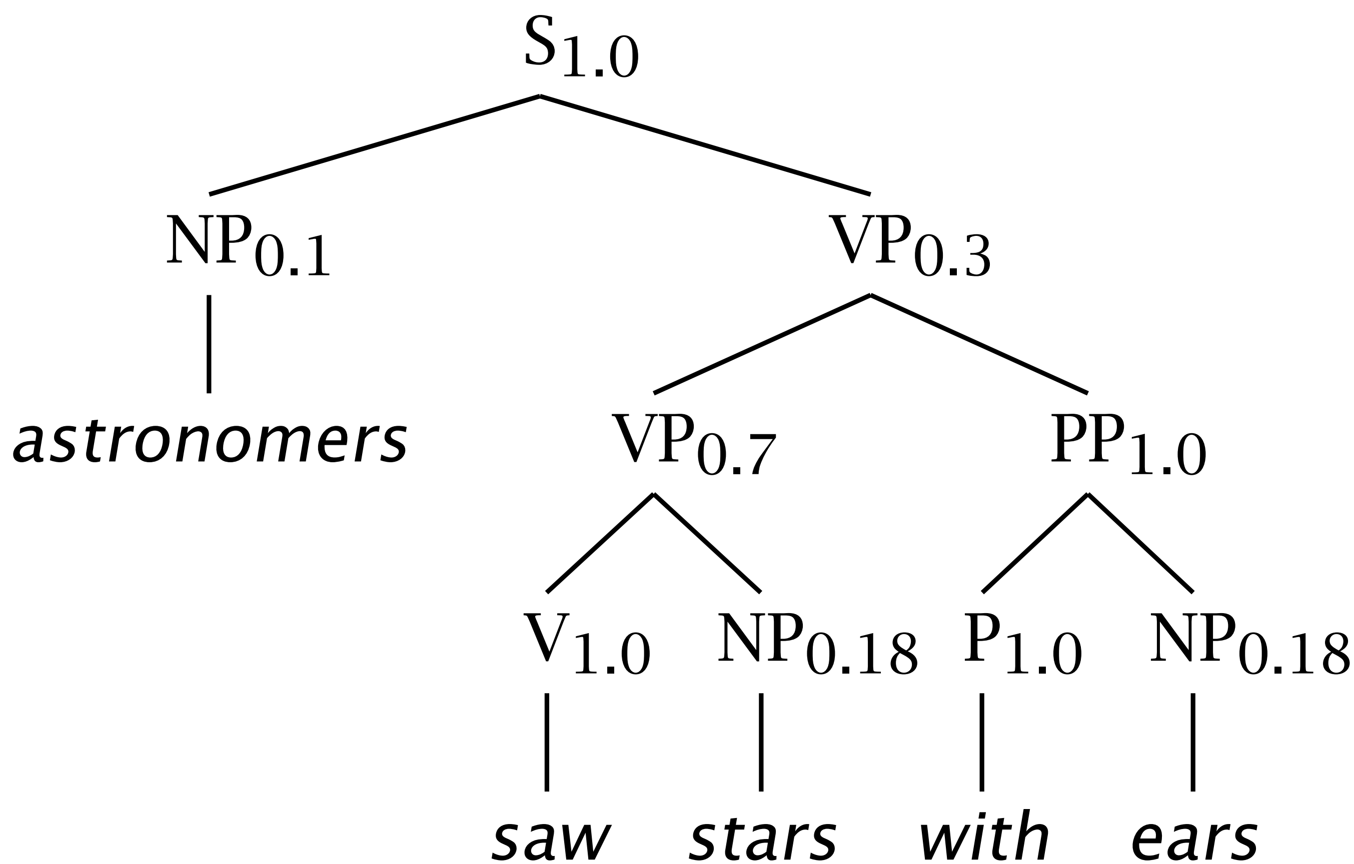
S → NP VP	1.0		NP → NP PP	0.4
PP → P NP	1.0		NP → *astronomers*	0.1
VP → V NP	0.7		NP → *ears*	0.18
VP → VP PP	0.3		NP → *saw*	0.04
P → *with*	1.0		NP → *stars*	0.18
V → *saw*	1.0		NP → *telescopes*	0.1

t_1:



S1.0
NP0.1
VP0.7
astronomers
V1.0
NP0.4
saw
NP0.18
PP1.0
stars
P1.0
NP0.18
with
ears

t_2:



S1.0
NP0.1
VP0.3
astronomers
VP0.7
PP1.0
V1.0
NP0.18
P1.0
NP0.18
saw
stars
with
ears

The two parse trees' probabilities and the sentence probability

$$
\begin{aligned}
P(t_1) &= 1.0 \times 0.1 \times 0.7 \times 1.0 \times 0.4 \\
&\quad \times 0.18 \times 1.0 \times 1.0 \times 0.18 \\
&= 0.0009072 \\
P(t_2) &= 1.0 \times 0.1 \times 0.3 \times 0.7 \times 1.0 \\
&\quad \times 0.18 \times 1.0 \times 1.0 \times 0.18 \\
&= 0.0006804 \\
P(w_{15}) &= P(t_1) + P(t_2) = 0.0015876
\end{aligned}
$$

Assumptions of PCFGs

1. Place invariance (like time invariance in HMM):

$$\forall k \quad P(N^j_{k(k+c)} \to \zeta) \text{is the same}$$

2. Context-free:

$$P(N^j_{kl} \to \zeta | \text{words outside } w_k \ldots w_l) = P(N^j_{kl} \to \zeta)$$

3. Ancestor-free:

$$P(N^j_{kl} \to \zeta | \text{ancestor nodes of } N^j_{kl}) = P(N^j_{kl} \to \zeta)$$

The sufficient statistics of a PCFG are thus simply counts of how often different local tree configurations occurred (= counts of which grammar rules were applied).

Some features of PCFGs

Reasons to use a PCFG, and some idea of their limitations:

- Partial solution for grammar ambiguity: a PCFG gives some idea of the plausibility of a sentence.
- But, in the simple case, not a very good idea, as independence assumptions are two strong (e.g., not lexicalized).
- Gives a probabilistic language model for English.
- In the simple case, a PCFG is a worse language model for English than a trigram model.
- Better for grammar induction (Gold 1967 vs. Horning 1969)
- Robustness. (Admit everything with low probability.)

Some features of PCFGs

- A PCFG encodes certain biases, e.g., that smaller trees are normally more probable.
- One can hope to combine the strengths of a PCFG and a trigram model.

We'll look at simple PCFGs first. They have certain inadequacies, but we'll see that most of the state-of-the-art probabilistic parsers are fundamentally PCFG models, just with various enrichments to the grammar