ILLINOIS INSTITUTE OF TECHNOLOGY

Probabilistic CFG Parsing

CS-585

Natural Language Processing

Derrick Higgins

PROBABILISTIC CONTEXT-FREE GRAMMARS

Parsing and ambiguity

- We saw how to generate tree structures (an arbitrary one or all of them) using the CYK algorithm
- But is a large set of potential structures very useful?

➢ Time flies like an arrow.	`NP VP`
➢ Fruit flies like a banana.	`NP VP`
➢ Time reactions like this one.	`V[stem] NP`
➢ Time reactions like a chemist.	`S PP`

Example from Jason Eisner, JHU CS 600.465

Potential approach for dealing with ambiguity

- Some rules/structures are less common/likely than others
- Associate each rule with a weight/cost
  - Rules with lower weights are preferred
  - Cost for structure is sum of weights of all rules used
  - Choose the structure with lowest cost
- How to select the weights? TBD

Time_0 flies_1 like_2 an_3 arrow_4

n (constituent start index)

m (constituent length -1)

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

```
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$_0$ flies$_1$ like$_2$ an$_3$ arrow$_4$

n (constituent start index)

m (constituent length -1)

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

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

Time$_0$ flies$_1$ like$_2$ an$_3$ arrow$_4$

n (constituent start index)

m (constituent length -1)

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

```
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$_0$ flies$_1$ like$_2$ an$_3$ arrow$_4$

n (constituent start index)

m (constituent length -1)

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

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

Time$_0$ flies$_1$ like$_2$ an$_3$ arrow$_4$

n (constituent start index)

m (constituent length -1)

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

```
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$_0$ flies$_1$ like$_2$ an$_3$ arrow$_4$

n (constituent start index)

m (constituent length -1)

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

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$_0$ flies$_1$ like$_2$ an$_3$ arrow$_4$

n (constituent start index)

m (constituent length -1)

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

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$_0$ flies$_1$ like$_2$ an$_3$ arrow$_4$

n (constituent start index)

m (constituent length -1)

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

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

Time$_0$ flies$_1$ like$_2$ an$_3$ arrow$_4$

n (constituent start index)

m (constituent length -1)

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

```
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_0 flies_1 like_2 an_3 arrow_4

n (constituent start index)

m (constituent length -1)

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

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

Time$_0$ flies$_1$ like$_2$ an$_3$ arrow$_4$

n (constituent start index)

m (constituent length -1)

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

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$_0$ flies$_1$ like$_2$ an$_3$ arrow$_4$

n (constituent start index)

m (constituent length -1)

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

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

Time$_0$ flies$_1$ like$_2$ an$_3$ arrow$_4$

n (constituent start index)

m (constituent length -1)

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

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$_0$ flies$_1$ like$_2$ an$_3$ arrow$_4$

n (constituent start index)

m (constituent length -1)

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

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$_0$ flies$_1$ like$_2$ an$_3$ arrow$_4$

n (constituent start index)

m (constituent length -1)

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

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

Time$_0$ flies$_1$ like$_2$ an$_3$ arrow$_4$

n (constituent start index)

m (constituent length -1)

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

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$_0$ flies$_1$ like$_2$ an$_3$ arrow$_4$

n (constituent start index)

m (constituent length -1)

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

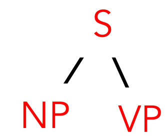
1 S $\rightarrow$ NP VP
6 S $\rightarrow$ Vst NP
2 S $\rightarrow$ S PP
1 VP $\rightarrow$ V NP
2 VP $\rightarrow$ VP PP
1 NP $\rightarrow$ Det N
2 NP $\rightarrow$ NP PP
3 NP $\rightarrow$ NP NP
0 PP $\rightarrow$ P NP

Time$_0$ flies$_1$ like$_2$ an$_3$ arrow$_4$

n (constituent start index)

m (constituent length -1)

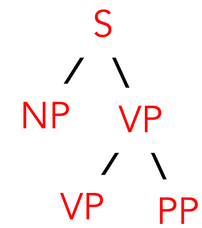
	0	1	2	3	4
0	NP 3 Vst 3	NP 4 VP 4	P 2 V 5	Det 1	N 8
1	NP 10 S 8 S 13			NP 10	
2			PP 12 VP 16		
3		NP 18 S 21 VP 18			
4	NP 24 S 22 S 27 NP 24 S 27 S 22 S 27				



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

Use back-pointers to recover best parse

	Time$_0$	flies$_1$	like$_2$	an$_3$	arrow$_4$
	n **(constituent start index)**				
m **(constituent length -1)**	**0**	**1**	**2**	**3**	**4**
0	NP 3 Vst 3	NP 4 VP 4	P 2 V 5	Det 1	N 8
1	NP 10 S 8 S 13			NP 10	
2			PP 12 VP 16		
3		NP 18 S 21 VP 18			
4	NP 24 S 22 S 27 NP 24 S 27 S 22 S 27				



1 S $\longrightarrow$ NP VP
6 S $\longrightarrow$ Vst NP
2 S $\longrightarrow$ S PP

1 VP $\longrightarrow$ V NP

2 VP $\longrightarrow$ VP PP

1 NP $\longrightarrow$ Det N
2 NP $\longrightarrow$ NP PP
3 NP $\longrightarrow$ NP NP

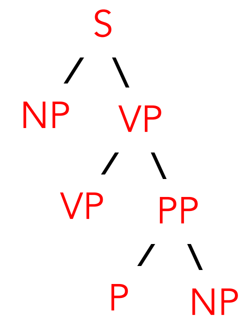
0 PP $\longrightarrow$ P NP

Time$_0$ flies$_1$ like$_2$ an$_3$ arrow$_4$

n (constituent start index)

m (constituent length -1)

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



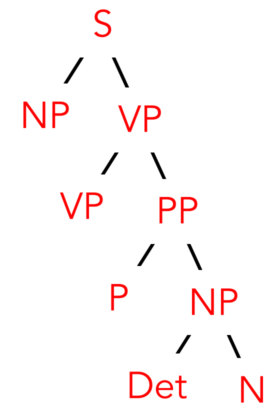
1	S	$\longrightarrow$	NP VP
6	S	$\longrightarrow$	Vst NP
2	S	$\longrightarrow$	S PP
1	VP	$\longrightarrow$	V NP
2	VP	$\longrightarrow$	VP PP
1	NP	$\longrightarrow$	Det N
2	NP	$\longrightarrow$	NP PP
3	NP	$\longrightarrow$	NP NP
0	PP	$\longrightarrow$	P NP

Time$_0$ flies$_1$ like$_2$ an$_3$ arrow$_4$

n (constituent start index)

m (constituent length -1)

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



1 S $\longrightarrow$ NP VP
6 S $\longrightarrow$ Vst NP
2 S $\longrightarrow$ S PP

1 VP $\longrightarrow$ V NP

2 VP $\longrightarrow$ VP PP

1 NP $\longrightarrow$ Det N
2 NP $\longrightarrow$ NP PP
3 NP $\longrightarrow$ NP NP

0 PP $\longrightarrow$ P NP

Time$_0$ flies$_1$ like$_2$ an$_3$ arrow$_4$

n (constituent start index)

m (constituent length -1)

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

Which entries do we actually need?

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$_0$	flies$_1$	like$_2$	an$_3$	arrow$_4$
m (constituent length -1) \ *n* (constituent start index)	0	1	2	3	4
0	NP 3 Vst 3	NP 4 VP 4	P 2 V 5	Det 1	N 8
1	NP 10 S 8 S 13			NP 10	
2			PP 12 VP 16		
3		NP 18 S 21 VP 18			
4	NP 24 S 22 S 27 NP 24 S 27 S 22 S 27				

These only give us worse options

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

Time$_0$ flies$_1$ like$_2$ an$_3$ arrow$_4$

n (constituent start index)

m (constituent length -1)

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

If we're only interested in the best parse, we can just keep best entry for each cell

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$_0$ flies$_1$ like$_2$ an$_3$ arrow$_4$

n (constituent start index)

m (constituent length -1)

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

This is the Viterbi recurrence: choose the entry with the minimum cost

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

From weights to probabilities

- To move to a probabilistic framework, we can associate probabilities with rules instead of weights

$$P(X \to Y\ Z) \stackrel{\text{def}}{=} P([_\alpha Y\ Z] \mid \alpha = X)$$

$$\therefore \forall X \sum_{RHS} P(X \to RHS) = 1$$

- The probability of a tree is just the product of the probabilities of all of the independent rule choices made, which is the product of the rule probabilities

How to apply CYK algorithm?

- Can we apply the CYK algorithm using summed weights to probabilities?
- Sure – just set the weight of a rule $X \rightarrow Y\ Z$ to $-\log P(X \rightarrow Y\ Z)$
- Now we can work with the minimum weight sum again instead of the maximum product of probabilities
- We can get $P(X \rightarrow Y\ Z)$ as $2^{-weight(X \rightarrow Y\ Z)}$

$$\boxed{P(VP \rightarrow VP\ PP) = 2^{-2} = \frac{1}{4}}$$

$$\boxed{P(PP \rightarrow P\ NP) = 2^{-0} = 1}$$

```
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_0 flies_1 like_2 an_3 arrow_4

n (constituent start index)

m (constituent length -1)

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

$$P(S \rightarrow NP\ VP) = \frac{1}{2}$$

$$P(S \rightarrow Vst\ NP) = \frac{1}{64}$$

$$P(S \rightarrow NP\ VP) = \frac{1}{4}$$

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

Time$_0$ flies$_1$ like$_2$ an$_3$ arrow$_4$

n (constituent start index)

m (constituent length -1)

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

$$P(VP \rightarrow V\ NP) = \frac{1}{2}$$

$$P(VP \rightarrow VP\ PP) = \frac{1}{4}$$

```
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$_0$ flies$_1$ like$_2$ an$_3$ arrow$_4$

n (constituent start index)

m (constituent length -1)

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

$$P(NP \rightarrow Det\ N) = \frac{1}{2}$$

$$P(NP \rightarrow NP\ PP) = \frac{1}{4}$$

$$P(NP \rightarrow NP\ NP) = \frac{1}{8}$$

```
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$_0$ flies$_1$ like$_2$ an$_3$ arrow$_4$

n (constituent start index)

m (constituent length -1)

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

$$\boxed{P(PP \rightarrow P\ NP) = 1}$$

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

Time$_0$ flies$_1$ like$_2$ an$_3$ arrow$_4$

n (constituent start index)

m (constituent length -1)

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

$$\boxed{2^{-3}\times 2^{-18}\times 2^{-1} = 2^{-22}}$$

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

Another question

- From before:
  - Under a PCFG the probability of a tree is just the product of the probabilities of all of the independent rule choices made, which is the product of the rule probabilities
  - This gives us $P(T, W)$
- What if we want $P(W)$—the probability of the word sequence under the model?
  - Sum over all possible trees

$$P(W) = \sum_{T} P(T, W)$$

  - How to compute this efficiently?

Time$_0$ flies$_1$ like$_2$ an$_3$ arrow$_4$

n (constituent start index)

m (constituent length -1)

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

$$2^{-8} + 2^{-13} \approx 2^{-8}$$

Same procedure as before, but instead of choosing the option with the highest probability, we *add* probabilities

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	

Time$_0$ flies$_1$ like$_2$ an$_3$ arrow$_4$

n (constituent start index)

m (constituent length -1)

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

Same procedure as before, but instead of choosing the option with the highest probability, we *add* probabilities

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

Time_0 flies_1 like_2 an_3 arrow_4

n (constituent start index)

m (constituent length -1)

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

$$2^{-22} + 2^{-27} \approx 2^{-22}$$

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$_0$ flies$_1$ like$_2$ an$_3$ arrow$_4$

n (constituent start index)

m (constituent length -1)

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

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

Inside probability

- The inside probability of a phrase is analogous to the forward probability of a string in HMMs
- It is the probability of a node with a given label and span being rewritten to generate the words associated with that span

$$
\begin{aligned}
P_{inside}(X, i, j) &= P(w_i \cdots w_{i+j} | X) \\
&= P(X \rightarrow^* w_i \cdots w_{i+j})
\end{aligned}
$$

LEARNING PCFGS

Probabilistic model for PCFGs

- As we've described them, PCFGs define a joint probability distribution over syntactic tree structures and the words in a sentence that we actually observe: $P_{PCFG}(T, W)$
- Is a PCFG, then, a generative or a discriminative model?

Probabilistic model for PCFGs

- PCFGs are **generative** models
  - They express a joint distribution over the observed words and the latent structure
  - They are based on a generative story about the data production process
    - Start with an S
    - Choose rules from the grammar to successively rewrite the symbol sequence, according to the probabilities associated with each rule
  - Analogous to HMMs, a generative model based on a sequential generative story

Supervised PCFG learning

- Only parameters we need to learn are probabilities associated with rules

$$P(X \rightarrow RHS)$$

nonterminal terminal

$$P(X \rightarrow Y\ Z) \qquad P(X \rightarrow w)$$

- We can estimate these using maximum-likelihood estimation:

$$P_{MLE}(X \rightarrow RHS) = \frac{Count(X \rightarrow RHS)}{Count(X)}$$

Supervised PCFG learning

$$P_{MLE}(X \rightarrow RHS) = \frac{Count(X \rightarrow RHS)}{Count(X)}$$

- To calculate these counts we need a treebank—a corpus of text with gold-standard tree structures associated with each sentence
- It's a good idea to do some count smoothing, too

```
( (S
    (NP (NP (NNP Pierre) (NNP Vinken) )
        (, ,)
        (ADJP (NP (CD 61) (NNS years) ) (JJ old) )
        (, ,) )
    (VP (MD will)
        (VP
           (VB join)
           (NP (DT the) (NN board) )
           (PP (IN as) (NP (DT a) (JJ nonexecutive) (NN director) ))
           (NP (NNP Nov.) (CD 29) )))
     (. .) ))
```

First sentence from Penn Treebank

PCFGS AND LOCALITY ASSUMPTIONS

Locality assumptions in PCFGs

- In a PCFG model, each subtree rule is associated with a probability, and the probability of the entire tree is the product of each subtree's probabilities

$$P\left(\begin{array}{ccc} & \text{S} & \\ & / \quad \backslash & \\ \begin{array}{c}\text{NP}\\ \text{it}\end{array} & & \text{VP} \\ & / \quad \backslash & \\ \begin{array}{c}\text{VP}\\ \text{was}\end{array} & & \begin{array}{c}\text{A}\\ \text{hot}\end{array} \end{array}\right) = P\left(\begin{array}{ccc} & \text{S} & \\ & / \quad \backslash & \\ \text{NP} & & \text{VP} \end{array}\right) P\left(\begin{array}{ccc} & \text{VP} & \\ & / \quad \backslash & \\ \text{VP} & & \text{A} \end{array}\right) P\left(\begin{array}{c}\text{NP}\\ \text{it}\end{array}\right) P\left(\begin{array}{c}\text{V}\\ \text{was}\end{array}\right) P\left(\begin{array}{c}\text{A}\\ \text{hot}\end{array}\right)$$

- Spot the independence assumption!

Locality assumptions in PCFGs

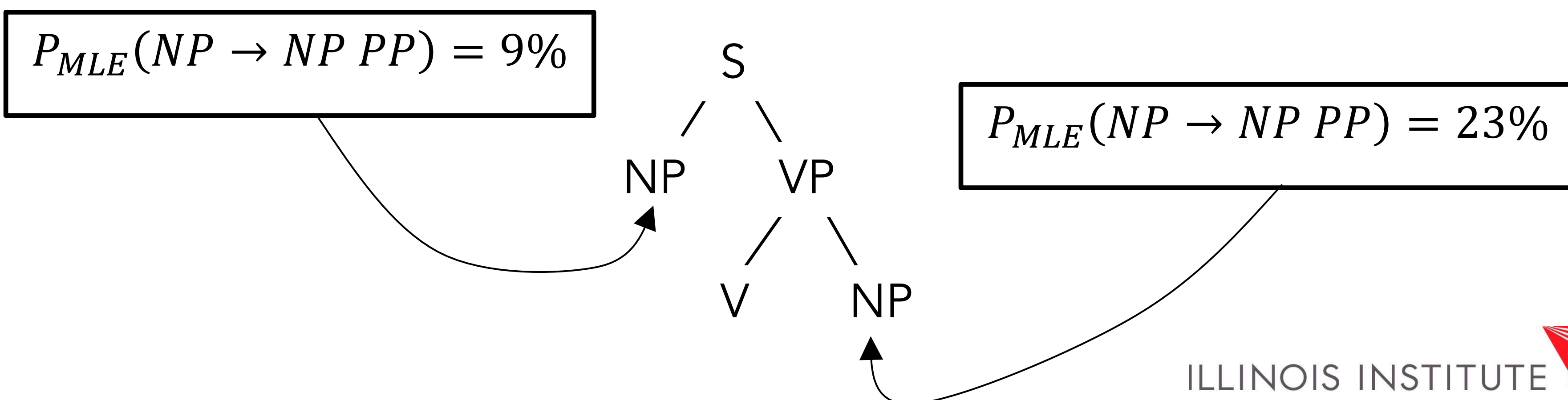
- HMMs make a Markov assumption, that we only need information about the last *n* tags/labels to determine the likelihood of the next label
- PCFGs make a similar locality assumption, but in this case it is about the independence of rules on their context in the syntactic tree
- Is this a reasonable assumption?

Locality assumptions in PCFGs

- How likely is a noun phrase to be modified by a prepositional phrase?

$$P_{MLE}(NP \rightarrow NP\ PP) = 11\%$$

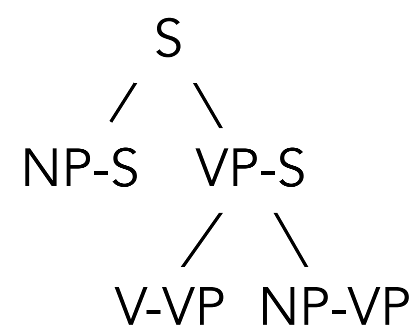
- But it varies with the NP's grammatical function



Parent annotation

- One simple approach to incorporating contextual information into PCFGs is *parent annotation*: augmenting each nonterminal with the identity of its parent node
- Similar to moving from a first-order to second-order Markov assumption in HMMs

Rule	P(Rule)
NP-S → NP-NP PP-NP	0.09
NP-VP → NP-NP PP-NP	0.23
…	…



Lexicalization

- Another approach to incorporating more contextual information into PCFGs is *lexicalization*
- Note that the likelihood of specific rules applying often depends on specific words (especially, subcategorization)
- The idea of lexicalization is to use properties of phrasal head words to get better estimates of rule probabilities

Rule	P(Rule)
VP $\rightarrow$ V NP	?
VP $\rightarrow$ V NP NP	?
AP $\rightarrow$ Adj	?
AP $\rightarrow$ Adj PP	?

see
give
...

exciting
proud
...

Head identification for lexicalization

- In a previous lecture we learned about phrasal heads:
- The head of a phrase is the a word that determines its attributes
  - Typically of the same category as the phrase: the head of a noun phrase is a noun, the head of a prepositional phrase is a preposition, etc.
  - Attributes of the head (e.g., tense in the case of verbs, number and case in the case of nouns) are shared by the phrase as a whole
- In NLP, phrasal heads are determined using a set of "head percolation rules"

Head identification for lexicalization

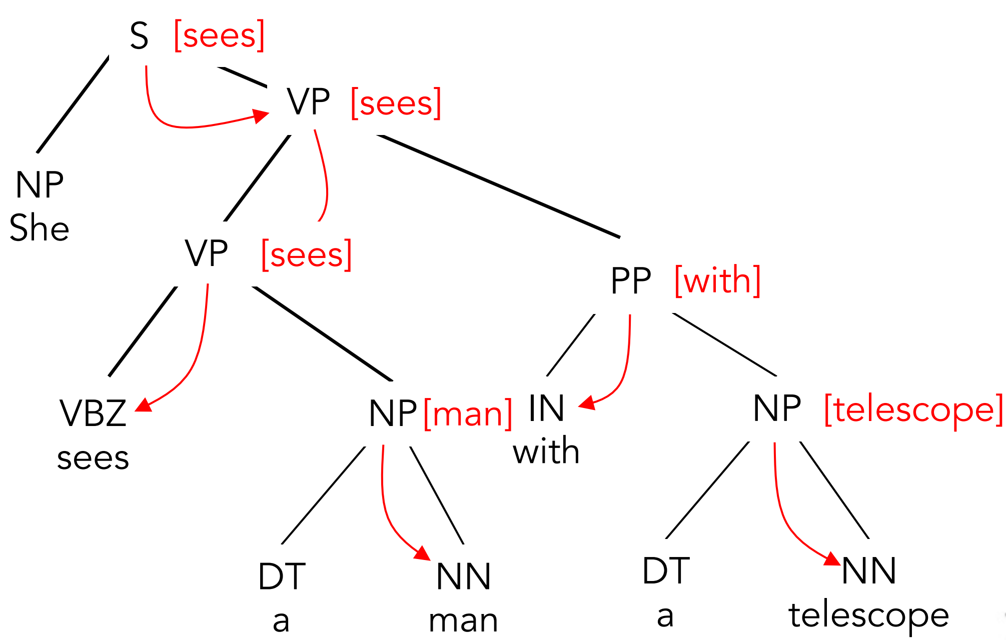
Nonterminal	Direction	Priority
S	right	VP SBAR ADJP UCP NP
VP	left	VBD VBN MD VBZ TO VB VP VBG VBP ADJP NP
NP	right	N* EX $ CD QP PRP …
PP	left	IN TO FW

Direction of search

Head categories in order of preference

Head identification for lexicalization

Nonterminal	Direction	Priority
S	right	VP SBAR ADJP UCP NP
VP	left	VBD VBN MD VBZ TO VB VP VBG VBP ADJP NP
NP	right	N* EX $ CD QP PRP …
PP	left	IN TO FW



Learning lexicalized PCFGs

- Essentially, instead of rules like

$$\texttt{VP} \longrightarrow \texttt{V NP NP}$$

- …we have rules like

$$\texttt{VP}_{\texttt{give}} \longrightarrow \texttt{V}_{\texttt{give}}\ \texttt{NP}_{\texttt{ball}}\ \texttt{PP}_{\texttt{to}}$$

- What about

$$\texttt{VP}_{\texttt{donate}} \longrightarrow \texttt{V}_{\texttt{donate}}\ \texttt{NP}_{\texttt{plasma}}\ \texttt{PP}_{\texttt{to}}$$

?

Sparse data!
We'd need a lot of data to estimate the necessary counts

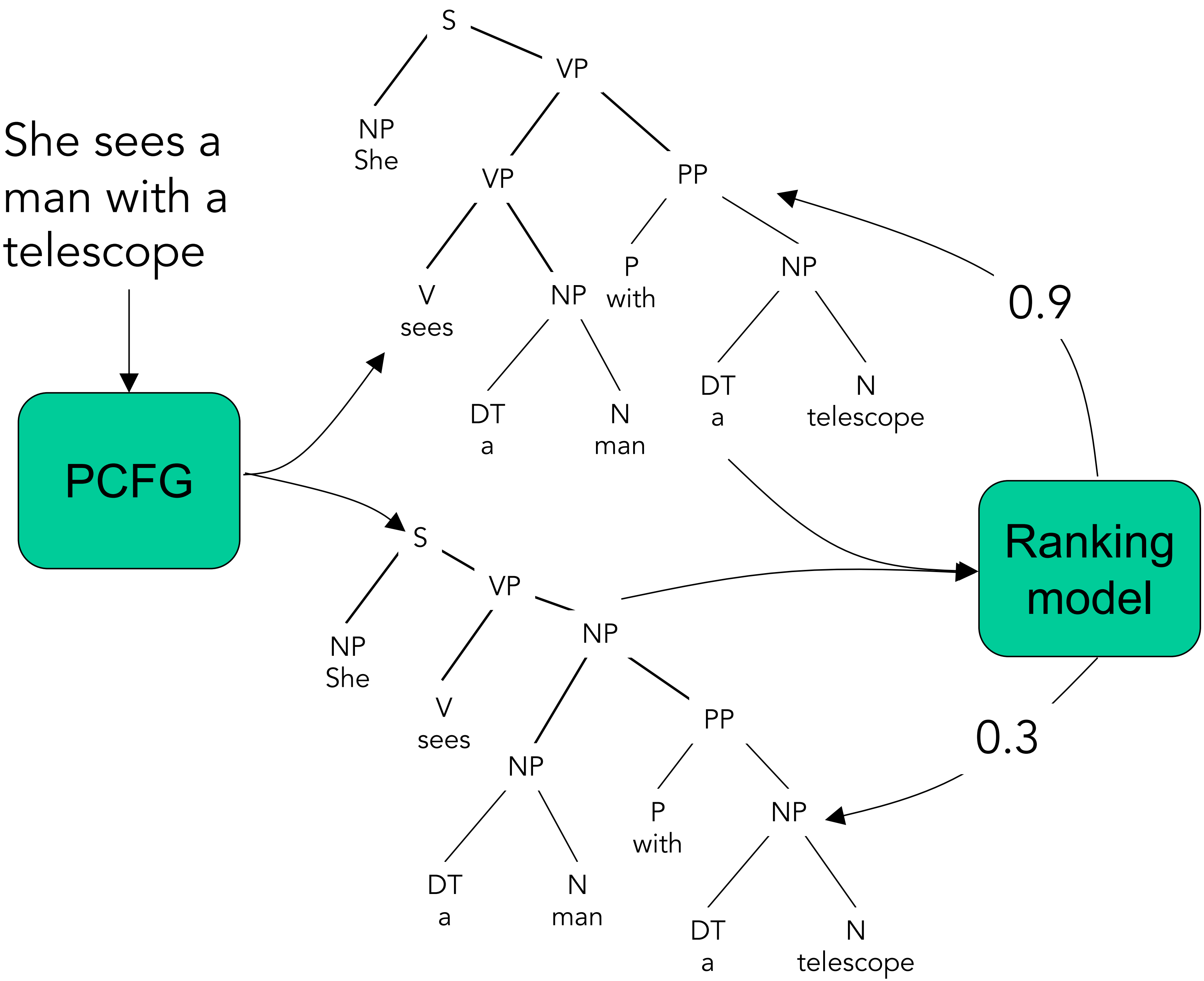
Learning lexicalized PCFGs

- How to deal with sparse data for lexicalized rules
  - Count smoothing
  - Backoff
- Backoff for PCFGs:

$$
\begin{aligned}
&P(VP_{give} \rightarrow V_{give}\ NP_{ball}\ PP_{to}) = f(\\
&\qquad P_{MLE}(VP_{give} \rightarrow V_{give}\ NP_{ball}\ PP_{to}), \\
&\qquad P_{MLE}(VP_{give} \rightarrow V_{give}\ NP\ PP), \\
&\qquad P_{MLE}(VP \rightarrow V\ NP\ PP) \\
&)
\end{aligned}
$$

Reranking

- Another way to work around the locality assumptions of PCFGs is to build a parse *reranking* model
- A standard PCFG is used to generate candidate parses for a sentence
  - N best parses from Viterbi algorithm
- Then a discriminative classifier is trained to predict whether each candidate is correct or incorrect
  - Scores from this classifier are used to rank candidates and select the best one
  - The classifier can use features from the entire parse, even combining information from structurally distant constituents


She sees a man with a telescope
PCFG
S
NP
She
VP
VP
V
sees
NP
DT
a
N
man
PP
P
with
NP
DT
a
N
telescope
S
NP
She
VP
V
sees
NP
NP
DT
a
N
man
PP
P
with
NP
DT
a
N
telescope
Ranking model
0.9
0.3

DISCRIMINATIVE PARSING MODELS

Discriminative parsing

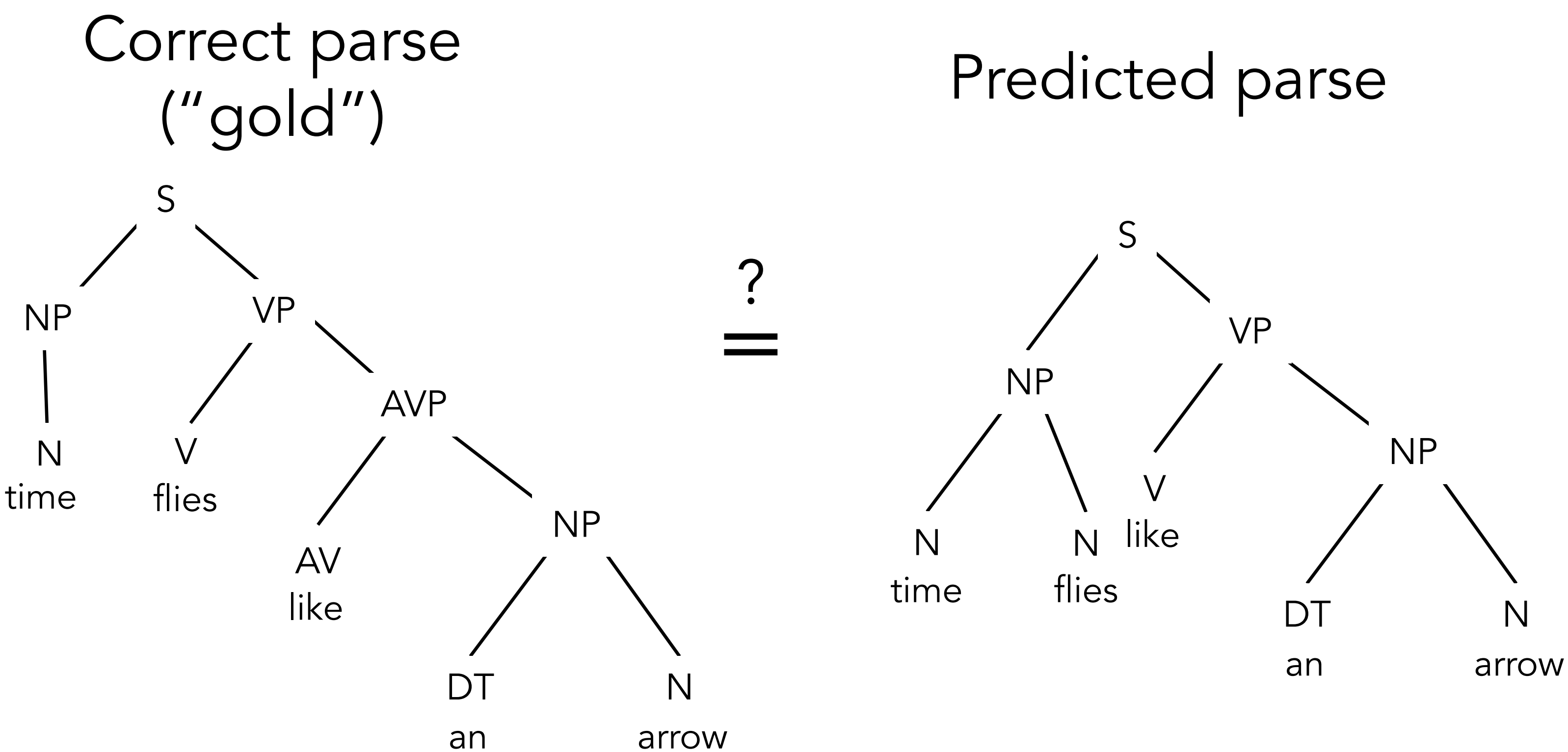
- In sequence modeling, the MEMMs and CRFs are the discriminative counterparts to generative HMMs
  - HMMs estimate the joint distribution over tags and words $P(T, W)$
  - MEMMs and CRFs estimate the conditional distribution of tags given words $P(T|W)$
- Similarly, parsing models may also be generative or discriminative
  - PCFGs are generative models that estimate the joint distribution of trees and words $P(T, W)$
  - Discriminative models estimate the conditional distribution of tree structures given the words observed $P(T|W)$

Discriminative parsing

- Discriminative models assign a score to each subtree/rule application $\phi(X \rightarrow Y\ Z)$
  - Scores based on a model using lexical features, features based on internal structure of constituent, combinations of these
  - May be probabilities (locally normalized) or just scores (globally normalized for entire tree)
  - Common frameworks are CRFs and neural network models
- Dynamic programming algorithms for inference and probability estimate (Viterbi, inside algorithm) apply similarly to PCFGs
- Many discriminative parsing models based on transition-based framework (shift-reduce) rather than chart parsing

PARSER EVALUATION

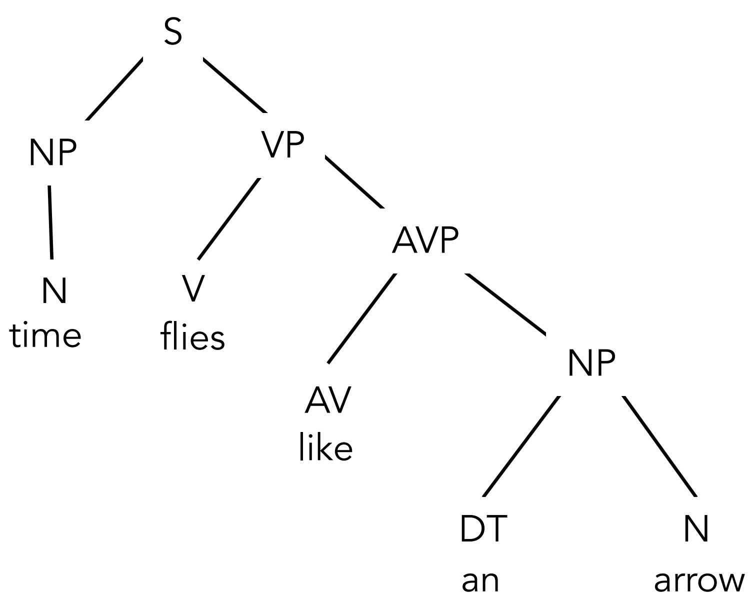
Comparing parse trees



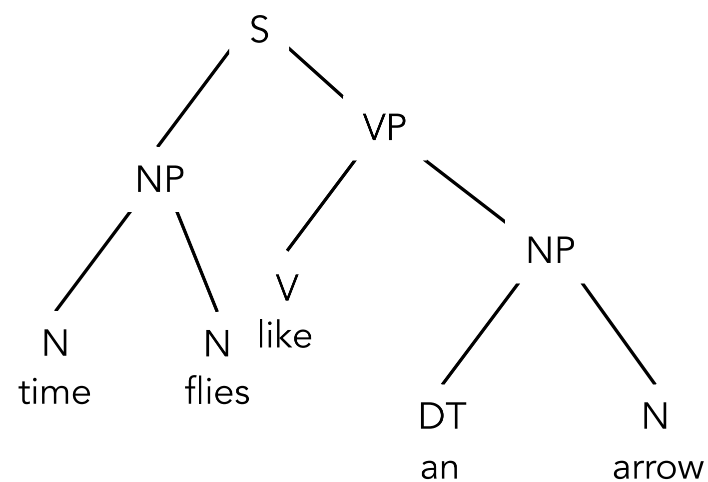
Absolute accuracy: **0%**

Proportion of constituents correctly identified

Correct parse ("gold")



Predicted parse



Proportion of constituents correctly identified

Correct parse ("gold")

(S time_0 flies_1 like_2 an_3 arrow_4)

(VP flies_1 like_2 an_3 arrow_4)

(NP time_0)

(AVP like_2 an_3 arrow_4)

(NP an_3 arrow_4)

Predicted parse

(S time_0 flies_1 like_2 an_3 arrow_4)

(NP time_0 flies_1)

(VP like_2 an_3 arrow_4)

(NP an_3 arrow_4)

Proportion of constituents correctly identified

Correct parse ("gold")

(S time_0 flies_1 like_2 an_3 arrow_4)

(VP flies_1 like_2 an_3 arrow_4)

(NP time_0)

(AVP like_2 an_3 arrow_4)

(NP an_3 arrow_4)

Predicted parse

(S time_0 flies_1 like_2 an_3 arrow_4)

(NP time_0 flies_1)

(VP like_2 an_3 arrow_4)

(NP an_3 arrow_4)

$$\boxed{\textit{Labeled Precision} = \frac{TP}{TP+FP} = \frac{2}{2+2} = 50\%}$$

Proportion of constituents correctly identified

Correct parse ("gold")

Predicted parse

```
(S time₀ flies₁ like₂ an₃ arrow₄)
```

Correct parse:

(S $time_0$ $flies_1$ $like_2$ an_3 $arrow_4$)

(VP $flies_1$ $like_2$ an_3 $arrow_4$)

(NP $time_0$)

(AVP $like_2$ an_3 $arrow_4$)

(NP an_3 $arrow_4$)

Predicted parse:

(S $time_0$ $flies_1$ $like_2$ an_3 $arrow_4$)

(NP $time_0$ $flies_1$)

(VP $like_2$ an_3 $arrow_4$)

(NP an_3 $arrow_4$)

$$\mathit{Labeled\ Recall} = \frac{TP}{TP+FN} = \frac{2}{2+3} = 40\%$$

Proportion of constituents correctly identified

Correct parse ("gold")

```
(S time₀ flies₁ like₂ an₃ arrow₄)
     (VP flies₁ like₂ an₃ arrow₄)
(NP time₀)
           (AVP like₂ an₃ arrow₄)
                  (NP an₃ arrow₄)
```

Predicted parse

```
(S time₀ flies₁ like₂ an₃ arrow₄)
(NP time₀ flies₁)
            (VP like₂ an₃ arrow₄)
                  (NP an₃ arrow₄)
```

Ignore the mismatched label

$$Unlabeled\ Precision = \frac{TP}{TP+FP} = \frac{3}{3+1} = 75\%$$

Proportion of constituents correctly identified

Correct parse ("gold")

(S time$_0$ flies$_1$ like$_2$ an$_3$ arrow$_4$)

(VP flies$_1$ like$_2$ an$_3$ arrow$_4$)

(NP time$_0$)

(AVP like$_2$ an$_3$ arrow$_4$)

(NP an$_3$ arrow$_4$)

Predicted parse

(S time$_0$ flies$_1$ like$_2$ an$_3$ arrow$_4$)

(NP time$_0$ flies$_1$)

(VP like$_2$ an$_3$ arrow$_4$)

(NP an$_3$ arrow$_4$)

$$\text{Unlabeled Recall} = \frac{TP}{TP+FN} = \frac{3}{3+2} = 60\%$$

State-of-the-art syntactic parsing metrics

Sample evaluation metrics for state-of-the-art neural parsers (Labeled F1)

	Berkeley		BLLIP		In-Order		Chart	
	F1	Δ Err.	F1	Δ Err.	F1	Δ Err.	F1	Δ Err.
WSJ Test	90.06	+0.0%	91.48	+0.0%	91.47	+0.0%	93.27	+0.0%
Brown All	84.64	+54.5%	85.89	+65.6%	85.60	+68.9%	88.04	+77.7%
Genia All	79.11	+110.2%	79.63	+139.1%	80.31	+130.9%	82.68	+157.4%
EWT All	77.38	+127.6%	79.91	+135.8%	79.07	+145.4%	82.22	+164.2%

Statistical parsers

Neural parsers

Out of domain:
Mixed genre, biomedical, web

In domain:
Wall Street Journal

Fried, Daniel et al. 2019. Cross-Domain Generalization of Neural Constituency Parsers. ACL 2019.