Context-Free Parsing:
CKY and Probabilistic CFGs

CS-585

Natural Language Processing

Sonjia Waxmonsky

Based on slides from Kai Shu and Derrick Higgins

REVISITNG GRAMMARS CONCEPTS

Constituents

To study syntax, we break sentences into **constituents**, or continuous sequences of words that <u>function as a coherent unit</u>.

I ate the last piece of candy

Constituent

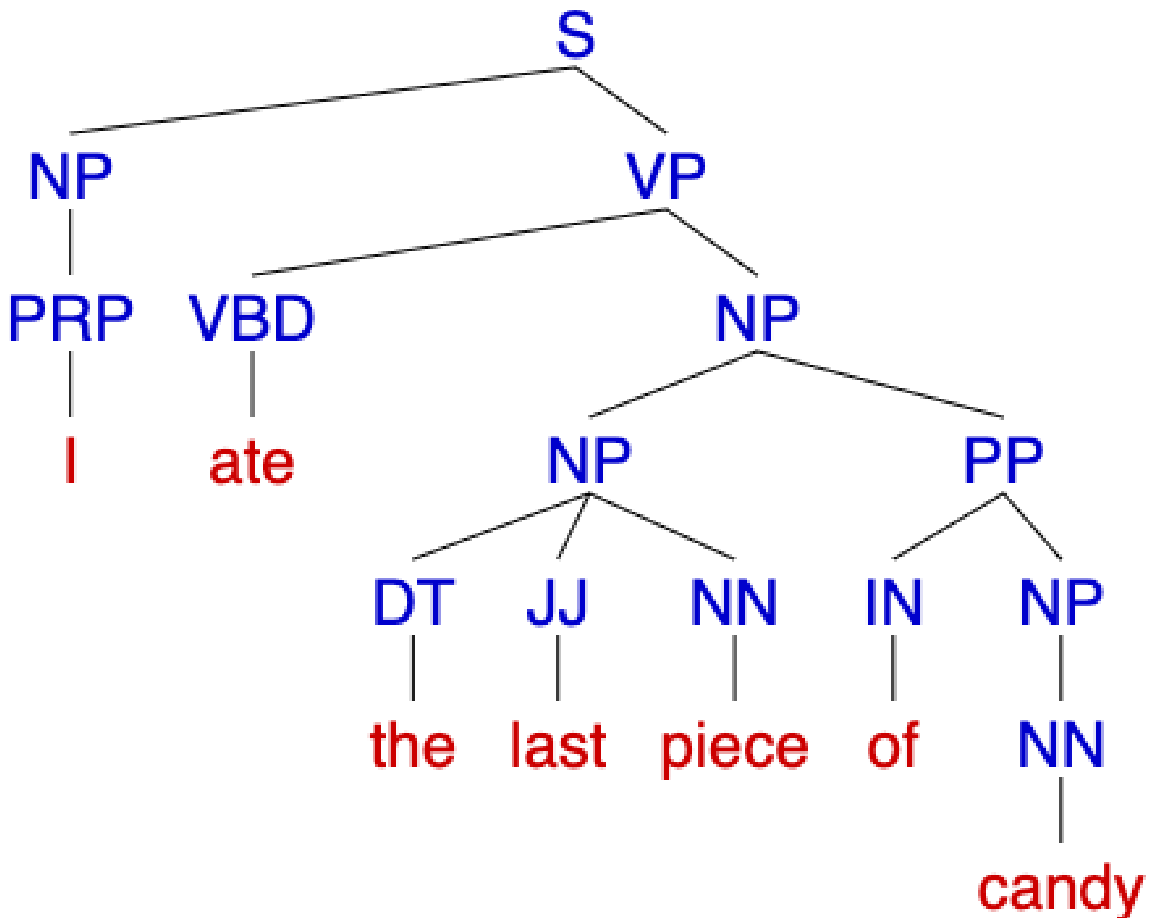
not a constituent

Constituency Tests

Constituents can be tested with these properties:

- **Movement**
  - "I ate the last piece of candy"
  - "The last piece of candy was eaten"
- **Substitution** or **Replacement**
  - "I ate the last piece of candy" vs "I ate it".
- **Coordination** or **Conjunction**
  - "I ate the last piece of candy and an apple".
- **Fragment Test**
  - Q: "What did you eat?" A: "The last piece of candy"

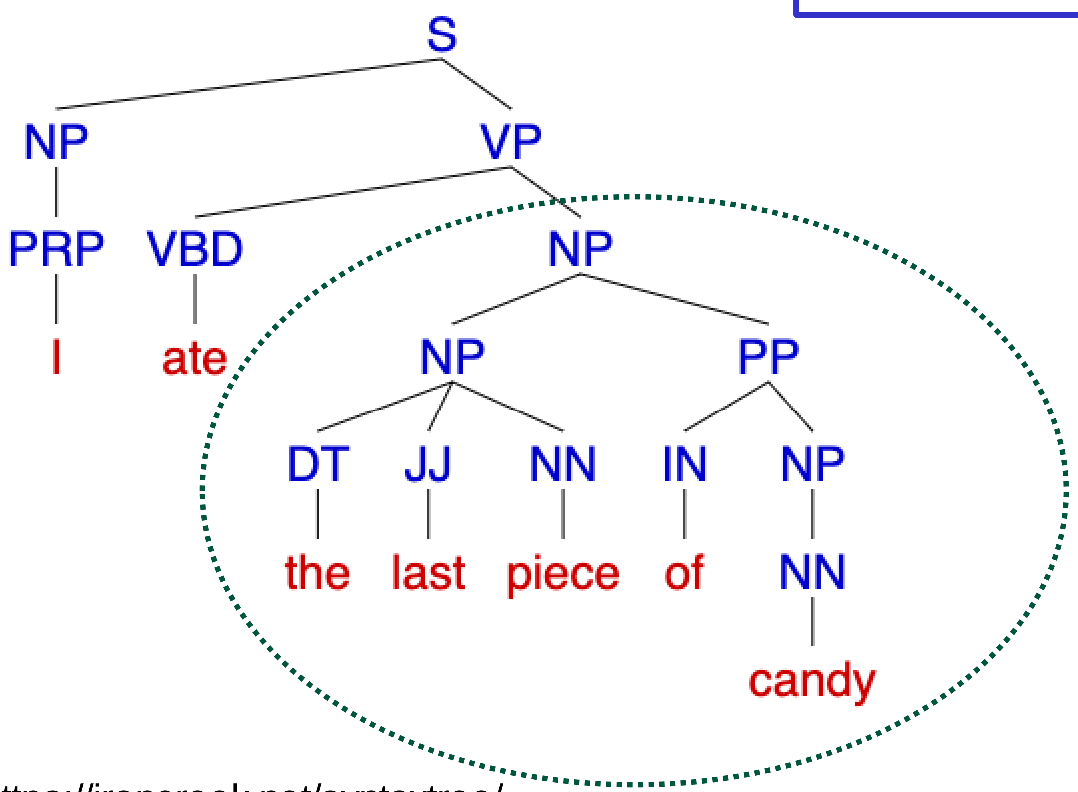
Phrase Structure Trees



https://ironcreek.net/syntaxtree/

Phrase Structure Trees

I ate the last piece of candy



https://ironcreek.net/syntaxtree/

PARSING WITH CONTEXT-FREE GRAMMARS

Natural Language Parsing

- Goals of Parsing:
  - Recognize if a sentence is valid: Can it be derived from a given grammar?
  - Determine the syntactic structure of the sentence – useful for downstream NLP tasks

- Problem: Some sentences are **ambiguous**, can be described by **multiple parse trees** licensed by a given grammar.
- Is it possible to parse a sentence deterministically as it is being read (e.g. from left-to-right)?

Context-free Grammars

Slide from Session 23

- **Production rules** of form

$$X \rightarrow y\ Z$$

$$X \rightarrow y$$

$$X \rightarrow Y\ Z$$

Terminal symbols [x, y, z,…]

Nonterminal symbols [X, Y, Z, …]

- For example:

S → NP VP

NP → DT N

DT → the

N → dog

…

Context-Free Grammar

- Start symbol S
- Set of non-terminal symbols {NP, VP, …}
- Set of terminal symbols (words)
- Set of production rules, of the form

```
NT → a b c …
```

where **NT** is a non-terminal and **a b c** comprise a sequence of 1 or more terminals and non-terminals

Example

S	→	NP VP
NP	→	Name
NP	→	N
NP	→	NP PP
VP	→	V NP
VP	→	V
VP	→	VP PP
PP	→	P NP

Name	→	joe
N	→	ice
N	→	drinks
N	→	water
V	→	drinks
P	→	with

"joe drinks water with ice"

Parser Properties

Soundness: A parser is sound if every parse returned is valid in the grammar.

Completeness: A parser is complete if for every grammar and sentence it returns all valid parses for that sentence.

- Soundness is key...
- ...but completeness may be difficult or even undesirable, e.g. for highly ambiguous grammars...

Bottom-Up Parsing

- Goal list initialized as list of terminals in the string to be parsed
- If sequence of goals matches RHS of a rule, replace it with the LHS of the rule
- Parsing complete when producing S
- Choices:
  1. RHS of multiple rules may match
  2. Order of subgoals (depth-first, breadth-first)

Inefficient when grammar has lexical ambiguity

Chart Parsing

- Remember intermediate results
- Explore all possible solutions in parallel

Sentence: $w_1\ w_2\ w_3\ w_4\ \ldots$

Chart: Array whose entries show the set of categories that could generate words from n to n+m

Formally:

$$chart(m,n) = \{ A \mid A \rightarrow* \ w_n \ldots w_{n+m} \}$$

Example

The$_0$ man$_1$ drinks$_2$ water$_3$ with$_4$ ice$_5$

n (constituent start index)

m (constituent length -1)

	0	1	2	3	4	5
0	Det	N,NP	V,VP,NP	N,NP	P	N,NP
1	NP	S	VP	{ }	PP	
2	S	S	{ }	NP		
3	S	{ }	VP			
4	{ }	S				
5	S					

Each cell stores a partial solution for words from n to n+m

Chomsky Normal Form

- Constraint on form of the grammar:
  - Each RHS is either 2 non-terminals or a terminal
- All CFGs can be written in CNF

```
S    → NP VP
NP   → NP PP
NP   → Det NP
VP   → V NP
VP   → VP PP
PP   → P NP
```

```
Det  → the
NP   → joe
NP   → ice
NP   → drinks
NP   → water
V    → drinks
VP   → drinks
P    → with
```

Cocke-Kasami-Younger (CKY)

Assume "Chomsky Normal Form" grammar

```
for n := 0 to Nw-1 do:
  chart[0, n] := {X | X → wordn }

for m := 1 to Nw-1 do:
   for n := 0 to Nw-m-1 do:
        chart[m, n] := {}
        for k := n+1 to n+m do
                for every rule A → B C do
                        if B ∈ chart[k-n-1, n] and C ∈ chart[n+m-k, k] then
                                chart[m, n] := chart[m, n] ∪ {A}

if S ∈ chart[Nw-1, 0] then accept else reject
```

CKY Example (in CNF)

S	→	NP VP	NP	→	joe
NP	→	NP PP	NP	→	ice
VP	→	V NP	NP	→	drinks
VP	→	VP PP	NP	→	water
PP	→	P NP	V	→	drinks
			VP	→	drinks
			P	→	with

“joe drinks water with ice”

Cocke-Kasami-Younger (CKY)

```
for n := 0 to N_w-1 do:
  chart[0, n] := {X | X → word_n }

for m := 1 to N_w-1 do:
   for n := 0 to N_w-m-1 do:
      chart[m, n] := {}
      for k := n+1 to n+m do
         for every rule A → B C do
            if B ∈ chart[k-n-1, n] and C ∈ chart[n+m-k,
   k] then
               chart[m, n] := chart[m, n] ∪ {A}

if S ∈ chart[N_w-1, 0] then accept else reject
```

Initialize chart with terminal symbols

Example

Joe_0 $drinks_1$ $water_2$ $with_3$ ice_4

n (constituent start index)

$N_w = 5$

m (constituent length -1)	0	1	2	3	4
0	NP	V,VP,NP	NP	P	NP
1					
2					
3					
4					

Cocke-Kasami-Younger (CKY)

Look for increasingly longer phrases

```
for n := 0 to N_w-1 do:
    chart[0, n] := {X | X → word_n}
```

Start from the left-hand edge and stop if the constituent would run past the end of the sentence

```
for m := 1 to N_w-1 do:
    for n := 0 to N_w-m-1 do:
        chart[m, n] := {}
        for k := n+1 to n+m do
            for every rule A → B C do
                if B ∈ chart[k-n-1, n] and C ∈ chart[n+m-k, k] then
                    chart[m, n] := chart[m, n] ∪ {A}
```

Consider all ways you could divide the text span into two parts, and look for a rule that matches

```
if S ∈ chart[N_w-1, 0] then accept else reject
```

Example

Joe$_0$ drinks$_1$ water$_2$ with$_3$ ice$_4$

n (constituent start index)

m (constituent length -1)	0	1	2	3	4
0	NP	V,VP,NP	NP	P	NP
1	S				
2					
3					
4					

$N_w = 5$

$k = n+1$

S → NP VP

Example

Joe$_0$ drinks$_1$ water$_2$ with$_3$ ice$_4$

n (constituent start index)

m (constituent length -1)	0	1	2	3	4
0	NP	V,VP,NP	NP	P	NP
1	S	VP			
2					
3					
4					

$N_w = 5$

$k = n+1$

VP → V NP

Example

Joe$_0$ drinks$_1$ water$_2$ with$_3$ ice$_4$

n (constituent start index)

m (constituent length -1)	0	1	2	3	4
0	NP	V,VP,NP	NP	P	NP
1	S	VP	{ }		
2					
3					
4					

$N_w = 5$

$k = n+1$

$\boxed{\emptyset}$

Example

Joe_0 drinks_1 water_2 with_3 ice_4

n (constituent start index)

m (constituent length -1)	0	1	2	3	4
0	NP	V,VP,NP	NP	P	NP
1	S	VP	{ }	PP	
2					
3					
4					

$N_w = 5$

$k = n+1$

PP → P NP

Example

Joe$_0$ drinks$_1$ water$_2$ with$_3$ ice$_4$

n (constituent start index)

m (constituent length -1)	0	1	2	3	4
0	NP	V,VP,NP	NP	P	NP
1	S	VP	{ }	PP	
2	S				
3					
4					

$N_w = 5$

$k = n+1$

S → NP VP

Example

Joe_0 $drinks_1$ $water_2$ $with_3$ ice_4

n (constituent start index)

m (constituent length -1)

	0	1	2	3	4
0	NP	V,VP,NP	NP	P	NP
1	S	VP	{ }	PP	
2	S				
3					
4					

$N_w = 5$

$k = n+2$

$\boxed{\emptyset}$

Example

Joe_0 drinks_1 water_2 with_3 ice_4

n (constituent start index)

m (constituent length -1)	0	1	2	3	4
0	NP	V,VP,NP	NP	P	NP
1	S	VP	{ }	PP	
2	S	{ }			
3					
4					

$N_w = 5$

$k = n+1$

$\boxed{\emptyset}$

Example

Joe_0 $drinks_1$ $water_2$ $with_3$ ice_4

n (constituent start index)

m (constituent length -1)

	0	1	2	3	4
0	NP	V,VP,NP	NP	P	NP
1	S	VP	{ }	PP	
2	S	{ }			
3					
4					

$N_w = 5$

$k = n+2$

$\boxed{\emptyset}$

Example

Joe$_0$ drinks$_1$ water$_2$ with$_3$ ice$_4$

n (constituent start index)

m (constituent length -1)	0	1	2	3	4
0	NP	V,VP,NP	NP	P	NP
1	S	VP	{ }	PP	
2	S	{ }	NP		
3					
4					

$N_w = 5$

$k = n+1$

$\boxed{NP \rightarrow NP \ PP}$

Example

Joe$_0$ drinks$_1$ water$_2$ with$_3$ ice$_4$

n (constituent start index)

m (constituent length -1)	0	1	2	3	4
0	NP	V,VP,NP	NP	P	NP
1	S	VP	{ }	PP	
2	S	{ }	NP		
3					
4					

$N_w = 5$

$k = n+2$

$\boxed{\emptyset}$

Example

$Joe_0 \quad drinks_1 \quad water_2 \quad with_3 \quad ice_4$

n (constituent start index)

m (constituent length -1)

	0	1	2	3	4
0	NP	V,VP,NP	NP	P	NP
1	S	VP	{ }	PP	
2	S	{ }	NP		
3	{ }				
4					

$N_w = 5$

$k = n+1$

$\boxed{\emptyset}$

Example

Joe$_0$ drinks$_1$ water$_2$ with$_3$ ice$_4$

n (constituent start index)

m (constituent length -1)	0	1	2	3	4
0	NP	V,VP,NP	NP	P	NP
1	S	VP	{ }	PP	
2	S	{ }	NP		
3	{ }				
4					

$N_w = 5$

$k = n+2$

$\boxed{\emptyset}$

Example

Joe_0 $drinks_1$ $water_2$ $with_3$ ice_4

n (constituent start index)

m (constituent length -1)

	0	1	2	3	4
0	NP	V,VP,NP	NP	P	NP
1	S	VP	{ }	PP	
2	S	{ }	NP		
3	{ }				
4					

$N_w = 5$

$k = n+3$

$\boxed{\emptyset}$

Example

Joe_0 $drinks_1$ $water_2$ $with_3$ ice_4

n (constituent start index)

m (constituent length -1)

	0	1	2	3	4
0	NP	V,VP,NP	NP	P	NP
1	S	VP	{ }	PP	
2	S	{ }	NP		
3	{ }	VP			
4					

$N_w = 5$

$k = n+1$

VP → V NP

Example

Joe_0 drinks_1 water_2 with_3 ice_4

n (constituent start index)

$N_w = 5$
$k = n+2$

m (constituent length -1)	0	1	2	3	4
0	NP	V,VP,NP	NP	P	NP
1	S	VP	{ }	PP	
2	S	{ }	NP		
3	{ }	VP			
4					

VP → VP PP

Example

Joe_0 $drinks_1$ $water_2$ $with_3$ ice_4

n (constituent start index)

m (constituent length -1)	0	1	2	3	4
0	NP	V,VP,NP	NP	P	NP
1	S	VP	{ }	PP	
2	S	{ }	NP		
3	{ }	VP			
4					

$N_w = 5$

$k = n+3$

$\boxed{\emptyset}$

Example

Joe$_0$ drinks$_1$ water$_2$ with$_3$ ice$_4$

n (constituent start index)

m (constituent length -1)	0	1	2	3	4
0	NP	V,VP,NP	NP	P	NP
1	S	VP	{ }	PP	
2	S	{ }	NP		
3	{ }	VP			
4	S				

$N_w = 5$

$k = n+1$

S → NP VP

Example

Joe$_0$ drinks$_1$ water$_2$ with$_3$ ice$_4$

n (constituent start index)

m (constituent length -1)	0	1	2	3	4
0	NP	V,VP,NP	NP	P	NP
1	S	VP	{ }	PP	
2	S	{ }	NP		
3	{ }	VP			
4	S				

$N_w = 5$

$k = n+2$

$\boxed{\emptyset}$

Example

Joe$_0$ drinks$_1$ water$_2$ with$_3$ ice$_4$

n (constituent start index)

m (constituent length -1)

	0	1	2	3	4
0	NP	V,VP,NP	NP	P	NP
1	S	VP	{ }	PP	
2	S	{ }	NP		
3	{ }	VP			
4	S				

$N_w = 5$

$k = n+3$

$\boxed{\varnothing}$

Example

Joe_0 $drinks_1$ $water_2$ $with_3$ ice_4

n (constituent start index)

m (constituent length -1)

	0	1	2	3	4
0	NP	V,VP,NP	NP	P	NP
1	S	VP	{ }	PP	
2	S	{ }	NP		
3	{ }	VP			
4	S				

$N_w = 5$

$k = n+4$

$\boxed{\emptyset}$

Example

Joe$_0$ drinks$_1$ water$_2$ with$_3$ ice$_4$

n (constituent start index)

$N_w = 5$

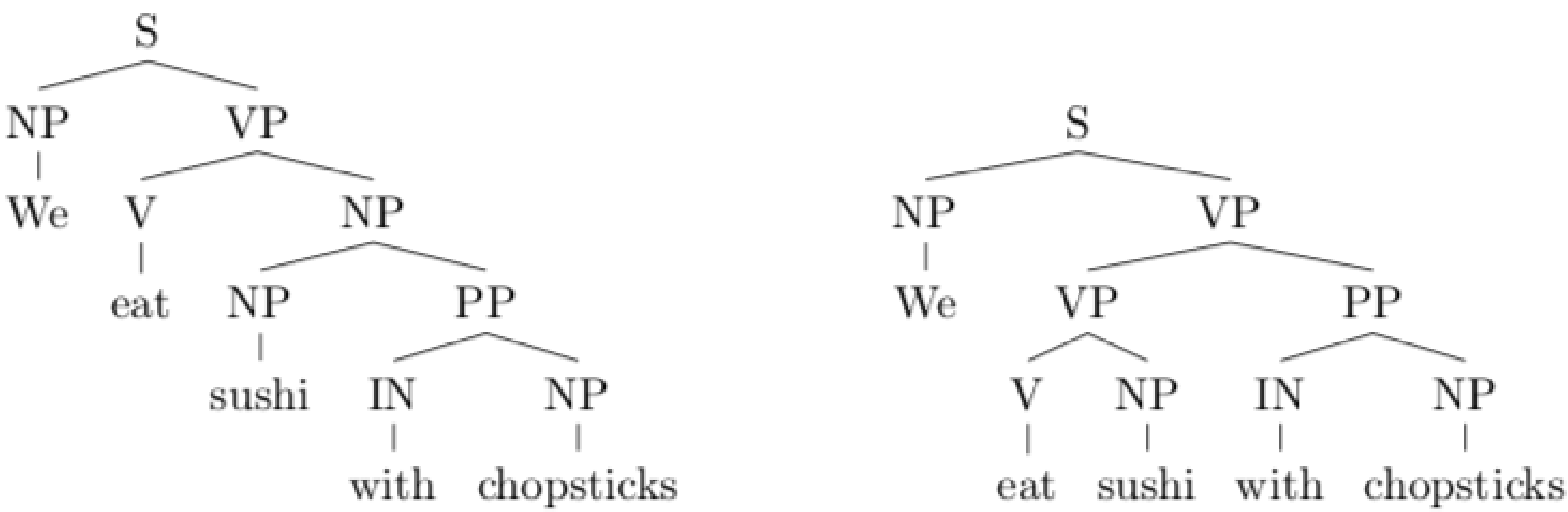
m (constituent length -1)

	0	1	2	3	4
0	NP	V,VP,NP	NP	P	NP
1	S	VP	{ }	PP	
2	S	{ }	NP		
3	{ }	VP			
4	S				

PROBABILISTIC CONTEXT-FREE GRAMMARS

Parsing and ambiguity

Ambiguity can result in more than one valid parse:



Eisenstein-NLP Figure 10.2 - Attachment ambiguity

Parsing and ambiguity

- We saw how to **generate** tree structures using the CKY 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?
  - Annotated treebank with supervised learning

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 → 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				
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 → 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					
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 → 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			
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 → 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				

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

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 → 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				

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 → 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

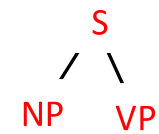
0 PP → P NP

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 → 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
```

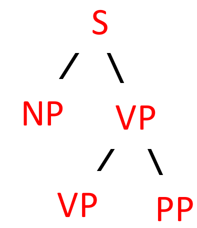
Use back-pointers to recover best parse

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 → 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

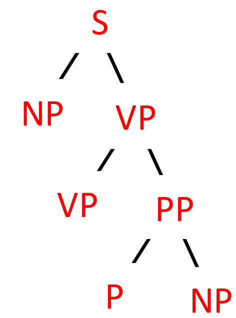
0 PP → P NP

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				



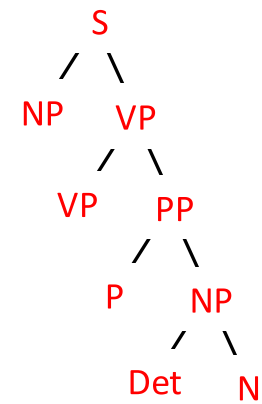

```
1   S  → NP VP
6   S  → Vst NP
2   S  → S PP

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 → 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				

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$_4$

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				

These only give us worse options

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				

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				

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

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

From weights to probabilities

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

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

$$\therefore \forall X \sum_{RHS} P(X \rightarrow RHS) = 1$$ ←Probabilities sum to 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

How to apply CKY 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 $\boxed{P(X \rightarrow Y\ Z) \text{ as } 2^{-weight(X \rightarrow Y\ Z)}}$

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

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

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				

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

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

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

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				

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

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

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 → 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				

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

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

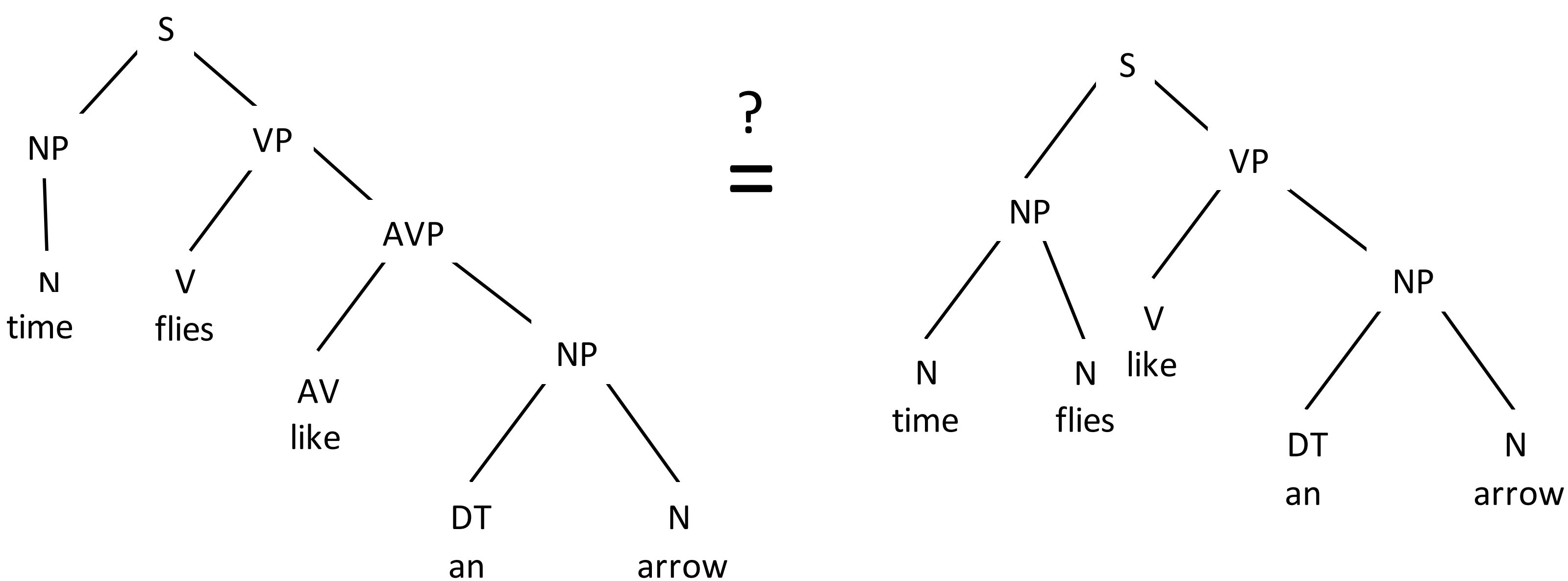
0 PP → P NP

PARSER EVALUATION

Comparing parse trees

Correct parse
("gold")

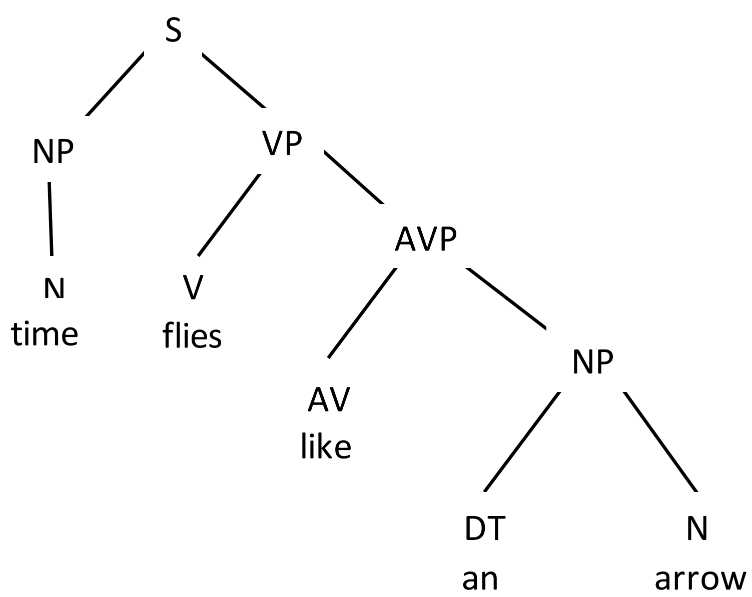
Predicted parse



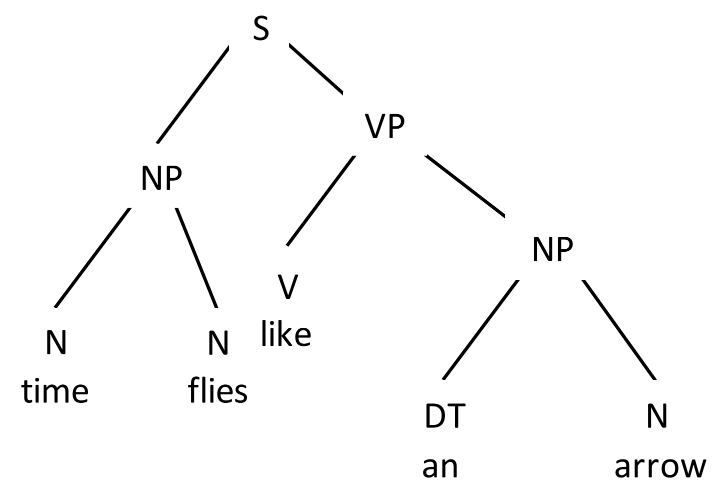
Absolute accuracy: **0%**

Proportion of constituents correctly identified

Correct parse
("gold")



Predicted parse



Proportion of constituents correctly identified

Correct parse ("gold")

$(S\ \mathtt{time}_0\ \mathtt{flies}_1\ \mathtt{like}_2\ \mathtt{an}_3\ \mathtt{arrow}_4)$

$(VP\ \mathtt{flies}_1\ \mathtt{like}_2\ \mathtt{an}_3\ \mathtt{arrow}_4)$

$(NP\ \mathtt{time}_0)$

$(AVP\ \mathtt{like}_2\ \mathtt{an}_3\ \mathtt{arrow}_4)$

$(NP\ \mathtt{an}_3\ \mathtt{arrow}_4)$

Predicted parse

$(S\ \mathtt{time}_0\ \mathtt{flies}_1\ \mathtt{like}_2\ \mathtt{an}_3\ \mathtt{arrow}_4)$

$(NP\ \mathtt{time}_0\ \mathtt{flies}_1)$

$(VP\ \mathtt{like}_2\ \mathtt{an}_3\ \mathtt{arrow}_4)$

$(NP\ \mathtt{an}_3\ \mathtt{arrow}_4)$

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")

(S $\texttt{time}_0$ $\texttt{flies}_1$ $\texttt{like}_2$ $\texttt{an}_3$ $\texttt{arrow}_4$)

(VP $\texttt{flies}_1$ $\texttt{like}_2$ $\texttt{an}_3$ $\texttt{arrow}_4$)

(NP $\texttt{time}_0$)

(AVP $\texttt{like}_2$ $\texttt{an}_3$ $\texttt{arrow}_4$)

(NP $\texttt{an}_3$ $\texttt{arrow}_4$)

Predicted parse

(S $\texttt{time}_0$ $\texttt{flies}_1$ $\texttt{like}_2$ $\texttt{an}_3$ $\texttt{arrow}_4$)

(NP $\texttt{time}_0$ $\texttt{flies}_1$)

(VP $\texttt{like}_2$ $\texttt{an}_3$ $\texttt{arrow}_4$)

(NP $\texttt{an}_3$ $\texttt{arrow}_4$)

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

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

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

Proportion of constituents correctly identified

Correct parse
("gold")

Predicted parse

Correct parse ("gold")	Predicted parse
(S time_0 flies_1 like_2 an_3 arrow_4)	(S time_0 flies_1 like_2 an_3 arrow_4)
(VP flies_1 like_2 an_3 arrow_4)	(NP time_0 flies_1)
(NP time_0)	(VP like_2 an_3 arrow_4)
(AVP like_2 an_3 arrow_4)	(NP an_3 arrow_4)
(NP an_3 arrow_4)	

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

Learning PCFGs

- Probabilistic CFGs are a form of **structured prediction** (why?)
- Learned from annotated treebanks datasets with supervised learning
  - Penn Treebank (1993) - 1st large scale Treebank
- Learning methods similar to other tasks
  - Generative models based on corpus counts and smoothing
  - Discriminative, feature-based learning
  - Neural network-based methods