

# Statistical Natural Language Parsing

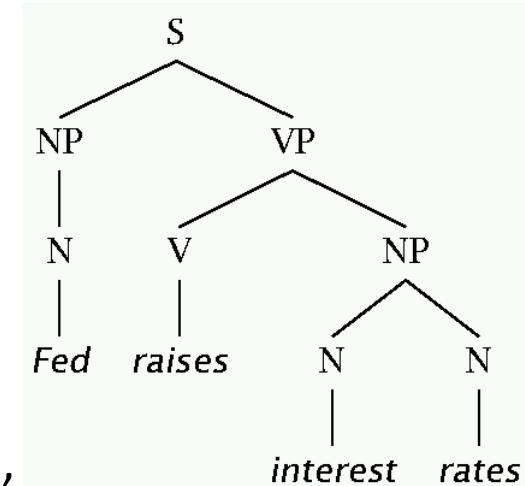
Mausam

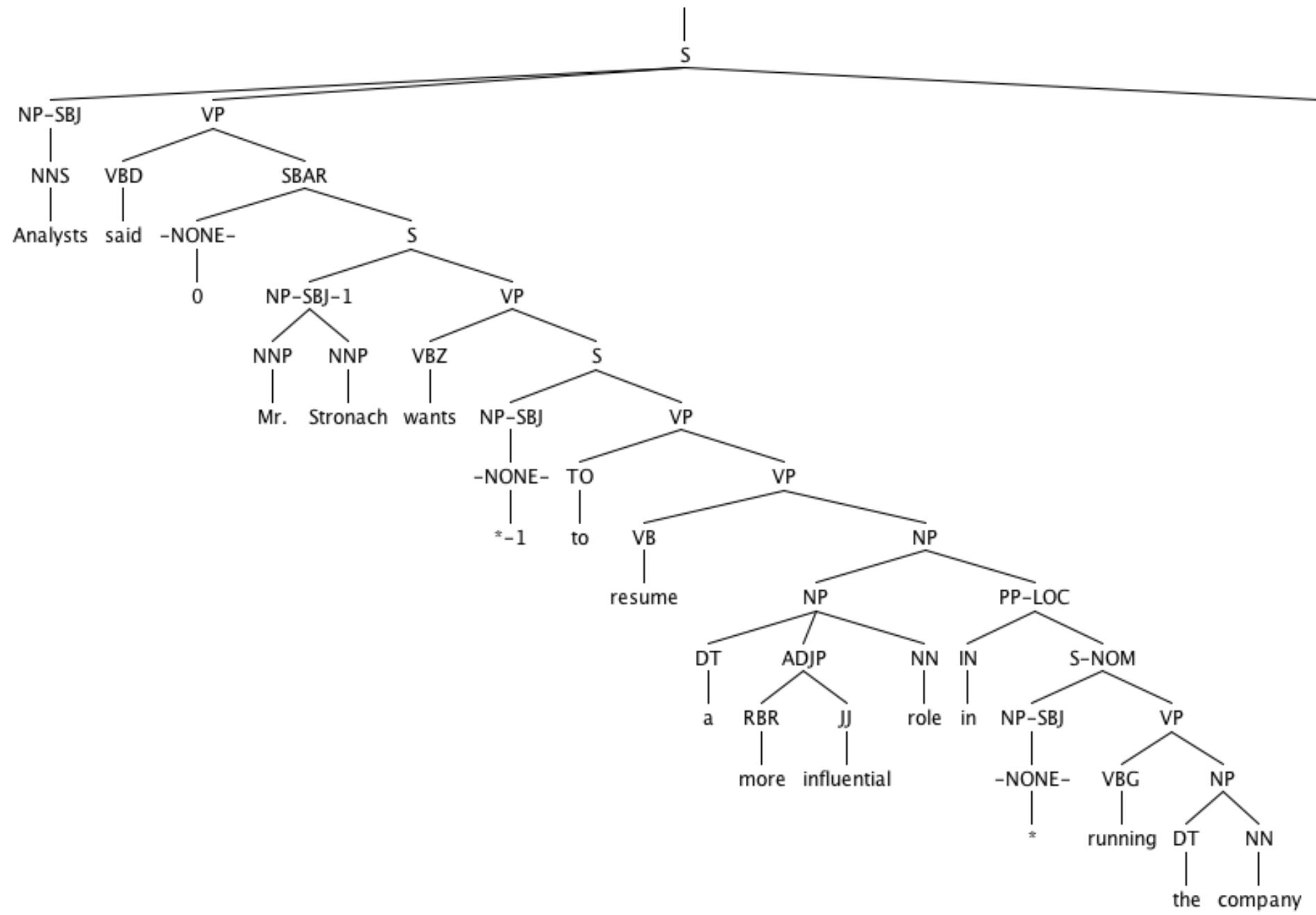
(Based on slides of Michael Collins, Dan Jurafsky, Dan Klein,  
Chris Manning, Ray Mooney, Luke Zettlemoyer)

# Two views of linguistic structure:

## 1. Constituency (phrase structure)

- Phrase structure organizes words into nested constituents.
- How do we know what is a **constituent**? (Not that linguists don't argue about some cases.)
  - Distribution: a constituent behaves as a unit that can appear in different places:
    - John talked [to the children] [about drugs].
    - John talked [about drugs] [to the children].
    - \*John talked drugs to the children about
  - Substitution/expansion/pro-forms:
    - I sat [on the box/right on top of the box/there].
  - Coordination, regular internal structure, no intrusion, fragments, semantics, ...

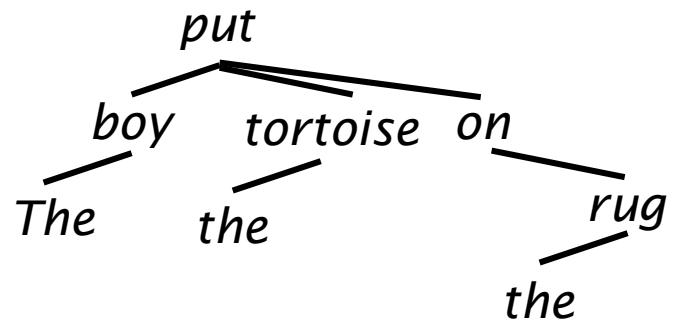
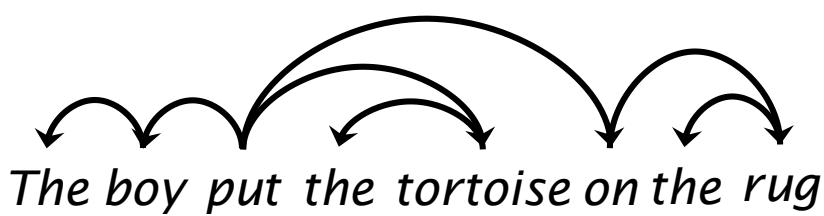




# Two views of linguistic structure:

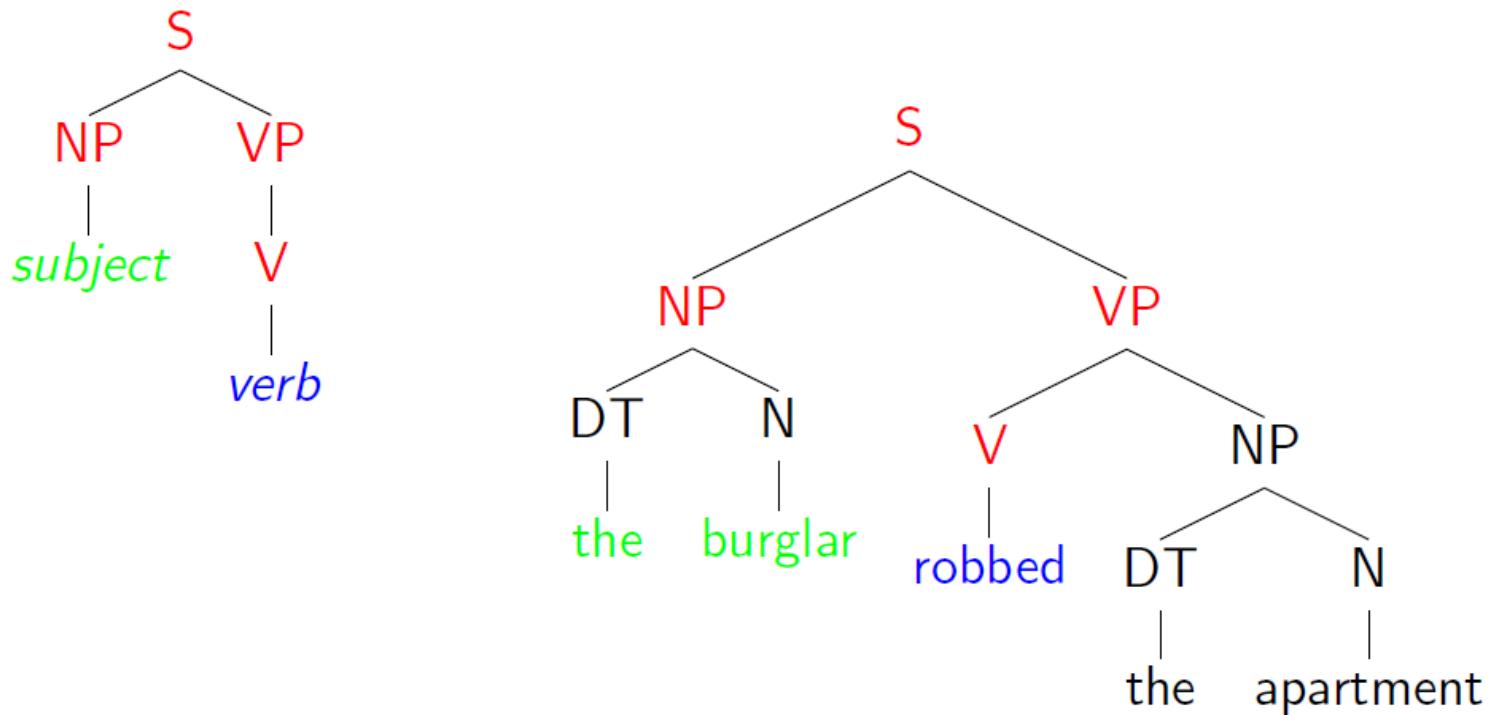
## 2. Dependency structure

- Dependency structure shows which words depend on (modify or are arguments of) which other words.



# Why Parse?

- Part of speech information
- Phrase information
- Useful relationships



# The rise of annotated data: The Penn Treebank

[Marcus et al. 1993, *Computational Linguistics*]

```
( (S
  (NP-SBJ (DT The) (NN move))
  (VP (VBD followed)
    (NP
      (NP (DT a) (NN round))
      (PP (IN of)
        (NP
          (NP (JJ similar) (NNS increases))
          (PP (IN by)
            (NP (JJ other) (NNS lenders)))
          (PP (IN against)
            (NP (NNP Arizona) (JJ real) (NN estate) (NNS loans)))))))
    (, ,)
    (S-ADV
      (NP-SBJ (-NONE- *))
      (VP (VBG reflecting)
        (NP
          (NP (DT a) (VBG continuing) (NN decline))
          (PP-LOC (IN in)
            (NP (DT that) (NN market)))))))
    (. .)))
```

# Penn Treebank Non-terminals

Table 1.2. The Penn Treebank syntactic tagset

---

|        |   |
|--------|---|
| ADJP   | Adjective phrase  |
| ADVP   | Adverb phrase   |
| NP     | Noun phrase   |
| PP     | Prepositional phrase  |
| S      | Simple declarative clause   |
| SBAR   | Subordinate clause  |
| SBARQ  | Direct question introduced by <i>wh</i> -element                          |
| SINV   | Declarative sentence with subject-aux inversion                           |
| SQ     | Yes/no questions and subconstituent of SBARQ excluding <i>wh</i> -element |
| VP     | Verb phrase   |
| WHADVP | Wh-adverb phrase  |
| WHNP   | Wh-noun phrase  |
| WHPP   | Wh-prepositional phrase   |
| X      | Constituent of unknown or uncertain category                              |
| *      | “Understood” subject of infinitive or imperative                          |
| 0      | Zero variant of <i>that</i> in subordinate clauses                        |
| T      | Trace of wh-Constituent   |

---

# The rise of annotated data

- Starting off, building a treebank seems a lot slower and less useful than building a grammar
- But a treebank gives us many things
  - Reusability of the labor
    - Many parsers, POS taggers, etc.
    - Valuable resource for linguistics
  - Broad coverage
  - Frequencies and distributional information
  - A way to evaluate systems

# Statistical parsing applications

Statistical parsers are now robust and widely used in larger NLP applications:

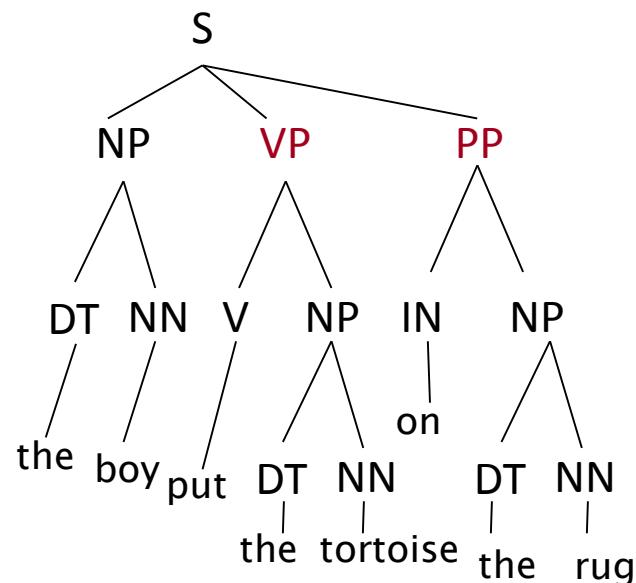
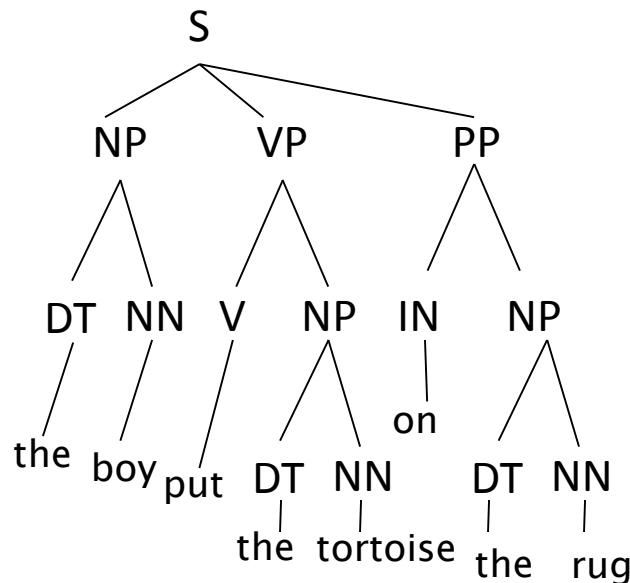
- High precision question answering [Pasca and Harabagiu SIGIR 2001]
- Improving biological named entity finding [Finkel et al. JNLPBA 2004]
- Syntactically based sentence compression [Lin and Wilbur 2007]
- Extracting opinions about products [Bloom et al. NAACL 2007]
- Improved interaction in computer games [Gorniak and Roy 2005]
- Helping linguists find data [Resnik et al. BLS 2005]
- Source sentence analysis for machine translation [Xu et al. 2009]
- Relation extraction systems [Fundel et al. *Bioinformatics* 2006]

# Example Application: Machine Translation

- The boy put the tortoise on the rug

          |      |      |      |  
लड़के ने रखा कछुआ ऊपर कालीन

- SVO vs. SOV; preposition vs. post-position

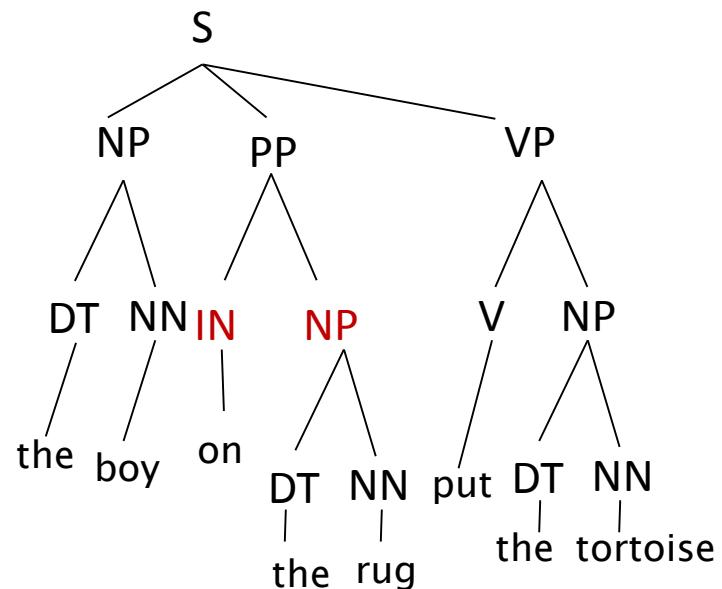
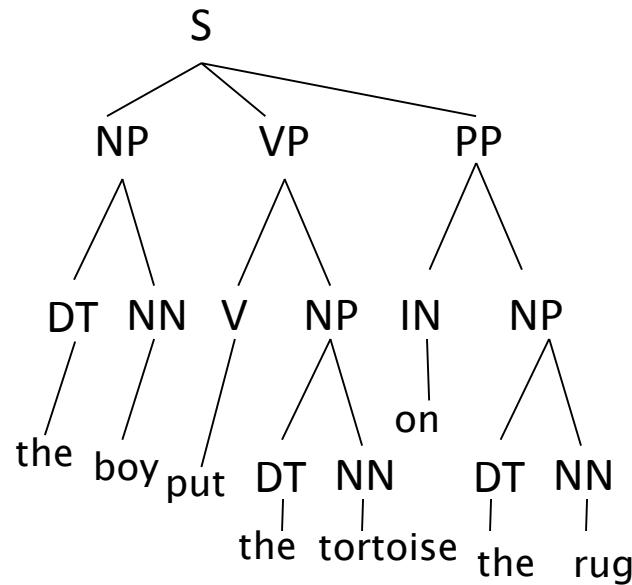


# Example Application: Machine Translation

- The boy put the tortoise on the rug

          |     |     |     |  
लड़के ने रखा कछुआ ऊपर कालीन

- SVO vs. SOV; preposition vs. post-position

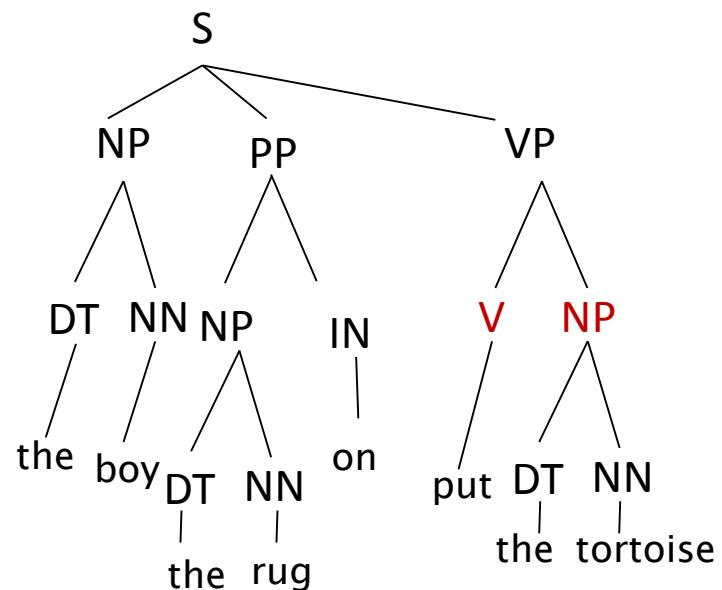
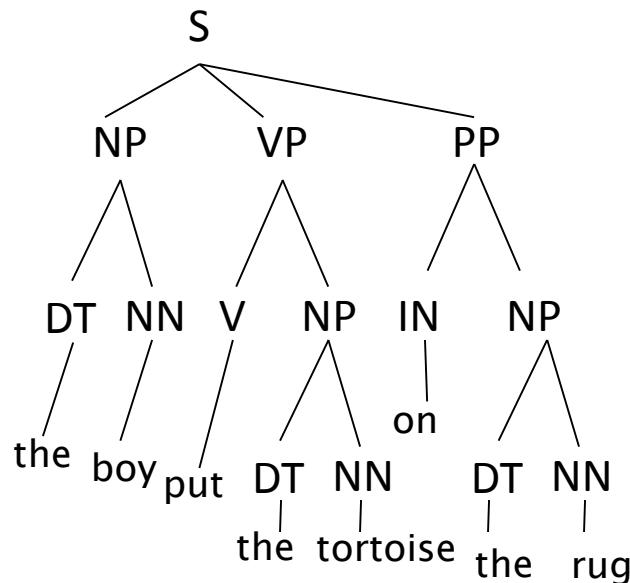


# Example Application: Machine Translation

- The boy put the tortoise on the rug

          |     |     |     |  
लड़के ने रखा कछुआ ऊपर कालीन

- SVO vs. SOV; preposition vs. post-position

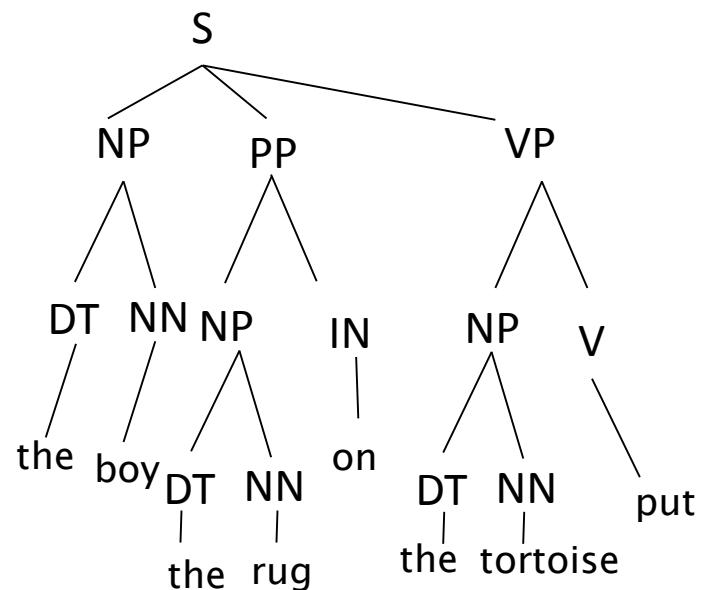
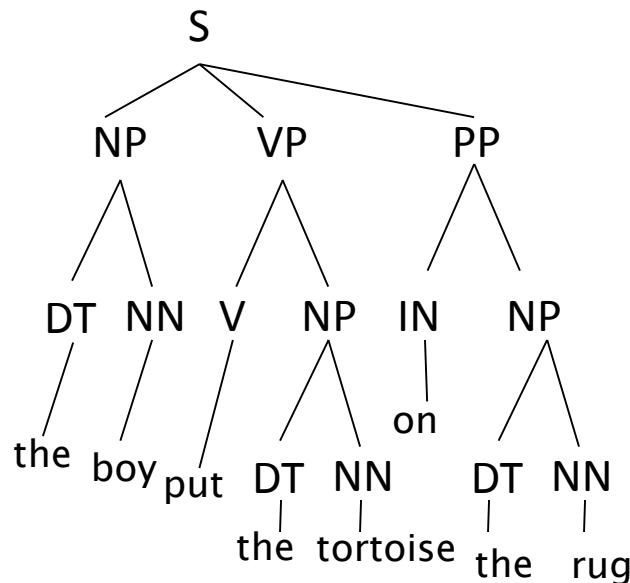


# Example Application: Machine Translation

- The boy put the tortoise on the rug

          |     |     |     |  
लड़के ने रखा कछुआ ऊपर कालीन

- SVO vs. SOV; preposition vs. post-position

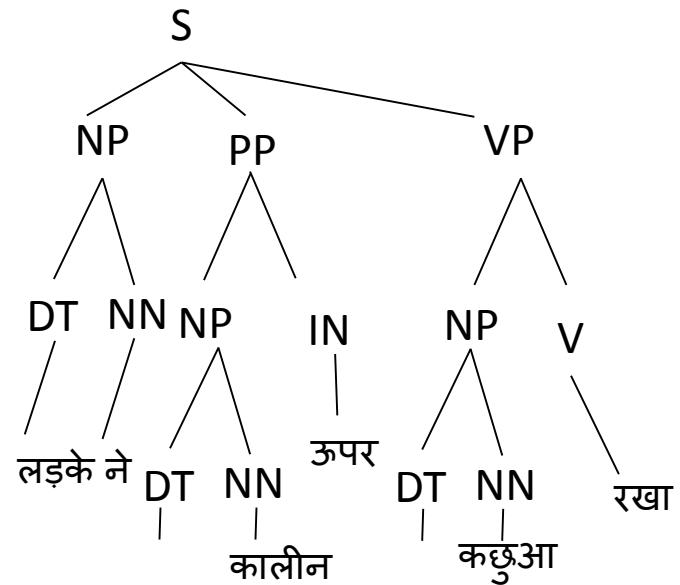
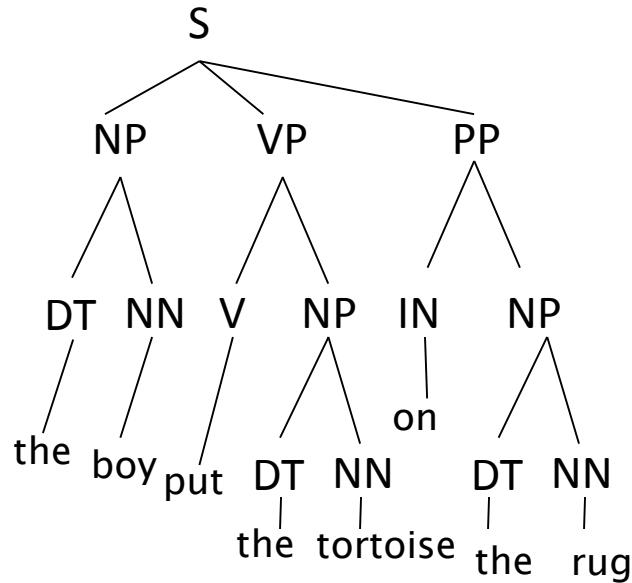


# Example Application: Machine Translation

- The boy put the tortoise on the rug

          |      |      |      |  
लड़के ने रखा कछुआ ऊपर कालीन

- SVO vs. SOV; preposition vs. post-position



# Pre 1990 (“Classical”) NLP Parsing

- Goes back to Chomsky’s PhD thesis in 1950s
- Wrote symbolic grammar (CFG or often richer) and lexicon

$S \rightarrow NP\ VP$

$NN \rightarrow interest$

$NP \rightarrow (DT)\ NN$

$NNS \rightarrow rates$

$NP \rightarrow NN\ NNS$

$NNS \rightarrow raises$

$NP \rightarrow NNP$

$VBP \rightarrow interest$

$VP \rightarrow V\ NP$

$VBZ \rightarrow rates$

- Used grammar/proof systems to prove parses from words
- This scaled very badly and didn’t give coverage. For sentence:

*Fed raises interest rates 0.5% in effort to control inflation*

- Minimal grammar: 36 parses
- Simple 10 rule grammar: 592 parses
- Real-size broad-coverage grammar: millions of parses

# Classical NLP Parsing: The problem and its solution

- Categorical constraints can be added to grammars to limit unlikely/weird parses for sentences
  - But the attempt make the grammars not robust
    - In traditional systems, commonly 30% of sentences in even an edited text would have *no* parse.
- A less constrained grammar can parse more sentences
  - But simple sentences end up with ever more parses with no way to choose between them
- We need mechanisms that allow us to find ***the most likely parse(s)*** for a sentence
  - Statistical parsing lets us work with very loose grammars that admit millions of parses for sentences but still quickly find the best parse(s)

# **Context Free Grammars and Ambiguities**

# Context-Free Grammars

Hopcroft and Ullman, 1979

A context free grammar  $G = (N, \Sigma, R, S)$  where:

- ▶  $N$  is a set of non-terminal symbols
- ▶  $\Sigma$  is a set of terminal symbols
- ▶  $R$  is a set of rules of the form  $X \rightarrow Y_1 Y_2 \dots Y_n$   
for  $n \geq 0$ ,  $X \in N$ ,  $Y_i \in (N \cup \Sigma)$
- ▶  $S \in N$  is a distinguished start symbol

# Context-Free Grammars in NLP

- A context free grammar G in NLP =  $(N, C, \Sigma, S, L, R)$ 
  - $\Sigma$  is a set of terminal symbols
  - $C$  is a set of preterminal symbols
  - $N$  is a set of nonterminal symbols
  - $S$  is the start symbol ( $S \in N$ )
  - $L$  is the lexicon, a set of items of the form  $X \rightarrow x$ 
    - $X \in C$  and  $x \in \Sigma$
  - $R$  is the grammar, a set of items of the form  $X \rightarrow \gamma$ 
    - $X \in N$  and  $\gamma \in (N \cup C)^*$
- By usual convention,  $S$  is the start symbol, but in statistical NLP, we usually have an extra node at the top (ROOT, TOP)
- We usually write  $e$  for an empty sequence, rather than nothing

# A Context Free Grammar of English

$$N = \{S, NP, VP, PP, DT, Vi, Vt, NN, IN\}$$

$$S = S$$

$$\Sigma = \{\text{sleeps, saw, man, woman, telescope, the, with, in}\}$$

$R =$

|    |               |    |    |
|----|---------------|----|----|
| S  | $\rightarrow$ | NP | VP |
| VP | $\rightarrow$ | Vi |    |
| VP | $\rightarrow$ | Vt | NP |
| VP | $\rightarrow$ | VP | PP |
| NP | $\rightarrow$ | DT | NN |
| NP | $\rightarrow$ | NP | PP |
| PP | $\rightarrow$ | IN | NP |

|    |               |           |
|----|---------------|-----------|
| Vi | $\rightarrow$ | sleeps    |
| Vt | $\rightarrow$ | saw       |
| NN | $\rightarrow$ | man       |
| NN | $\rightarrow$ | woman     |
| NN | $\rightarrow$ | telescope |
| DT | $\rightarrow$ | the       |
| IN | $\rightarrow$ | with      |
| IN | $\rightarrow$ | in        |

Note: S=sentence, VP=verb phrase, NP=noun phrase,  
PP=prepositional phrase, DT=determiner, Vi=intransitive verb,  
Vt=transitive verb, NN=noun, IN=preposition

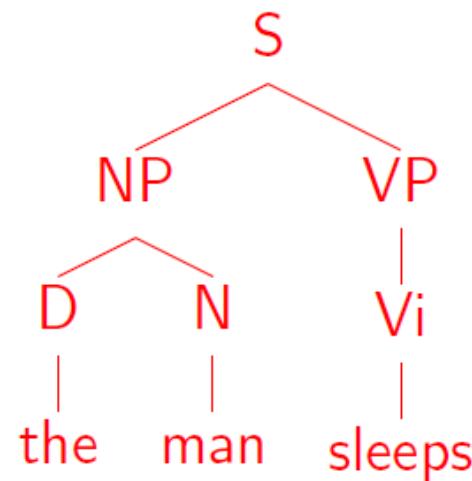
# Left-Most Derivations

A left-most derivation is a sequence of strings  $s_1 \dots s_n$ , where

- ▶  $s_1 = S$ , the start symbol
- ▶  $s_n \in \Sigma^*$ , i.e.  $s_n$  is made up of terminal symbols only
- ▶ Each  $s_i$  for  $i = 2 \dots n$  is derived from  $s_{i-1}$  by picking the left-most non-terminal  $X$  in  $s_{i-1}$  and replacing it by some  $\beta$  where  $X \rightarrow \beta$  is a rule in  $R$

For example: [S], [NP VP], [D N VP], [the N VP], [the man VP], [the man Vi], [the man sleeps]

Representation of a derivation as a tree:



# Properties of CFGs

- ▶ A CFG defines a set of possible derivations
- ▶ A string  $s \in \Sigma^*$  is in the *language* defined by the CFG if there is at least one derivation that yields  $s$
- ▶ Each string in the language generated by the CFG may have more than one derivation (“ambiguity”)

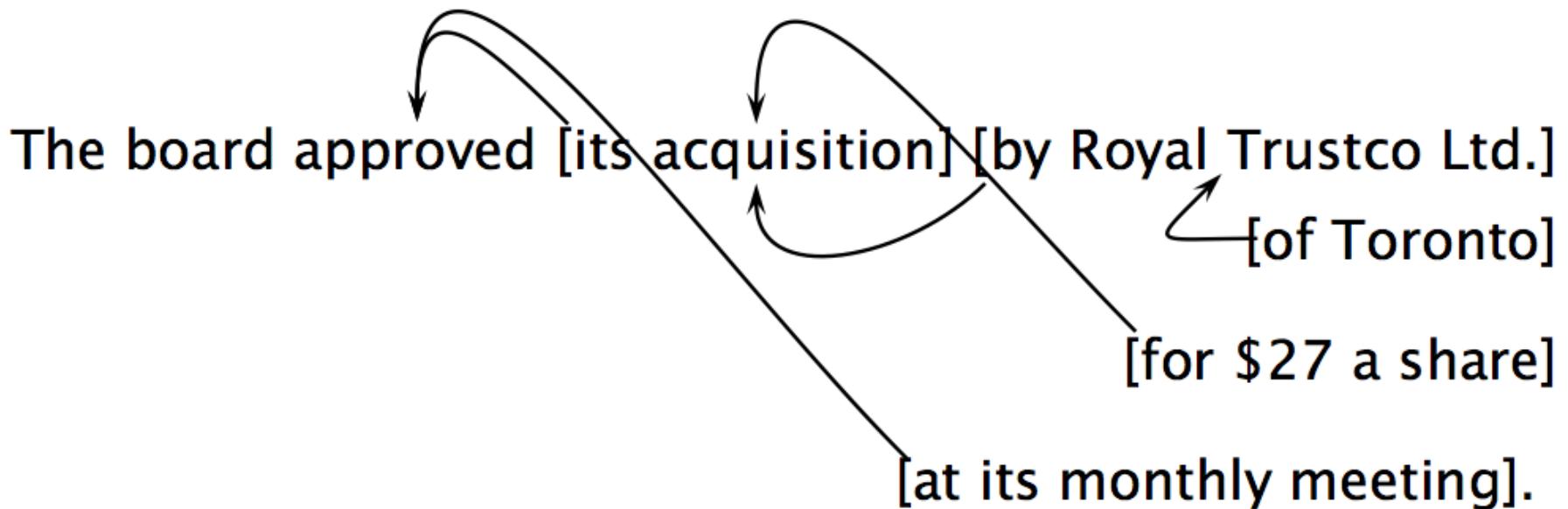
# Attachment ambiguities

- A key parsing decision is how we ‘attach’ various constituents
  - PPs, adverbial or participial phrases, infinitives, coordinations, etc.

The board approved [its acquisition] [by Royal Trustco Ltd.]  
[of Toronto]  
[for \$27 a share]  
[at its monthly meeting].

# Attachment ambiguities

- A key parsing decision is how we ‘attach’ various constituents
  - PPs, adverbial or participial phrases, infinitives, coordinations, etc.



- Catalan numbers:  $C_n = (2n)!/[(n+1)!n!]$
- An exponentially growing series, which arises in many tree-like contexts:
  - E.g., the number of possible triangulations of a polygon with  $n+2$  sides
  - Turns up in triangulation of probabilistic graphical models....

# Attachments

- I cleaned the dishes from dinner
- I cleaned the dishes with detergent
- I cleaned the dishes in my pajamas
- I cleaned the dishes in the sink

# Syntactic Ambiguities I

- Prepositional phrases:  
*They cooked the beans in the pot on the stove with handles.*
- Particle vs. preposition:  
*The lady dressed up the staircase.*
- Complement structures  
*The tourists objected to the guide that they couldn't hear.*  
*She knows you like the back of her hand.*
- Gerund vs. participial adjective  
*Visiting relatives can be boring.*  
*Changing schedules frequently confused passengers.*

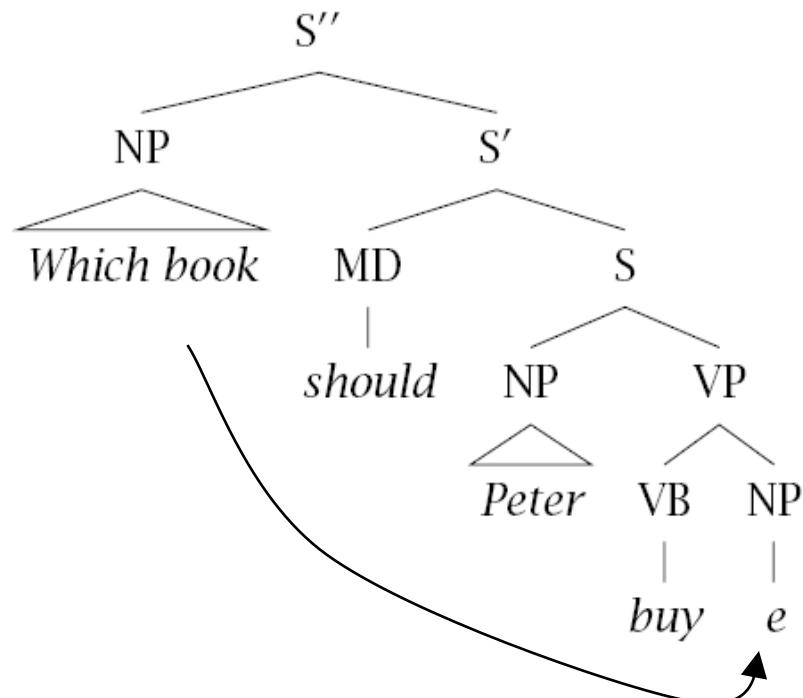
## Syntactic Ambiguities II

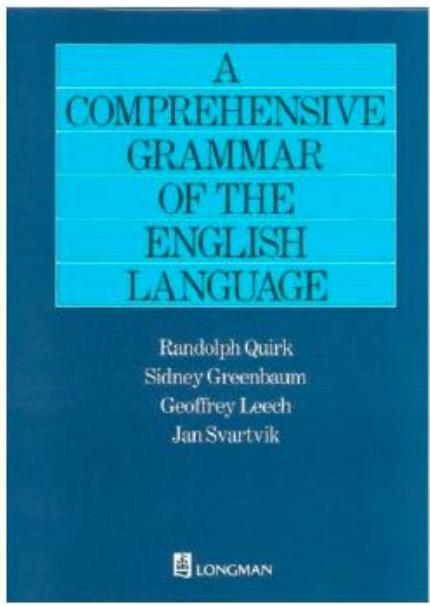
- Modifier scope within NPs  
*impractical design requirements*  
*plastic cup holder*
- Multiple gap constructions  
*The chicken is ready to eat.*  
*The contractors are rich enough to sue.*
- Coordination scope:  
*Small rats and mice can squeeze into holes or cracks in the wall.*

# Non-Local Phenomena

- Dislocation / gapping
  - Which book should Peter buy?
  - A debate arose which continued until the election.

- Binding
  - Reference
    - The IRS audits itself
  - Control
    - I want to go
    - I want you to go





### Product Details (from Amazon)

Hardcover: 1779 pages

Publisher: Longman; 2nd Revised edition

Language: English

ISBN-10: 0582517346

ISBN-13: 978-0582517349

Product Dimensions: 8.4 x 2.4 x 10 inches

Shipping Weight: 4.6 pounds

# A Fragment of a Noun Phrase Grammar

|           |               |                     |
|-----------|---------------|---------------------|
| $\bar{N}$ | $\Rightarrow$ | NN                  |
| $\bar{N}$ | $\Rightarrow$ | NN $\bar{N}$        |
| $\bar{N}$ | $\Rightarrow$ | JJ $\bar{N}$        |
| $\bar{N}$ | $\Rightarrow$ | $\bar{N}$ $\bar{N}$ |
| NP        | $\Rightarrow$ | DT $\bar{N}$        |

|    |               |          |
|----|---------------|----------|
| NN | $\Rightarrow$ | box      |
| NN | $\Rightarrow$ | car      |
| NN | $\Rightarrow$ | mechanic |
| NN | $\Rightarrow$ | pigeon   |
| DT | $\Rightarrow$ | the      |
| DT | $\Rightarrow$ | a        |

|    |               |            |
|----|---------------|------------|
| JJ | $\Rightarrow$ | fast       |
| JJ | $\Rightarrow$ | metal      |
| JJ | $\Rightarrow$ | idealistic |
| JJ | $\Rightarrow$ | clay       |

# Extended Grammar with Prepositional Phrases

|           |               |           |           |    |               |          |    |               |            |
|-----------|---------------|-----------|-----------|----|---------------|----------|----|---------------|------------|
| $\bar{N}$ | $\Rightarrow$ | NN        |           | NN | $\Rightarrow$ | box      | JJ | $\Rightarrow$ | fast       |
| $\bar{N}$ | $\Rightarrow$ | NN        | $\bar{N}$ | NN | $\Rightarrow$ | car      | JJ | $\Rightarrow$ | metal      |
| $\bar{N}$ | $\Rightarrow$ | JJ        | $\bar{N}$ | NN | $\Rightarrow$ | mechanic | JJ | $\Rightarrow$ | idealistic |
| $\bar{N}$ | $\Rightarrow$ | $\bar{N}$ | $\bar{N}$ | NN | $\Rightarrow$ | pigeon   | JJ | $\Rightarrow$ | clay       |
| NP        | $\Rightarrow$ | DT        | $\bar{N}$ | NN |               |          | IN | $\Rightarrow$ | in         |
| PP        | $\Rightarrow$ | IN        | NP        | DT | $\Rightarrow$ | the      | IN | $\Rightarrow$ | under      |
| $\bar{N}$ | $\Rightarrow$ | $\bar{N}$ | PP        | DT | $\Rightarrow$ | a        | IN | $\Rightarrow$ | of         |
|           |               |           |           |    |               |          | IN | $\Rightarrow$ | on         |
|           |               |           |           |    |               |          | IN | $\Rightarrow$ | with       |
|           |               |           |           |    |               |          | IN | $\Rightarrow$ | as         |

**Generates:**

in a box, under the box, the fast car mechanic under the pigeon in the box, ...

# Verbs, Verb Phrases and Sentences

- ▶ Basic Verb Types

$V_i$  = Intransitive verb      e.g., sleeps, walks, laughs

$V_t$  = Transitive verb      e.g., sees, saw, likes

$V_d$  = Ditransitive verb      e.g., gave

- ▶ Basic VP Rules

$VP \rightarrow V_i$

$VP \rightarrow V_t \quad NP$

$VP \rightarrow V_d \quad NP \quad NP$

- ▶ Basic S Rule

$S \rightarrow NP \quad VP$

## Examples of VP:

sleeps, walks, likes the mechanic, gave the mechanic the fast car

## Examples of S:

the man sleeps, the dog walks, the dog gave the mechanic the fast car

# PPs Modifying Verb Phrases

**A new rule:** VP → VP PP

**New examples of VP:**

sleeps in the car, walks like the mechanic, gave the mechanic the fast car on Tuesday, ...

# Complementizers and SBARs

- ▶ Complementizers  
COMP = complementizer e.g., that
- ▶ SBAR  
SBAR → COMP S

## Examples:

that the man sleeps, that the mechanic saw the dog ..

# More Verbs

- ▶ New Verb Types
  - V[5] e.g., said, reported
  - V[6] e.g., told, informed
  - V[7] e.g., bet

- ▶ New VP Rules
  - VP → V[5] SBAR
  - VP → V[6] NP SBAR
  - VP → V[7] NP NP SBAR

## Examples of New VPs:

said that the man sleeps

told the dog that the mechanic likes the pigeon

bet the pigeon \$50 that the mechanic owns a fast car

# Coordination

- ▶ A New Part-of-Speech:  
CC = Coordinator e.g., and, or, but

- ▶ New Rules

NP → NP      CC      NP

Ṅ → Ṅ      CC      Ṅ

VP → VP      CC      VP

S → S      CC      S

SBAR → SBAR      CC      SBAR

# Much more remains...

- ▶ Agreement

The dogs laugh *vs.* The dog laughs

- ▶ Wh-movement

The dog that the cat liked \_\_\_

- ▶ Active vs. passive

The dog saw the cat *vs.*

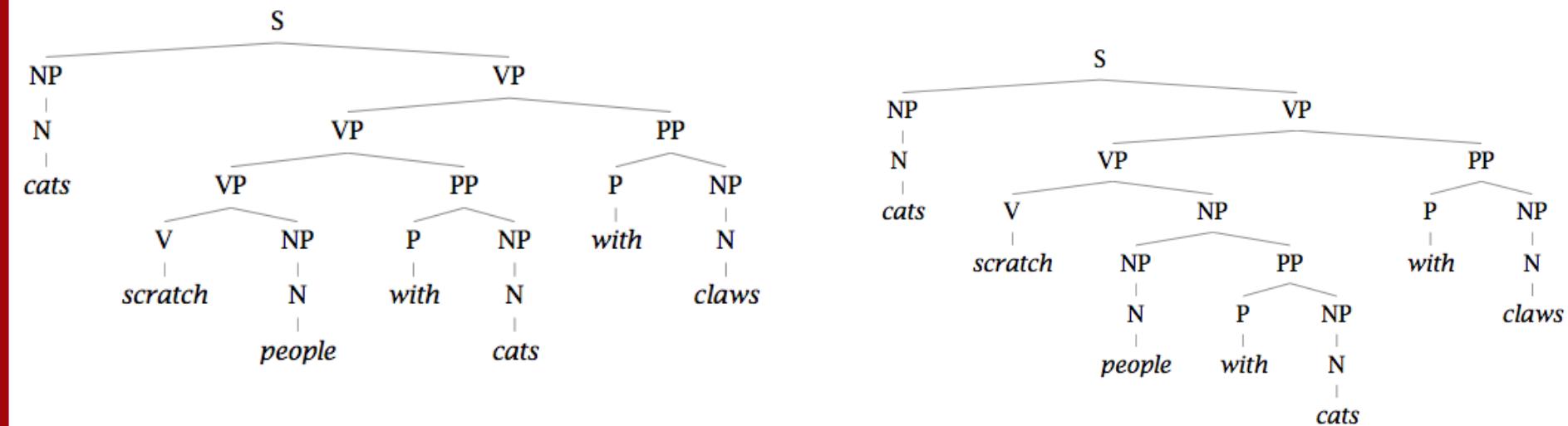
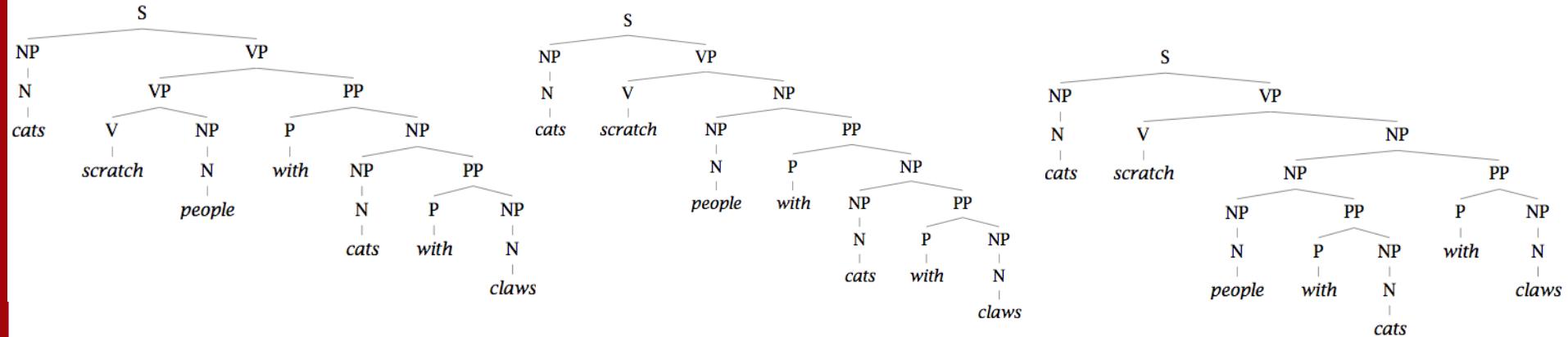
The cat was seen by the dog

- ▶ If you're interested in reading more:

*Syntactic Theory: A Formal Introduction, 2nd Edition.* Ivan A. Sag, Thomas Wasow, and Emily M. Bender.

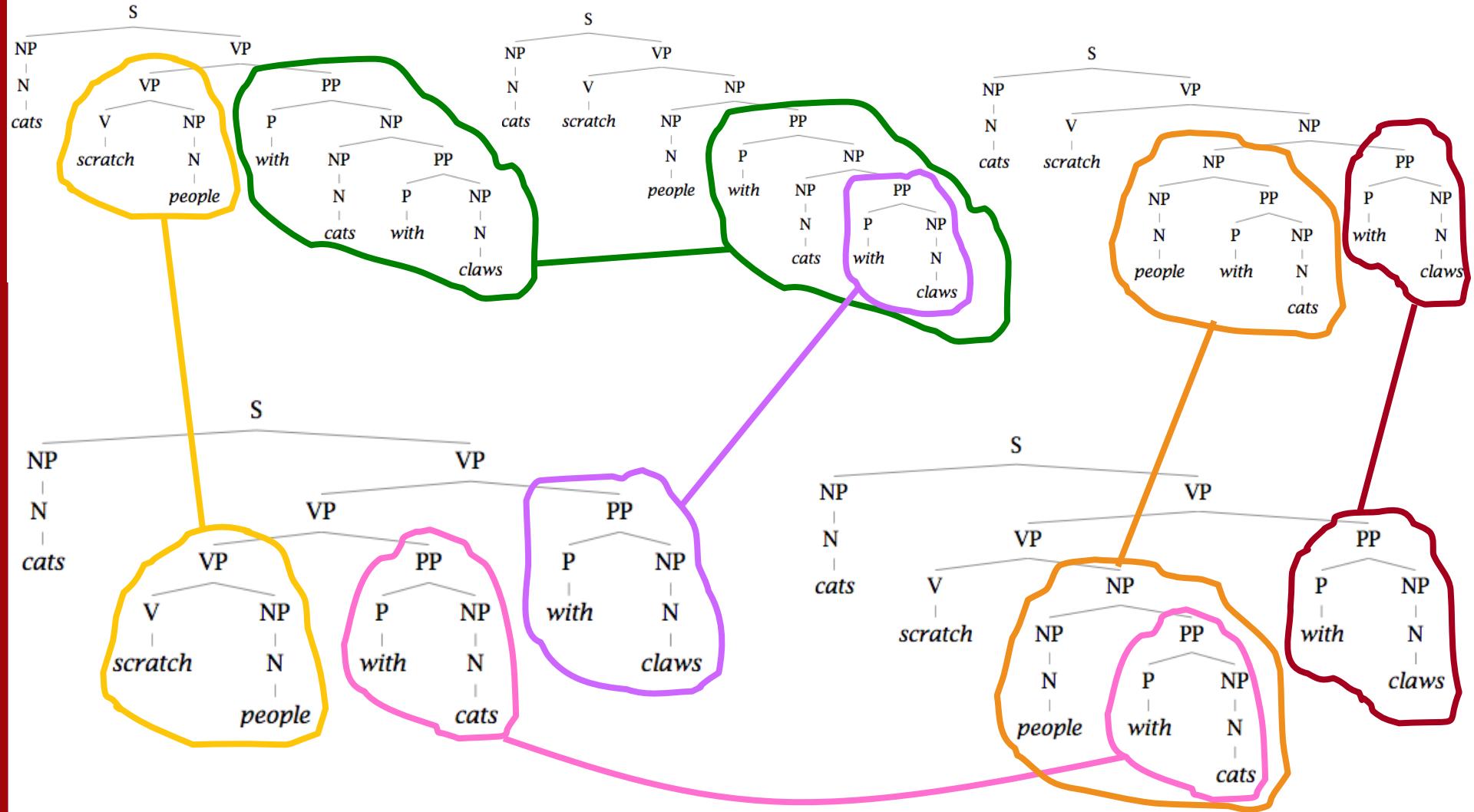
# Parsing: Two problems to solve:

## 1. Repeated work...



# Parsing: Two problems to solve:

## 1. Repeated work...



# Parsing: Two problems to solve:

## 2. Choosing the correct parse

- How do we work out the correct attachment:
  - She saw the man with a telescope
  - Is the problem ‘AI complete’? Yes, but ...
  - Words are good predictors of attachment
    - Even absent full understanding
    - Moscow sent more than 100,000 soldiers into Afghanistan ...
    - Sydney Water breached an agreement with NSW Health ...
  - Our statistical parsers will try to exploit such statistics.

# Probabilistic Context Free Grammar

# Probabilistic – or stochastic – context-free grammars (PCFGs)

- $G = (\Sigma, N, S, R, P)$ 
  - $\Sigma$  is a set of terminal symbols
  - $N$  is a set of nonterminal symbols
  - $S$  is the start symbol ( $S \in N$ )
  - $R$  is a set of rules/productions of the form  $X \rightarrow \gamma$
  - $P$  is a probability function
    - $P: R \rightarrow [0,1]$
    - $\forall X \in N, \sum_{X \rightarrow \gamma \in R} P(X \rightarrow \gamma) = 1$
- A grammar  $G$  generates a language model  $L$ .

$$\sum_{g \vdash T^*} P(g) = 1$$

# PCFG Example

|    |               |    |    |     |
|----|---------------|----|----|-----|
| S  | $\Rightarrow$ | NP | VP | 1.0 |
| VP | $\Rightarrow$ | Vi |    | 0.4 |
| VP | $\Rightarrow$ | Vt | NP | 0.4 |
| VP | $\Rightarrow$ | VP | PP | 0.2 |
| NP | $\Rightarrow$ | DT | NN | 0.3 |
| NP | $\Rightarrow$ | NP | PP | 0.7 |
| PP | $\Rightarrow$ | P  | NP | 1.0 |

|    |               |           |     |
|----|---------------|-----------|-----|
| Vi | $\Rightarrow$ | sleeps    | 1.0 |
| Vt | $\Rightarrow$ | saw       | 1.0 |
| NN | $\Rightarrow$ | man       | 0.7 |
| NN | $\Rightarrow$ | woman     | 0.2 |
| NN | $\Rightarrow$ | telescope | 0.1 |
| DT | $\Rightarrow$ | the       | 1.0 |
| IN | $\Rightarrow$ | with      | 0.5 |
| IN | $\Rightarrow$ | in        | 0.5 |

- Probability of a tree t with rules

$$\alpha_1 \rightarrow \beta_1, \alpha_2 \rightarrow \beta_2, \dots, \alpha_n \rightarrow \beta_n$$

is

$$p(t) = \prod_{i=1}^n q(\alpha_i \rightarrow \beta_i)$$

where  $q(\alpha \rightarrow \beta)$  is the probability for rule  $\alpha \rightarrow \beta$ .

## Example of a PCFG

|    |               |       |     |
|----|---------------|-------|-----|
| S  | $\Rightarrow$ | NP VP | 1.0 |
| VP | $\Rightarrow$ | Vi    | 0.4 |
| VP | $\Rightarrow$ | Vt NP | 0.4 |
| VP | $\Rightarrow$ | VP PP | 0.2 |
| NP | $\Rightarrow$ | DT NN | 0.3 |
| NP | $\Rightarrow$ | NP PP | 0.7 |
| PP | $\Rightarrow$ | P NP  | 1.0 |

|    |               |           |     |
|----|---------------|-----------|-----|
| Vi | $\Rightarrow$ | sleeps    | 1.0 |
| Vt | $\Rightarrow$ | saw       | 1.0 |
| NN | $\Rightarrow$ | man       | 0.7 |
| NN | $\Rightarrow$ | woman     | 0.2 |
| NN | $\Rightarrow$ | telescope | 0.1 |
| DT | $\Rightarrow$ | the       | 1.0 |
| IN | $\Rightarrow$ | with      | 0.5 |
| IN | $\Rightarrow$ | in        | 0.5 |

- ▶ Probability of a tree  $t$  with rules

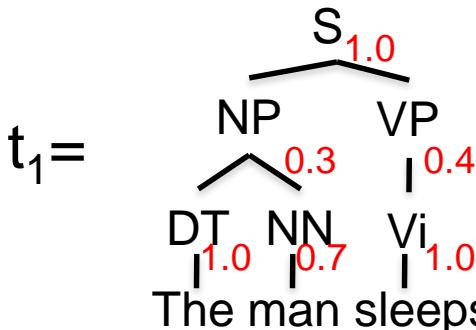
$$\alpha_1 \rightarrow \beta_1, \alpha_2 \rightarrow \beta_2, \dots, \alpha_n \rightarrow \beta_n$$

is  $p(t) = \prod_{i=1}^n q(\alpha_i \rightarrow \beta_i)$  where  $q(\alpha \rightarrow \beta)$  is the probability for rule  $\alpha \rightarrow \beta$ .

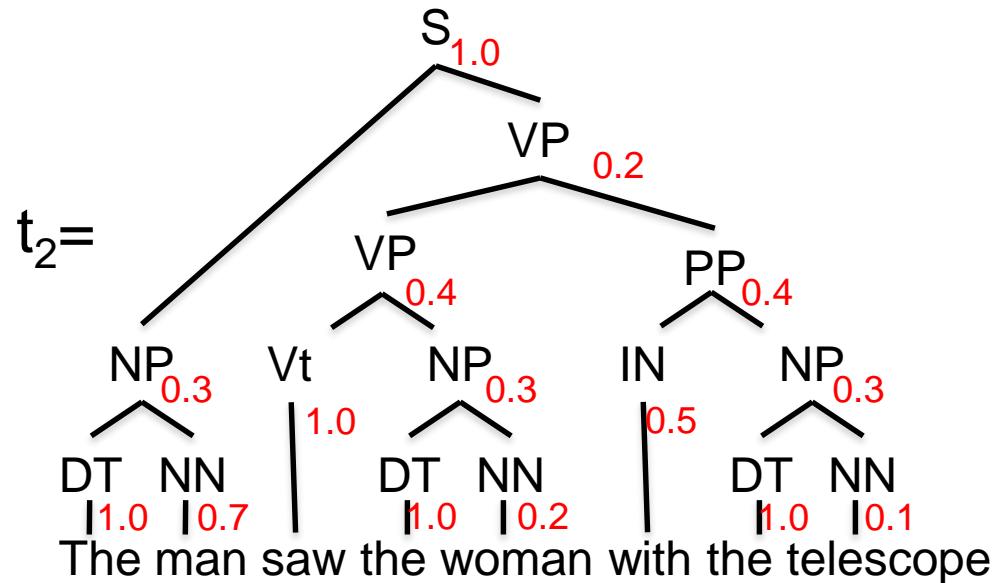
# Probability of a Parse

|    |               |    |    |     |
|----|---------------|----|----|-----|
| S  | $\Rightarrow$ | NP | VP | 1.0 |
| VP | $\Rightarrow$ | Vi |    | 0.4 |
| VP | $\Rightarrow$ | Vt | NP | 0.4 |
| VP | $\Rightarrow$ | VP | PP | 0.2 |
| NP | $\Rightarrow$ | DT | NN | 0.3 |
| NP | $\Rightarrow$ | NP | PP | 0.7 |
| PP | $\Rightarrow$ | P  | NP | 1.0 |

|    |               |           |     |
|----|---------------|-----------|-----|
| Vi | $\Rightarrow$ | sleeps    | 1.0 |
| Vt | $\Rightarrow$ | saw       | 1.0 |
| NN | $\Rightarrow$ | man       | 0.7 |
| NN | $\Rightarrow$ | woman     | 0.2 |
| NN | $\Rightarrow$ | telescope | 0.1 |
| DT | $\Rightarrow$ | the       | 1.0 |
| IN | $\Rightarrow$ | with      | 0.5 |
| IN | $\Rightarrow$ | in        | 0.5 |



$$p(t_1) = 1.0 * 0.3 * 1.0 * 0.7 * 0.4 * 1.0$$



$$p(t_s) = 1.0 * 0.3 * 1.0 * 0.7 * 0.2 * 0.4 * 1.0 * 0.3 * 1.0 * 0.2 * 0.4 * 0.5 * 0.3 * 1.0 * 0.1$$

# PCFGs: Learning and Inference

- Model

- The probability of a tree  $t$  with  $n$  rules  $\alpha_i \rightarrow \beta_i$ ,  $i = 1..n$

$$p(t) = \prod_{i=1}^n q(\alpha_i \rightarrow \beta_i)$$

- Learning

- Read the rules off of labeled sentences, use ML estimates for probabilities

$$q_{ML}(\alpha \rightarrow \beta) = \frac{\text{Count}(\alpha \rightarrow \beta)}{\text{Count}(\alpha)}$$

- and use all of our standard smoothing tricks!

- Inference

- For input sentence  $s$ , define  $T(s)$  to be the set of trees whose *yield* is  $s$  (whole leaves, read left to right, match the words in  $s$ )

$$t^*(s) = \arg \max_{t \in T(s)} p(t)$$

# Grammar Transforms

# Chomsky Normal Form

- All rules are of the form  $X \rightarrow YZ$  or  $X \rightarrow w$ 
  - $X, Y, Z \in N$  and  $w \in \Sigma$
- A transformation to this form doesn't change the weak generative capacity of a CFG
  - That is, it recognizes the same language
    - But maybe with different trees
- Empties and unaries are removed recursively
- n-ary rules are divided by introducing new nonterminals ( $n > 2$ )

# A phrase structure grammar

$S \rightarrow NP\ VP$

$N \rightarrow people$

$VP \rightarrow V\ NP$

$N \rightarrow fish$

$VP \rightarrow V\ NP\ PP$

$N \rightarrow tanks$

$NP \rightarrow NP\ NP$

$N \rightarrow rods$

$NP \rightarrow NP\ PP$

$V \rightarrow people$

$NP \rightarrow N$

$V \rightarrow fish$

$NP \rightarrow e$

$V \rightarrow tanks$

$PP \rightarrow P\ NP$

$P \rightarrow with$

# Chomsky Normal Form steps

$S \rightarrow NP\ VP$

$S \rightarrow VP$

$VP \rightarrow V\ NP$

$VP \rightarrow V$

$VP \rightarrow V\ NP\ PP$

$VP \rightarrow V\ PP$

$NP \rightarrow NP\ NP$

$NP \rightarrow NP$

$NP \rightarrow NP\ PP$

$NP \rightarrow PP$

$NP \rightarrow N$

$PP \rightarrow P\ NP$

$PP \rightarrow P$

$N \rightarrow people$

$N \rightarrow fish$

$N \rightarrow tanks$

$N \rightarrow rods$

$V \rightarrow people$

$V \rightarrow fish$

$V \rightarrow tanks$

$P \rightarrow with$

# Chomsky Normal Form steps

$S \rightarrow NP\ VP$

$VP \rightarrow V\ NP$

$S \rightarrow V\ NP$

$VP \rightarrow V$

$S \rightarrow V$

$VP \rightarrow V\ NP\ PP$

$S \rightarrow V\ NP\ PP$

$VP \rightarrow V\ PP$

$S \rightarrow V\ PP$

$NP \rightarrow NP\ NP$

$NP \rightarrow NP$

$NP \rightarrow NP\ PP$

$NP \rightarrow PP$

$NP \rightarrow N$

$PP \rightarrow P\ NP$

$PP \rightarrow P$

$N \rightarrow people$

$N \rightarrow fish$

$N \rightarrow tanks$

$N \rightarrow rods$

$V \rightarrow people$

$V \rightarrow fish$

$V \rightarrow tanks$

$P \rightarrow with$

# Chomsky Normal Form steps

$S \rightarrow NP\ VP$

$VP \rightarrow V\ NP$

$S \rightarrow V\ NP$

$VP \rightarrow V$

$VP \rightarrow V\ NP\ PP$

$S \rightarrow V\ NP\ PP$

$VP \rightarrow V\ PP$

$S \rightarrow V\ PP$

$NP \rightarrow NP\ NP$

$NP \rightarrow NP$

$NP \rightarrow NP\ PP$

$NP \rightarrow PP$

$NP \rightarrow N$

$PP \rightarrow P\ NP$

$PP \rightarrow P$

$N \rightarrow people$

$N \rightarrow fish$

$N \rightarrow tanks$

$N \rightarrow rods$

$V \rightarrow people$

$S \rightarrow people$

$V \rightarrow fish$

$S \rightarrow fish$

$V \rightarrow tanks$

$S \rightarrow tanks$

$P \rightarrow with$

# Chomsky Normal Form steps

$S \rightarrow NP\ VP$

$N \rightarrow people$

$VP \rightarrow V\ NP$

$N \rightarrow fish$

$S \rightarrow V\ NP$

$N \rightarrow tanks$

$VP \rightarrow V\ NP\ PP$

$N \rightarrow rods$

$S \rightarrow V\ NP\ PP$

$V \rightarrow people$

$VP \rightarrow V\ PP$

$S \rightarrow people$

$S \rightarrow V\ PP$

$VP \rightarrow people$

$NP \rightarrow NP\ NP$

$V \rightarrow fish$

$NP \rightarrow NP$

$S \rightarrow fish$

$NP \rightarrow NP\ PP$

$VP \rightarrow fish$

$NP \rightarrow PP$

$V \rightarrow tanks$

$NP \rightarrow N$

$S \rightarrow tanks$

$PP \rightarrow P\ NP$

$VP \rightarrow tanks$

$PP \rightarrow P$

$P \rightarrow with$

# Chomsky Normal Form steps

|                            |                         |
|----------------------------|-------------------------|
| $S \rightarrow NP\ VP$     | $NP \rightarrow people$ |
| $VP \rightarrow V\ NP$     | $NP \rightarrow fish$   |
| $S \rightarrow V\ NP$      | $NP \rightarrow tanks$  |
| $VP \rightarrow V\ NP\ PP$ | $NP \rightarrow rods$   |
| $S \rightarrow V\ NP\ PP$  | $V \rightarrow people$  |
| $VP \rightarrow V\ PP$     | $S \rightarrow people$  |
| $S \rightarrow V\ PP$      | $VP \rightarrow people$ |
| $NP \rightarrow NP\ NP$    | $V \rightarrow fish$    |
| $NP \rightarrow NP\ PP$    | $S \rightarrow fish$    |
| $NP \rightarrow P\ NP$     | $VP \rightarrow fish$   |
| $PP \rightarrow P\ NP$     | $V \rightarrow tanks$   |
|                            | $S \rightarrow tanks$   |
|                            | $VP \rightarrow tanks$  |
|                            | $P \rightarrow with$    |
|                            | $PP \rightarrow with$   |

# Chomsky Normal Form steps

|                             |                         |
|-----------------------------|-------------------------|
| $S \rightarrow NP\ VP$      | $NP \rightarrow people$ |
| $VP \rightarrow V\ NP$      | $NP \rightarrow fish$   |
| $S \rightarrow V\ NP$       | $NP \rightarrow tanks$  |
| $VP \rightarrow V @VP\_V$   | $NP \rightarrow rods$   |
| $@VP\_V \rightarrow NP\ PP$ | $V \rightarrow people$  |
| $S \rightarrow V @S\_V$     | $S \rightarrow people$  |
| $@S\_V \rightarrow NP\ PP$  | $VP \rightarrow people$ |
| $VP \rightarrow V\ PP$      | $V \rightarrow fish$    |
| $S \rightarrow V\ PP$       | $S \rightarrow fish$    |
| $NP \rightarrow NP\ NP$     | $VP \rightarrow fish$   |
| $NP \rightarrow NP\ PP$     | $V \rightarrow tanks$   |
| $NP \rightarrow P\ NP$      | $S \rightarrow tanks$   |
| $PP \rightarrow P\ NP$      | $VP \rightarrow tanks$  |
|                             | $P \rightarrow with$    |
|                             | $PP \rightarrow with$   |

# A phrase structure grammar

$S \rightarrow NP\ VP$

$N \rightarrow people$

$VP \rightarrow V\ NP$

$N \rightarrow fish$

$VP \rightarrow V\ NP\ PP$

$N \rightarrow tanks$

$NP \rightarrow NP\ NP$

$N \rightarrow rods$

$NP \rightarrow NP\ PP$

$V \rightarrow people$

$NP \rightarrow N$

$V \rightarrow fish$

$NP \rightarrow e$

$V \rightarrow tanks$

$PP \rightarrow P\ NP$

$P \rightarrow with$

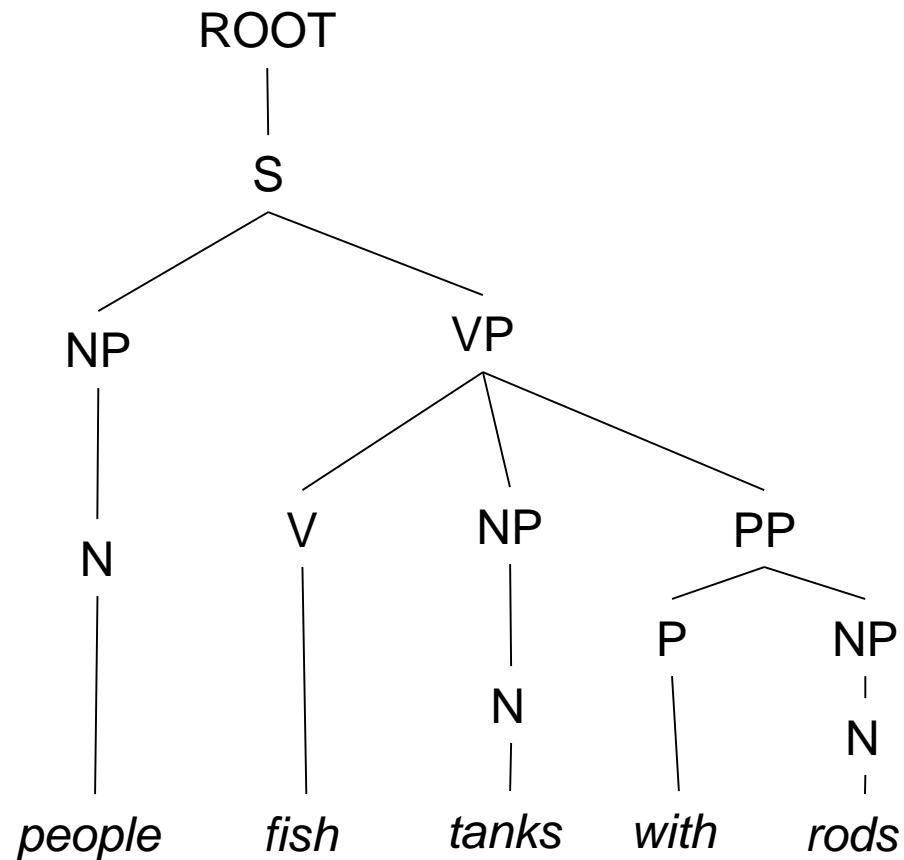
# Chomsky Normal Form steps

|                             |                         |
|-----------------------------|-------------------------|
| $S \rightarrow NP\ VP$      | $NP \rightarrow people$ |
| $VP \rightarrow V\ NP$      | $NP \rightarrow fish$   |
| $S \rightarrow V\ NP$       | $NP \rightarrow tanks$  |
| $VP \rightarrow V @VP\_V$   | $NP \rightarrow rods$   |
| $@VP\_V \rightarrow NP\ PP$ | $V \rightarrow people$  |
| $S \rightarrow V @S\_V$     | $S \rightarrow people$  |
| $@S\_V \rightarrow NP\ PP$  | $VP \rightarrow people$ |
| $VP \rightarrow V\ PP$      | $V \rightarrow fish$    |
| $S \rightarrow V\ PP$       | $S \rightarrow fish$    |
| $NP \rightarrow NP\ NP$     | $VP \rightarrow fish$   |
| $NP \rightarrow NP\ PP$     | $V \rightarrow tanks$   |
| $NP \rightarrow P\ NP$      | $S \rightarrow tanks$   |
| $PP \rightarrow P\ NP$      | $VP \rightarrow tanks$  |
|                             | $P \rightarrow with$    |
|                             | $PP \rightarrow with$   |

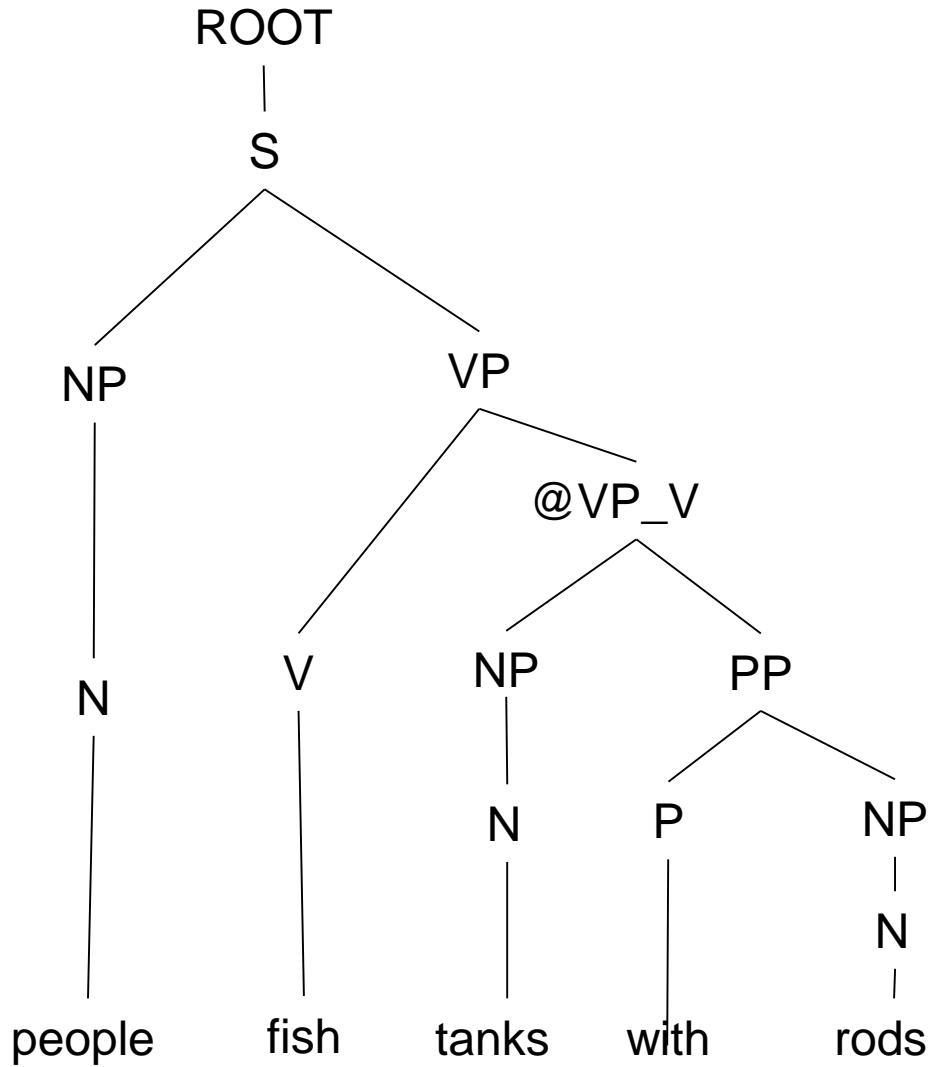
# Chomsky Normal Form

- You should think of this as a transformation for efficient parsing
- With some extra book-keeping in symbol names, you can even reconstruct the same trees with a detransform
- In practice full Chomsky Normal Form is a pain
  - Reconstructing n-aries is easy
  - Reconstructing unaries/empties is trickier
- **Binarization** is crucial for cubic time CFG parsing
- The rest isn't necessary; it just makes the algorithms cleaner and a bit quicker

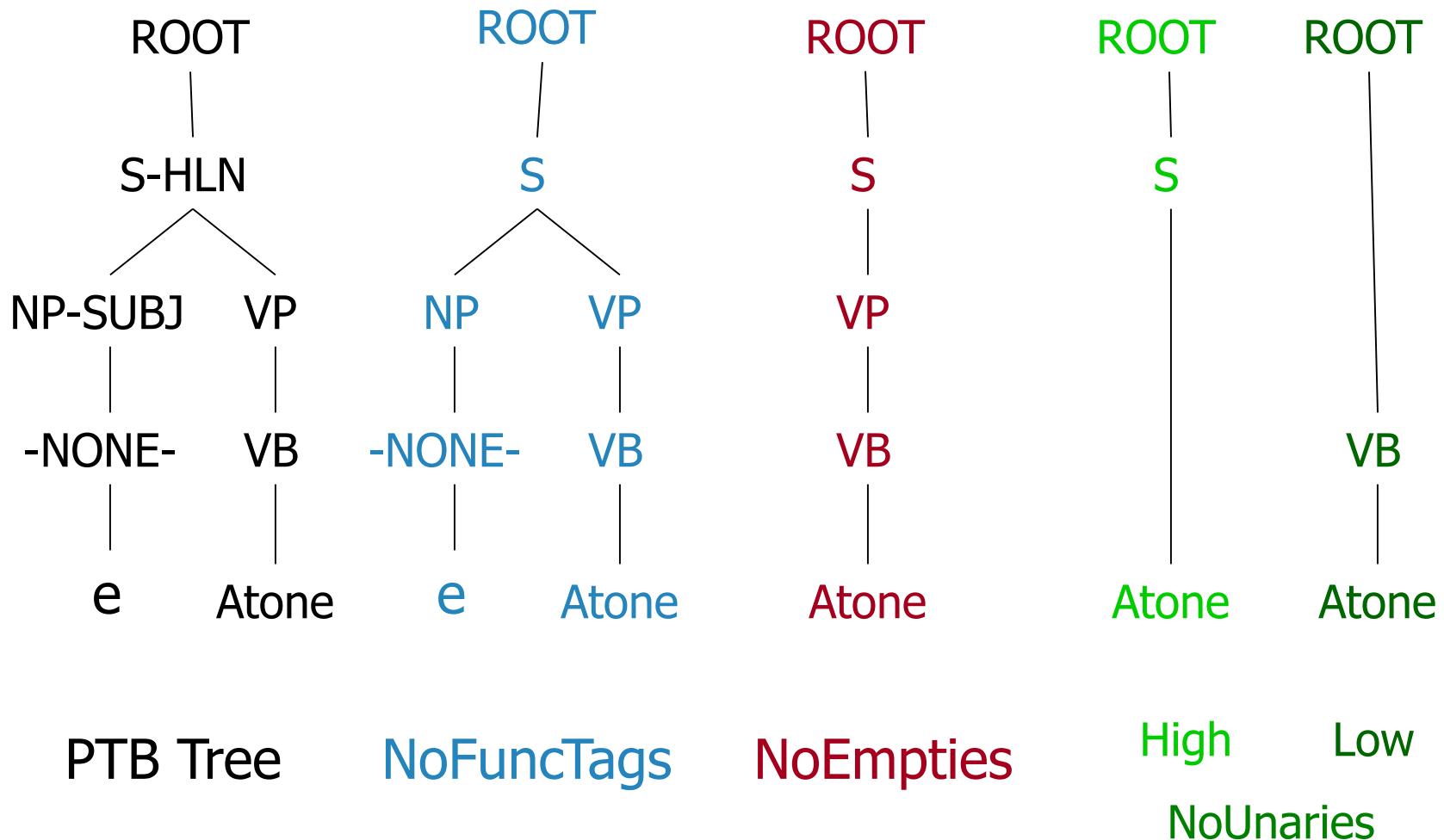
# An example: before binarization...



# After binarization...

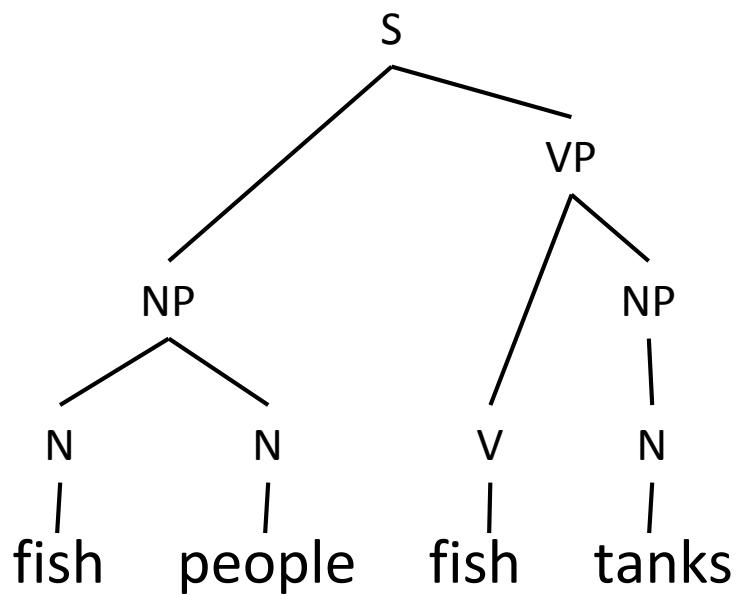


# Treebank: empties and unaries



# Parsing

# Constituency Parsing

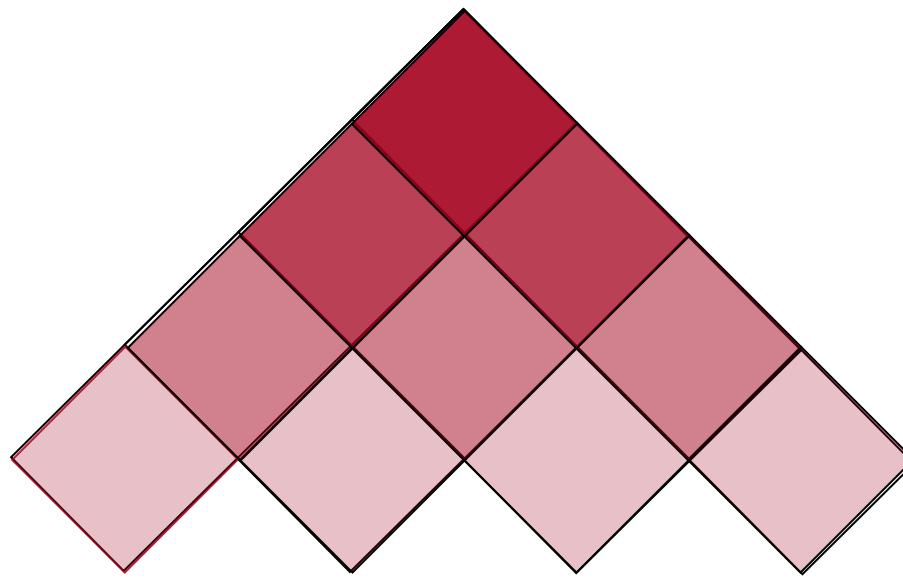


**PCFG**

**Rule Prob  $\theta_i$**

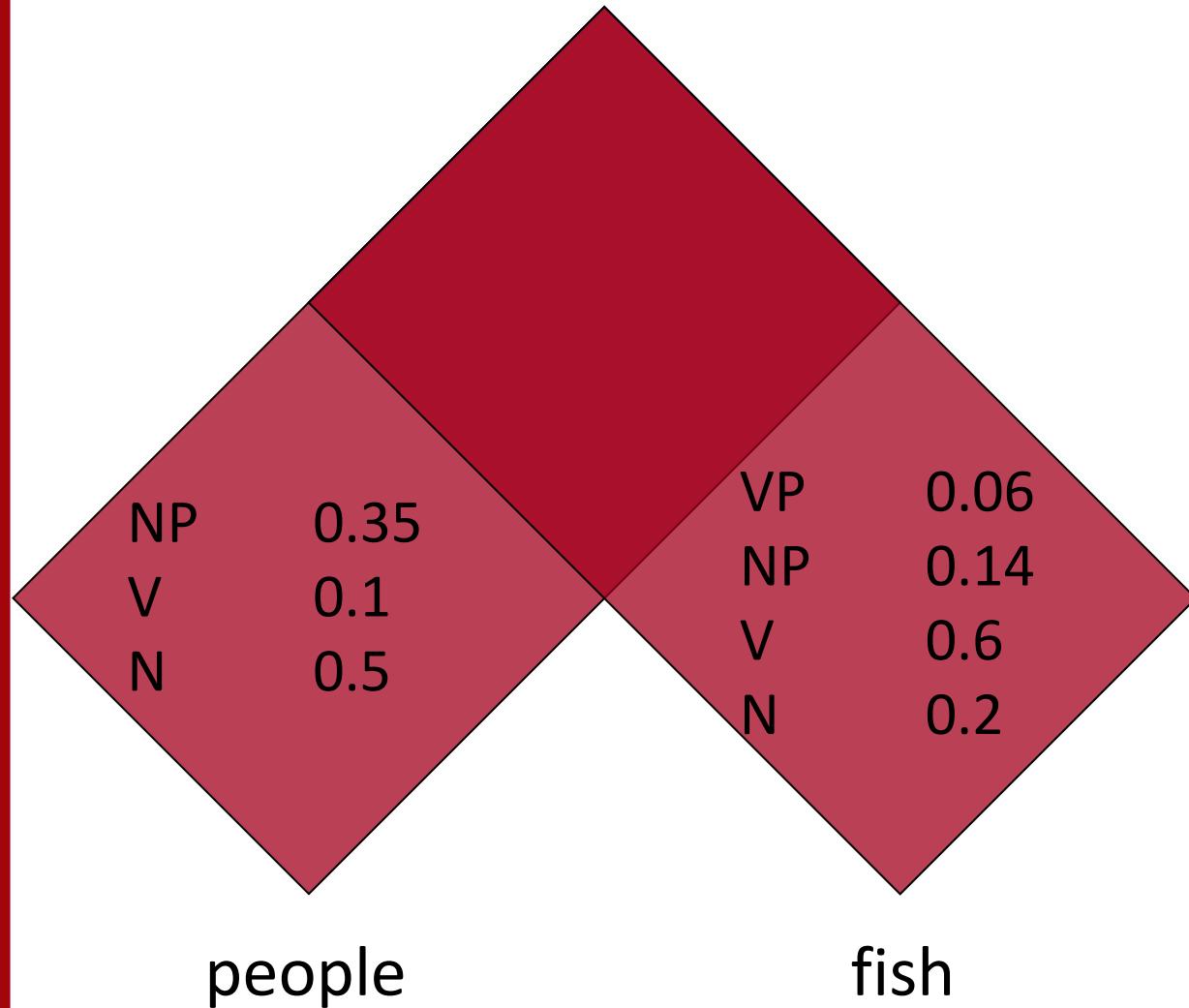
|                         |               |
|-------------------------|---------------|
| $S \rightarrow NP\ VP$  | $\theta_0$    |
| $NP \rightarrow NP\ NP$ | $\theta_1$    |
| ...                     |               |
| $N \rightarrow fish$    | $\theta_{42}$ |
| $N \rightarrow people$  | $\theta_{43}$ |
| $V \rightarrow fish$    | $\theta_{44}$ |
| ...                     |               |

# Cocke-Kasami-Younger (CKY) Constituency Parsing



fish people fish tanks

# Viterbi (Max) Scores



|                            |     |
|----------------------------|-----|
| $S \rightarrow NP VP$      | 0.9 |
| $S \rightarrow VP$         | 0.1 |
| $VP \rightarrow V NP$      | 0.5 |
| $VP \rightarrow V$         | 0.1 |
| $VP \rightarrow V @VP\_V$  | 0.3 |
| $VP \rightarrow V PP$      | 0.1 |
| $@VP\_V \rightarrow NP PP$ | 1.0 |
| $NP \rightarrow NP NP$     | 0.1 |
| $NP \rightarrow NP PP$     | 0.2 |
| $NP \rightarrow N$         | 0.7 |
| $PP \rightarrow P NP$      | 1.0 |

# Extended CKY parsing

- Unaries can be incorporated into the algorithm
  - Messy, but doesn't increase algorithmic complexity
- Empties can be incorporated
  - Use fenceposts
  - Doesn't increase complexity; essentially like unaries
- Binarization is *vital*
  - Without binarization, you don't get parsing cubic in the length of the sentence and in the number of nonterminals in the grammar
    - Binarization may be an explicit transformation or implicit in how the parser works (Early-style dotted rules), but it's always there.

# A Recursive Parser

```
bestScore(X,i,j,s)
if (j == i)
    return q(X->s[i])
else
    return max q(X->YZ) *
                k,X->YZ      bestScore(Y,i,k,s) *
                                bestScore(Z,k+1,j,s)
```

# The CKY algorithm (1960/1965)

## ... extended to unaries

```
function CKY(words, grammar) returns [most_probable_parse, prob]
    score = new double[#{words}+1][#{words}+1][#{nonterms}]
    back = new Pair[#{words}+1][#{words}+1][#{nonterms}]
    for i=0; i<#{words}; i++
        for A in nonterms
            if A -> words[i] in grammar
                score[i][i+1][A] = P(A -> words[i])
    //handle unaries
    boolean added = true
    while added
        added = false
        for A, B in nonterms
            if score[i][i+1][B] > 0 && A->B in grammar
                prob = P(A->B)*score[i][i+1][B]
                if prob > score[i][i+1][A]
                    score[i][i+1][A] = prob
                    back[i][i+1][A] = B
                    added = true
```

# The CKY algorithm (1960/1965)

## ... extended to unaries

```
for span = 2 to #(words)
    for begin = 0 to #(words)- span
        end = begin + span
        for split = begin+1 to end-1
            for A,B,C in nonterms
                prob=score[begin][split][B]*score[split][end][C]*P(A->BC)
                if prob > score[begin][end][A]
                    score[begin][end][A] = prob
                    back[begin][end][A] = new Triple(split,B,C)
//handle unaries
boolean added = true
while added
    added = false
    for A, B in nonterms
        prob = P(A->B)*score[begin][end][B];
        if prob > score[begin][end][A]
            score[begin][end][A] = prob
            back[begin][end][A] = B
            added = true
return buildTree(score, back)
```

# The grammar: Binary, no epsilons,

|                            |     |                        |     |
|----------------------------|-----|------------------------|-----|
| $S \rightarrow NP VP$      | 0.9 | $N \rightarrow people$ | 0.5 |
| $S \rightarrow VP$         | 0.1 | $N \rightarrow fish$   | 0.2 |
| $VP \rightarrow V NP$      | 0.5 | $N \rightarrow tanks$  | 0.2 |
| $VP \rightarrow V$         | 0.1 | $N \rightarrow rods$   | 0.1 |
| $VP \rightarrow V @VP\_V$  | 0.3 | $V \rightarrow people$ | 0.1 |
| $VP \rightarrow V PP$      | 0.1 | $V \rightarrow fish$   | 0.6 |
| $@VP\_V \rightarrow NP PP$ | 1.0 | $V \rightarrow tanks$  | 0.3 |
| $NP \rightarrow NP NP$     | 0.1 | $P \rightarrow with$   | 1.0 |
| $NP \rightarrow NP PP$     | 0.2 |                        |     |
| $NP \rightarrow N$         | 0.7 |                        |     |
| $PP \rightarrow P NP$      | 1.0 |                        |     |

|   | fish        | 1 | people      | 2 | fish        | 3           | tanks       | 4 |
|---|-------------|---|-------------|---|-------------|-------------|-------------|---|
| 0 | score[0][1] |   | score[0][2] |   | score[0][3] |             | score[0][4] |   |
| 1 |             |   |             |   |             |             |             |   |
| 2 |             |   | score[1][2] |   | score[1][3] |             | score[1][4] |   |
| 3 |             |   |             |   | score[2][3] |             | score[2][4] |   |
| 4 |             |   |             |   |             | score[3][4] |             |   |

|                           |     |
|---------------------------|-----|
| $S \rightarrow NP VP$     | 0.9 |
| $S \rightarrow VP$        | 0.1 |
| $VP \rightarrow V NP$     | 0.5 |
| $VP \rightarrow V$        | 0.1 |
| $VP \rightarrow V @VP_V$  | 0.3 |
| $VP \rightarrow V PP$     | 0.1 |
| $@VP_V \rightarrow NP PP$ | 1.0 |
| $NP \rightarrow NP NP$    | 0.1 |
| $NP \rightarrow NP PP$    | 0.2 |
| $NP \rightarrow N$        | 0.7 |
| $PP \rightarrow P NP$     | 1.0 |
| <br>                      |     |
| $N \rightarrow people$    | 0.5 |
| $N \rightarrow fish$      | 0.2 |
| $N \rightarrow tanks$     | 0.2 |
| $N \rightarrow rods$      | 0.1 |
| $V \rightarrow people$    | 0.1 |
| $V \rightarrow fish$      | 0.6 |
| $V \rightarrow tanks$     | 0.3 |
| $P \rightarrow with$      | 1.0 |



|               |     | fish  | people                           | fish                         | tanks |   |
|---------------|-----|---|----------------------------------|------------------------------|-------|---|
|               | 0   | N → fish 0.2<br>V → fish 0.6  |                                  |                              |       | 4 |
| S → NP VP     | 0.9 |   |                                  |                              |       |   |
| S → VP        | 0.1 |   |                                  |                              |       |   |
| VP → V NP     | 0.5 |   |                                  |                              |       |   |
| VP → V        | 0.1 |   |                                  |                              |       |   |
| VP → V @VP_V  | 0.3 |   |                                  |                              |       |   |
| VP → V PP     | 0.1 |   |                                  |                              |       |   |
| @VP_V → NP PP | 1.0 |   | N → people 0.5<br>V → people 0.1 |                              |       |   |
| NP → NP NP    | 0.1 |   |                                  |                              |       |   |
| NP → NP PP    | 0.2 |   |                                  |                              |       |   |
| NP → N        | 0.7 |   |                                  |                              |       |   |
| PP → P NP     | 1.0 |   |                                  | N → fish 0.2<br>V → fish 0.6 |       |   |
| N → people    | 0.5 |   |                                  |                              |       |   |
| N → fish      | 0.2 | // handle unaries<br>boolean added = true<br>while added<br>added = false<br>for A, B in nonterms<br>if score[i][i+1][B] > 0 && A->B in grammar<br>prob = P(A->B)*score[i][i+1][B]<br>if(prob > score[i][i+1][A])<br>score[i][i+1][A] = prob<br>back[i][i+1][A] = B<br>added = true |                                  |                              |       |   |
| N → tanks     | 0.2 |   |                                  |                              |       |   |
| N → rods      | 0.1 |   |                                  |                              |       |   |
| V → people    | 0.1 |   |                                  |                              |       |   |
| V → fish      | 0.6 |   |                                  |                              |       |   |
| V → tanks     | 0.3 |   |                                  |                              |       |   |
| P → with      | 1.0 |   |                                  |                              |       |   |

|                   |     | fish  | people   | fish   | tanks |   |
|-------------------|-----|---|--|--|-------|---|
|                   | 0   | N → fish 0.2<br>V → fish 0.6<br>NP → N 0.14<br>VP → V 0.06<br>S → VP 0.006  |  |  |       | 4 |
| S → NP VP         | 0.9 |   |  |  |       |   |
| S → VP            | 0.1 |   |  |  |       |   |
| VP → V NP         | 0.5 |   |  |  |       |   |
| VP → V            | 0.1 |   |  |  |       |   |
| VP → V @VP_V      | 0.3 |   |  |  |       |   |
| VP → V PP         | 0.1 |   | N → people 0.5<br>V → people 0.1<br>NP → N 0.35<br>VP → V 0.01<br>S → VP 0.001 |  |       |   |
| @VP_V → NP PP     | 1.0 |   |  |  |       |   |
| NP → NP NP        | 0.1 |   |  |  |       |   |
| NP → NP PP        | 0.2 |   |  |  |       |   |
| NP → N            | 0.7 |   |  |  |       |   |
| PP → P NP         | 1.0 |   |  | N → fish 0.2<br>V → fish 0.6<br>NP → N 0.14<br>VP → V 0.06<br>S → VP 0.006 |       |   |
| N → <i>people</i> | 0.5 |   |  |  |       |   |
| N → <i>fish</i>   | 0.2 |   |  |  |       |   |
| N → <i>tanks</i>  | 0.2 |   |  |  |       |   |
| N → <i>rods</i>   | 0.1 | prob=score[begin][split][B]*score[split][end][C]*P(A->BC)<br>if (prob > score[begin][end][A])<br>score[begin][end][A] = prob<br>back[begin][end][A] = new Triple(split,B,C) |  |  |       |   |
| V → <i>people</i> | 0.1 |   |  |  |       |   |
| V → <i>fish</i>   | 0.6 | 4   |  |  |       |   |
| V → <i>tanks</i>  | 0.3 |   |  |  |       |   |
| P → <i>with</i>   | 1.0 |   |  |  |       |   |

|                            |     | fish  | 1  | people  | 2  | fish  | 3  | tanks | 4 |
|----------------------------|-----|---|--|---|--|---|--|-------|---|
| $S \rightarrow NP VP$      | 0.9 |   |  |   |  |   |  |       |   |
| $S \rightarrow VP$         | 0.1 |   |  |   |  |   |  |       |   |
| $VP \rightarrow V NP$      | 0.5 |   |  |   |  |   |  |       |   |
| $VP \rightarrow V$         | 0.1 |   |  |   |  |   |  |       |   |
| $VP \rightarrow V @VP\_V$  | 0.3 |   |  |   |  |   |  |       |   |
| $VP \rightarrow V PP$      | 0.1 |   |  |   |  |   |  |       |   |
| $@VP\_V \rightarrow NP PP$ | 1.0 |   |  |   |  |   |  |       |   |
| $NP \rightarrow NP NP$     | 0.1 |   |  |   |  |   |  |       |   |
| $NP \rightarrow NP PP$     | 0.2 |   |  |   |  |   |  |       |   |
| $NP \rightarrow N$         | 0.7 |   |  |   |  |   |  |       |   |
| $PP \rightarrow P NP$      | 1.0 |   |  |   |  |   |  |       |   |
| $N \rightarrow people$     | 0.5 |   |  |   |  |   |  |       |   |
| $N \rightarrow fish$       | 0.2 |   |  |   |  |   |  |       |   |
| $N \rightarrow tanks$      | 0.2 |   |  |   |  |   |  |       |   |
| $N \rightarrow rods$       | 0.1 |   |  |   |  |   |  |       |   |
| $V \rightarrow people$     | 0.1 |   |  |   |  |   |  |       |   |
| $V \rightarrow fish$       | 0.6 |   |  |   |  |   |  |       |   |
| $V \rightarrow tanks$      | 0.3 |   |  |   |  |   |  |       |   |
| $P \rightarrow with$       | 1.0 |   |  |   |  |   |  |       |   |
|                            | 0   | $N \rightarrow fish 0.2$<br>$V \rightarrow fish 0.6$<br>$NP \rightarrow N 0.14$<br>$VP \rightarrow V 0.06$<br>$S \rightarrow VP 0.006$  | $NP \rightarrow NP NP$<br>0.0049<br>$VP \rightarrow V NP$<br>0.105<br>$S \rightarrow NP VP$<br>0.00126                                     |   |  |   |  |       |   |
|                            | 1   |   | $N \rightarrow people 0.5$<br>$V \rightarrow people 0.1$<br>$NP \rightarrow N 0.35$<br>$VP \rightarrow V 0.01$<br>$S \rightarrow VP 0.001$ | $NP \rightarrow NP NP$<br>0.0049<br>$VP \rightarrow V NP$<br>0.007<br>$S \rightarrow NP VP$<br>0.0189 |  |   |  |       |   |
|                            | 2   |   |  |   | $N \rightarrow fish 0.2$<br>$V \rightarrow fish 0.6$<br>$NP \rightarrow N 0.14$<br>$VP \rightarrow V 0.06$<br>$S \rightarrow VP 0.006$ | $NP \rightarrow NP NP$<br>0.00196<br>$VP \rightarrow V NP$<br>0.042<br>$S \rightarrow NP VP$<br>0.00378 |  |       |   |
|                            | 3   | <pre>//handle unaries boolean added = true while added     added = false     for A, B in nonterms         prob = P(A-&gt;B)*score[begin][end][B];         if prob &gt; score[begin][end][A]             score[begin][end][A] = prob             back[begin][end][A] = B             added = true         </pre> |  |   |  |   | $N \rightarrow tanks 0.2$<br>$V \rightarrow tanks 0.1$<br>$NP \rightarrow N 0.14$<br>$VP \rightarrow V 0.03$<br>$S \rightarrow VP 0.003$ |       |   |
|                            | 4   |   |  |   |  |   |  |       |   |

|                            |     | fish                     | 1                                | people                           | 2                                   | fish                            | 3                                   | tanks                     | 4 |
|----------------------------|-----|--------------------------|----------------------------------|----------------------------------|-------------------------------------|---------------------------------|-------------------------------------|---------------------------|---|
| $S \rightarrow NP VP$      | 0.9 |                          |                                  |                                  |                                     |                                 |                                     |                           |   |
| $S \rightarrow VP$         | 0.1 |                          |                                  |                                  |                                     |                                 |                                     |                           |   |
| $VP \rightarrow V NP$      | 0.5 |                          |                                  |                                  |                                     |                                 |                                     |                           |   |
| $VP \rightarrow V$         | 0.1 |                          |                                  |                                  |                                     |                                 |                                     |                           |   |
| $VP \rightarrow V @VP\_V$  | 0.3 |                          |                                  |                                  |                                     |                                 |                                     |                           |   |
| $VP \rightarrow V PP$      | 0.1 |                          |                                  |                                  |                                     |                                 |                                     |                           |   |
| $@VP\_V \rightarrow NP PP$ | 1.0 |                          |                                  |                                  |                                     |                                 |                                     |                           |   |
| $NP \rightarrow NP NP$     | 0.1 |                          |                                  |                                  |                                     |                                 |                                     |                           |   |
| $NP \rightarrow NP PP$     | 0.2 |                          |                                  |                                  |                                     |                                 |                                     |                           |   |
| $NP \rightarrow N$         | 0.7 |                          |                                  |                                  |                                     |                                 |                                     |                           |   |
| $PP \rightarrow P NP$      | 1.0 |                          |                                  |                                  |                                     |                                 |                                     |                           |   |
| $N \rightarrow people$     | 0.5 |                          |                                  |                                  |                                     |                                 |                                     |                           |   |
| $N \rightarrow fish$       | 0.2 |                          |                                  |                                  |                                     |                                 |                                     |                           |   |
| $N \rightarrow tanks$      | 0.2 |                          |                                  |                                  |                                     |                                 |                                     |                           |   |
| $N \rightarrow rods$       | 0.1 |                          |                                  |                                  |                                     |                                 |                                     |                           |   |
| $V \rightarrow people$     | 0.1 |                          |                                  |                                  |                                     |                                 |                                     |                           |   |
| $V \rightarrow fish$       | 0.6 |                          |                                  |                                  |                                     |                                 |                                     |                           |   |
| $V \rightarrow tanks$      | 0.3 |                          |                                  |                                  |                                     |                                 |                                     |                           |   |
| $P \rightarrow with$       | 1.0 |                          |                                  |                                  |                                     |                                 |                                     |                           |   |
|                            | 0   | $N \rightarrow fish 0.2$ | $NP \rightarrow NP NP$<br>0.0049 | $VP \rightarrow V NP$<br>0.105   | $S \rightarrow VP$<br><b>0.0105</b> |                                 |                                     |                           |   |
|                            | 1   | $V \rightarrow fish 0.6$ |                                  |                                  |                                     |                                 |                                     |                           |   |
|                            | 2   | $NP \rightarrow N 0.14$  |                                  |                                  |                                     |                                 |                                     |                           |   |
|                            | 3   | $VP \rightarrow V 0.06$  |                                  |                                  |                                     |                                 |                                     |                           |   |
|                            | 4   | $S \rightarrow VP 0.006$ |                                  |                                  |                                     |                                 |                                     |                           |   |
|                            | 0   |                          | $N \rightarrow people 0.5$       | $NP \rightarrow NP NP$<br>0.0049 | $VP \rightarrow V NP$<br>0.007      | $S \rightarrow NP VP$<br>0.0189 |                                     |                           |   |
|                            | 1   |                          | $V \rightarrow people 0.1$       |                                  |                                     |                                 |                                     |                           |   |
|                            | 2   |                          | $NP \rightarrow N 0.35$          |                                  |                                     |                                 |                                     |                           |   |
|                            | 3   |                          | $VP \rightarrow V 0.01$          |                                  |                                     |                                 |                                     |                           |   |
|                            | 4   |                          | $S \rightarrow VP 0.001$         |                                  |                                     |                                 |                                     |                           |   |
|                            | 0   |                          |                                  | $N \rightarrow fish 0.2$         | $NP \rightarrow NP NP$<br>0.00196   | $VP \rightarrow V NP$<br>0.042  | $S \rightarrow VP$<br><b>0.0042</b> |                           |   |
|                            | 1   |                          |                                  | $V \rightarrow fish 0.6$         |                                     |                                 |                                     |                           |   |
|                            | 2   |                          |                                  | $NP \rightarrow N 0.14$          |                                     |                                 |                                     |                           |   |
|                            | 3   |                          |                                  | $VP \rightarrow V 0.06$          |                                     |                                 |                                     |                           |   |
|                            | 4   |                          |                                  | $S \rightarrow VP 0.006$         |                                     |                                 |                                     |                           |   |
|                            | 0   |                          |                                  |                                  |                                     |                                 |                                     | $N \rightarrow tanks 0.2$ |   |
|                            | 1   |                          |                                  |                                  |                                     |                                 |                                     | $V \rightarrow tanks 0.1$ |   |
|                            | 2   |                          |                                  |                                  |                                     |                                 |                                     | $NP \rightarrow N 0.14$   |   |
|                            | 3   |                          |                                  |                                  |                                     |                                 |                                     | $VP \rightarrow V 0.03$   |   |
|                            | 4   |                          |                                  |                                  |                                     |                                 |                                     | $S \rightarrow VP 0.003$  |   |

```

for split = begin+1 to end-1
  for A,B,C in nonterms
    prob=score[begin][split][B]*score[split][end][C]*P(A->BC)
    if prob > score[begin][end][A]
      score[begin][end][A] = prob
      back[begin][end][A] = new Triple(split,B,C)
  
```

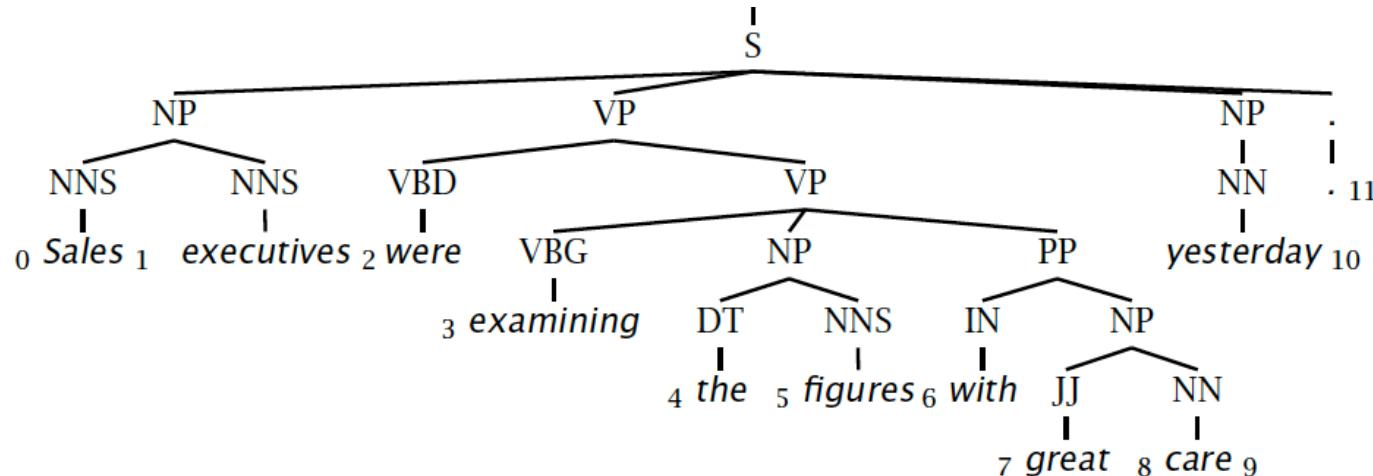
|                            |     | fish   | 1                          | people                 | 2 | fish | 3 | tanks                     | 4 |
|----------------------------|-----|--|----------------------------|------------------------|---|------|---|---------------------------|---|
| $S \rightarrow NP VP$      | 0.9 |  |                            |                        |   |      |   |                           |   |
| $S \rightarrow VP$         | 0.1 |  |                            |                        |   |      |   |                           |   |
| $VP \rightarrow V NP$      | 0.5 |  |                            |                        |   |      |   |                           |   |
| $VP \rightarrow V$         | 0.1 |  |                            |                        |   |      |   |                           |   |
| $VP \rightarrow V @VP\_V$  | 0.3 |  |                            |                        |   |      |   |                           |   |
| $VP \rightarrow V PP$      | 0.1 |  |                            |                        |   |      |   |                           |   |
| $@VP\_V \rightarrow NP PP$ | 1.0 |  |                            |                        |   |      |   |                           |   |
| $NP \rightarrow NP NP$     | 0.1 |  |                            |                        |   |      |   |                           |   |
| $NP \rightarrow NP PP$     | 0.2 |  |                            |                        |   |      |   |                           |   |
| $NP \rightarrow N$         | 0.7 |  |                            |                        |   |      |   |                           |   |
| $PP \rightarrow P NP$      | 1.0 |  |                            |                        |   |      |   |                           |   |
| $N \rightarrow people$     | 0.5 |  |                            |                        |   |      |   |                           |   |
| $N \rightarrow fish$       | 0.2 |  |                            |                        |   |      |   |                           |   |
| $N \rightarrow tanks$      | 0.2 |  |                            |                        |   |      |   |                           |   |
| $N \rightarrow rods$       | 0.1 |  |                            |                        |   |      |   |                           |   |
| $V \rightarrow people$     | 0.1 |  |                            |                        |   |      |   |                           |   |
| $V \rightarrow fish$       | 0.6 |  |                            |                        |   |      |   |                           |   |
| $V \rightarrow tanks$      | 0.3 |  |                            |                        |   |      |   |                           |   |
| $P \rightarrow with$       | 1.0 |  |                            |                        |   |      |   |                           |   |
|                            | 0   | $N \rightarrow fish 0.2$   | $NP \rightarrow NP NP$     | $NP \rightarrow NP NP$ |   |      |   |                           |   |
|                            |     | $V \rightarrow fish 0.6$   | $0.0049$                   | $0.0000686$            |   |      |   |                           |   |
|                            |     | $NP \rightarrow N 0.14$  | $VP \rightarrow V NP$      | $VP \rightarrow V NP$  |   |      |   |                           |   |
|                            |     | $VP \rightarrow V 0.06$  | $0.105$                    | $0.00147$              |   |      |   |                           |   |
|                            | 1   | $S \rightarrow VP 0.006$   | $S \rightarrow VP$         | $S \rightarrow NP VP$  |   |      |   |                           |   |
|                            |     |  | $0.0105$                   | $0.000882$             |   |      |   |                           |   |
|                            | 2   |  | $N \rightarrow people 0.5$ | $NP \rightarrow NP NP$ |   |      |   |                           |   |
|                            |     |  | $V \rightarrow people 0.1$ | $0.0049$               |   |      |   |                           |   |
|                            |     |  | $NP \rightarrow N 0.35$    | $VP \rightarrow V NP$  |   |      |   |                           |   |
|                            |     |  | $VP \rightarrow V 0.01$    | $0.007$                |   |      |   |                           |   |
|                            | 3   |  | $S \rightarrow VP 0.001$   | $S \rightarrow NP VP$  |   |      |   |                           |   |
|                            |     |  |                            | $0.0189$               |   |      |   |                           |   |
|                            | 4   | for split = begin+1 to end-1<br>for A,B,C in nonterms<br>prob=score[begin][split][B]*score[split][end][C]*P(A->BC)<br>if prob > score[begin][end][A]<br>score[begin][end][A] = prob<br>back[begin][end][A] = new Triple(split,B,C) |                            |                        |   |      |   | $N \rightarrow tanks 0.2$ |   |
|                            |     |  |                            |                        |   |      |   | $V \rightarrow tanks 0.1$ |   |
|                            |     |  |                            |                        |   |      |   | $NP \rightarrow N 0.14$   |   |
|                            |     |  |                            |                        |   |      |   | $VP \rightarrow V 0.03$   |   |
|                            |     |  |                            |                        |   |      |   | $S \rightarrow VP 0.003$  |   |



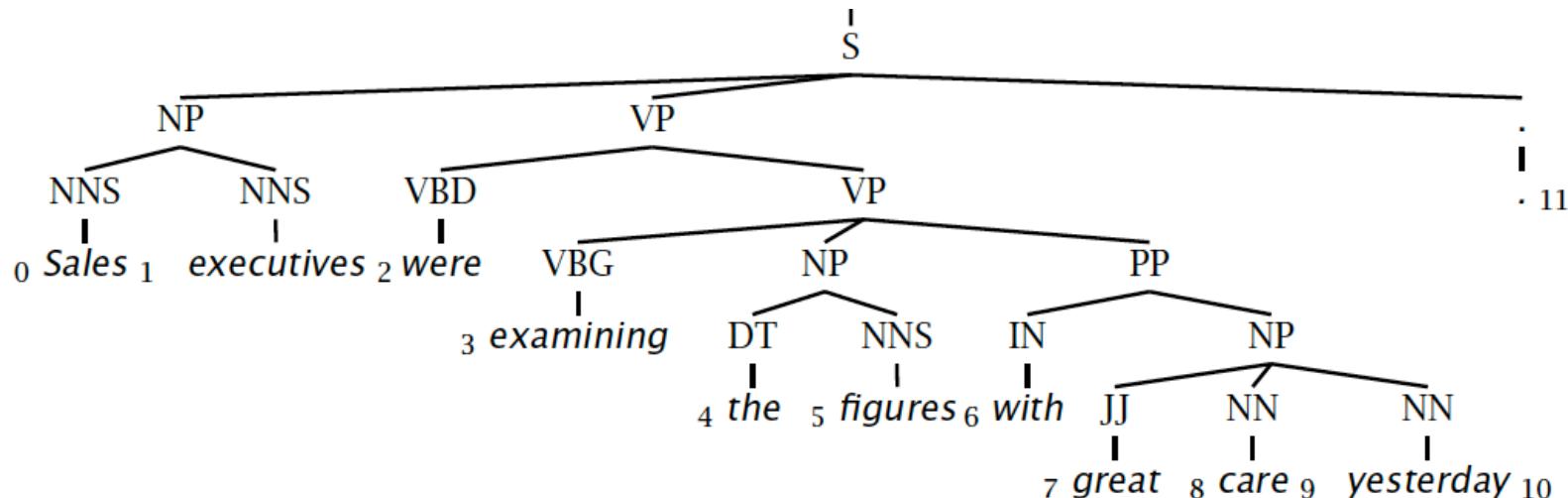


# Evaluating constituency parsing

Gold standard brackets: S-(0:11), NP-(0:2), VP-(2:9), VP-(3:9), NP-(4:6), PP-(6-9), NP-(7,9), NP-(9:10)



Candidate brackets: S-(0:11), NP-(0:2), VP-(2:10), VP-(3:10), NP-(4:6), PP-(6-10), NP-(7,10)



# Evaluating constituency parsing

**Gold standard brackets:**

S-(0:11), NP-(0:2), VP-(2:9), VP-(3:9), NP-(4:6), PP-(6-9), NP-(7,9), NP-(9:10)

**Candidate brackets:**

S-(0:11), NP-(0:2), VP-(2:10), VP-(3:10), NP-(4:6), PP-(6-10), NP-(7,10)

Labeled Precision             $3/7 = 42.9\%$

Labeled Recall                 $3/8 = 37.5\%$

LP/LR F1                      40.0%

Tagging Accuracy              $11/11 = 100.0\%$

# How good are PCFGs?

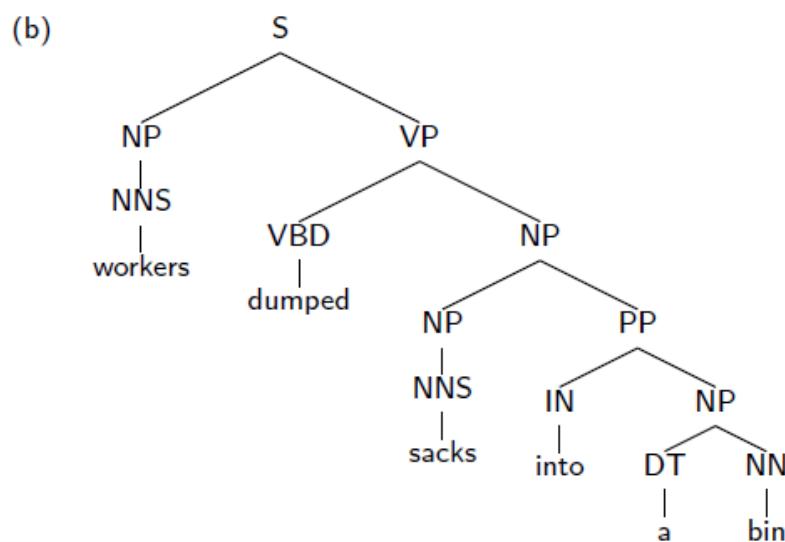
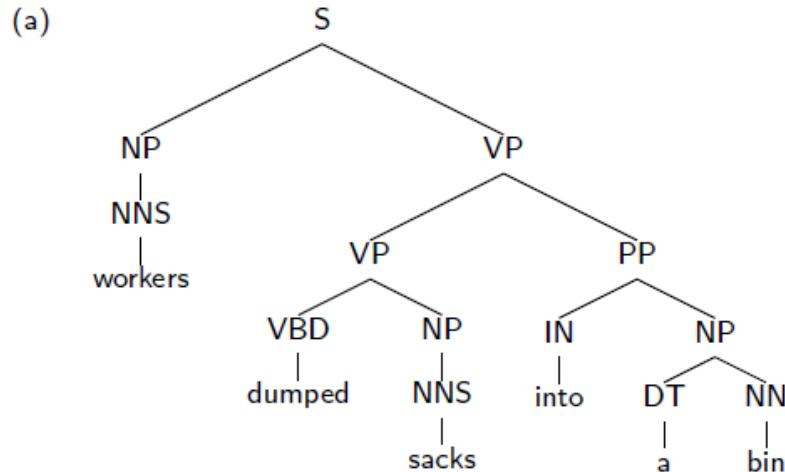
- Penn WSJ parsing accuracy: about 73% LP/LR F1
- Robust
  - Usually admit everything, but with low probability
- Partial solution for grammar ambiguity
  - A PCFG gives some idea of the plausibility of a parse
  - But not so good because the independence assumptions are too strong
- Give a probabilistic language model
  - But in the simple case it performs worse than a trigram model
- The problem seems to be that PCFGs lack the lexicalization of a trigram model

## Weaknesses of PCFGs

# Weaknesses

- Lack of sensitivity to structural frequencies
- Lack of sensitivity to lexical information
- (A word is independent of the rest of the tree given its POS!)

# A Case of PP Attachment Ambiguity



| Rules                                     |
|---|
| $S \rightarrow NP\ VP$                    |
| $NP \rightarrow NNS$                      |
| <b><math>VP \rightarrow VP\ PP</math></b> |
| $VP \rightarrow VBD\ NP$                  |
| $NP \rightarrow NNS$                      |
| $PP \rightarrow IN\ NP$                   |
| $NP \rightarrow DT\ NN$                   |
| $NNS \rightarrow workers$                 |
| $VBD \rightarrow dumped$                  |
| $NNS \rightarrow sacks$                   |
| $IN \rightarrow into$                     |
| $DT \rightarrow a$                        |
| $NN \rightarrow bin$                      |

(a)

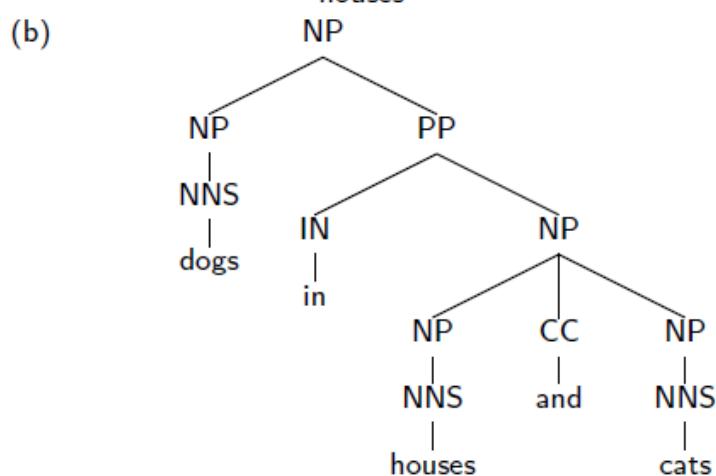
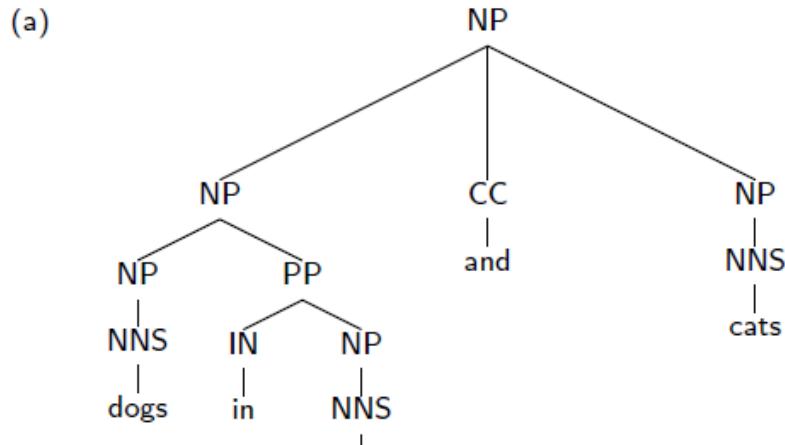
| Rules                                     |
|---|
| $S \rightarrow NP\ VP$                    |
| $NP \rightarrow NNS$                      |
| <b><math>NP \rightarrow NP\ PP</math></b> |
| $VP \rightarrow VBD\ NP$                  |
| $NP \rightarrow NNS$                      |
| $PP \rightarrow IN\ NP$                   |
| $NP \rightarrow DT\ NN$                   |
| $NNS \rightarrow workers$                 |
| $VBD \rightarrow dumped$                  |
| $NNS \rightarrow sacks$                   |
| $IN \rightarrow into$                     |
| $DT \rightarrow a$                        |
| $NN \rightarrow bin$                      |

(b)

If  $q(NP \rightarrow NP\ PP) > q(VP \rightarrow VP\ PP)$  then (b) is more probable, else (a) is more probable.

**Attachment decision is completely independent of the words**

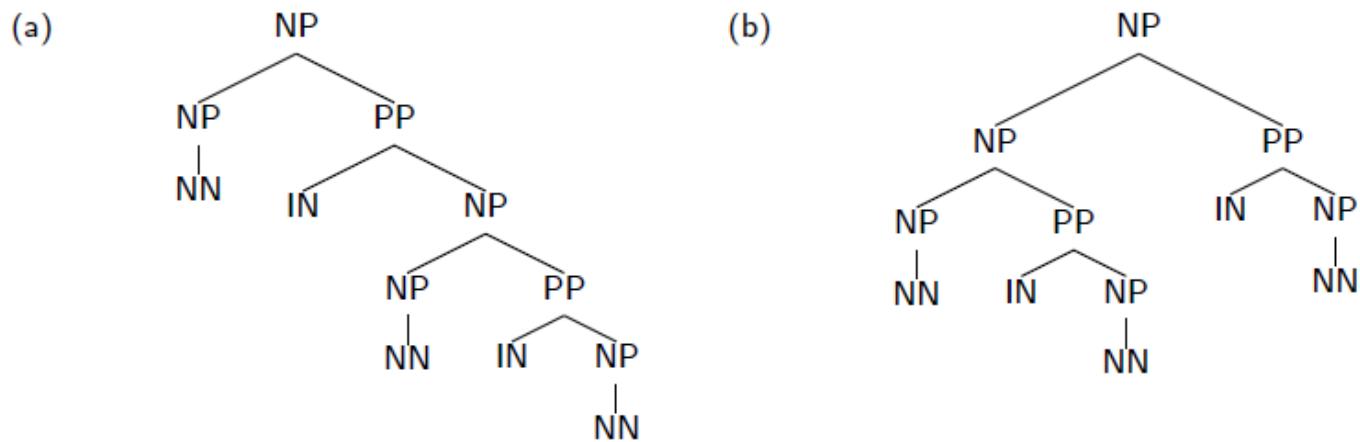
# A Case of Coordination Ambiguity



| Rules  | Rules  |
|--|--|
| $\text{NP} \rightarrow \text{NP CC NP}$<br>$\text{NP} \rightarrow \text{NP PP}$<br>$\text{NP} \rightarrow \text{NNS}$<br>$\text{PP} \rightarrow \text{IN NP}$<br>$\text{NP} \rightarrow \text{NNS}$<br>$\text{NP} \rightarrow \text{NNS}$<br>$\text{NNS} \rightarrow \text{dogs}$<br>$\text{IN} \rightarrow \text{in}$<br>$\text{NNS} \rightarrow \text{houses}$<br>$\text{CC} \rightarrow \text{and}$<br>$\text{NNS} \rightarrow \text{cats}$ | $\text{NP} \rightarrow \text{NP CC NP}$<br>$\text{NP} \rightarrow \text{NP PP}$<br>$\text{NP} \rightarrow \text{NNS}$<br>$\text{PP} \rightarrow \text{IN NP}$<br>$\text{NP} \rightarrow \text{NNS}$<br>$\text{NP} \rightarrow \text{NNS}$<br>$\text{NNS} \rightarrow \text{dogs}$<br>$\text{IN} \rightarrow \text{in}$<br>$\text{NNS} \rightarrow \text{houses}$<br>$\text{CC} \rightarrow \text{and}$<br>$\text{NNS} \rightarrow \text{cats}$ |

Here the two parses have identical rules, and therefore have identical probability under any assignment of PCFG rule probabilities

# Structural Preferences: Close Attachment



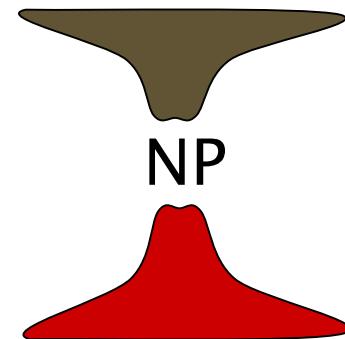
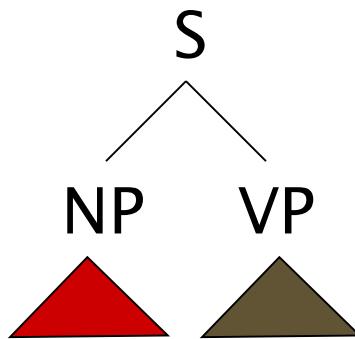
- ▶ Example: president of a company in Africa
- ▶ Both parses have the same rules, therefore receive same probability under a PCFG
- ▶ “Close attachment” (structure (a)) is twice as likely in Wall Street Journal text.

# Structural Preferences: Close Attachment

- **Example:** John was believed to have been shot by Bill
- Low attachment analysis (Bill does the shooting) contains same rules as high attachment analysis (Bill does the believing)
  - Two analyses receive the same probability

# PCFGs and Independence

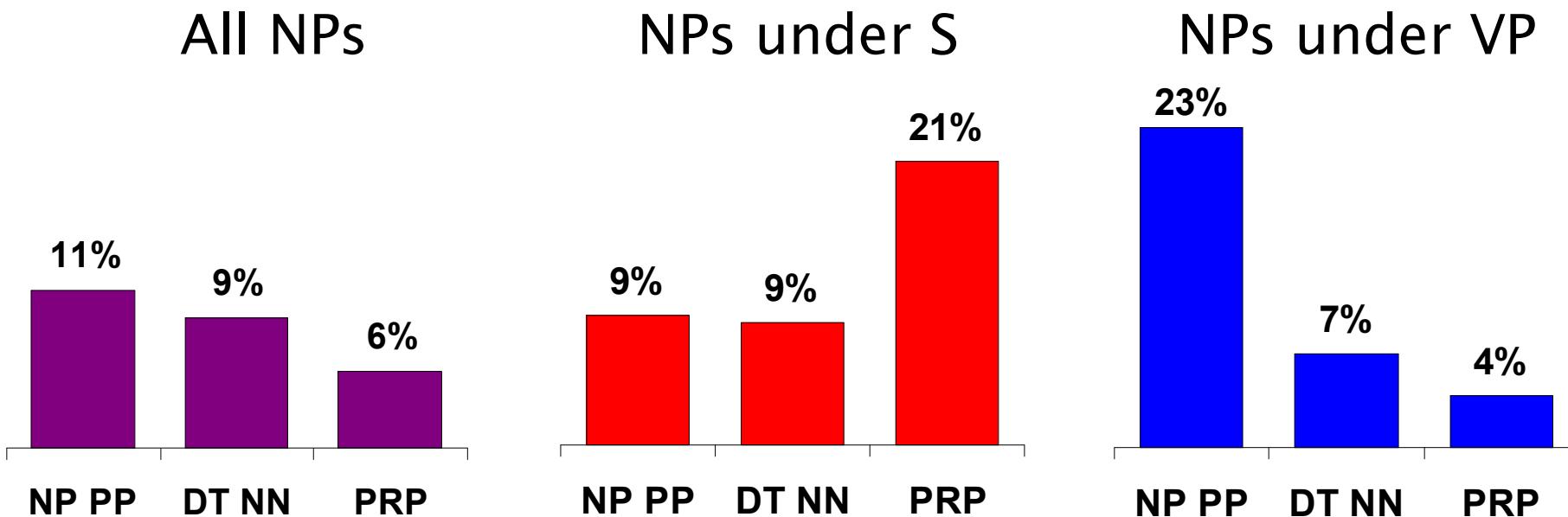
- The symbols in a PCFG define independence assumptions:

$$\begin{aligned} S &\rightarrow NP \ VP \\ NP &\rightarrow DT \ NN \end{aligned}$$


- At any node, **the material inside that node** is independent of the material outside that node, given the label of that node
- Any information that statistically connects behavior **inside** and outside a node must flow through that node's label

# Non-Independence I

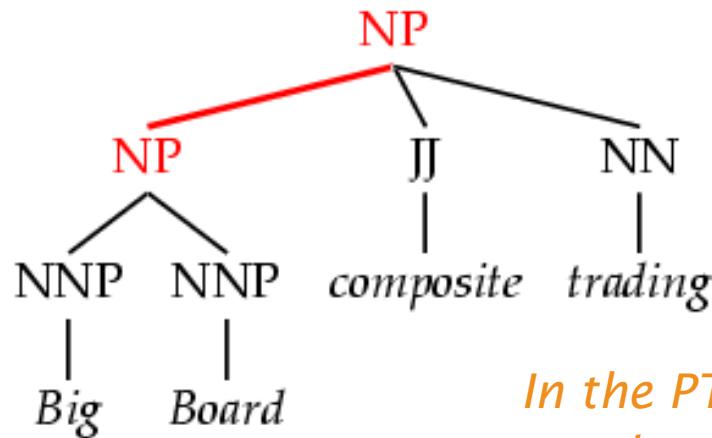
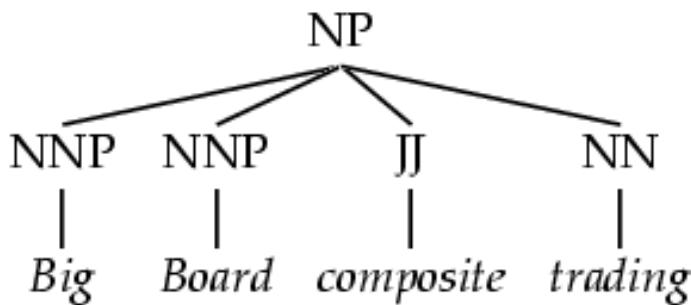
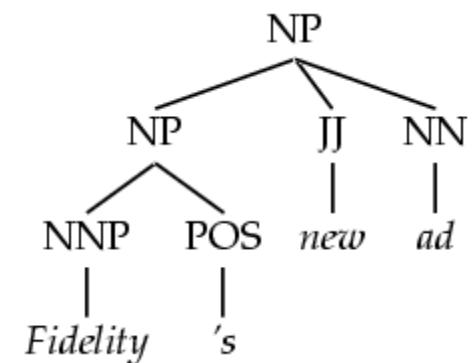
- The independence assumptions of a PCFG are often too strong



- Example: the expansion of an NP is highly dependent on the parent of the NP (i.e., subjects vs. objects)

## Non-Independence II

- Symptoms of overly strong assumptions:
  - Rewrites get used where they don't belong

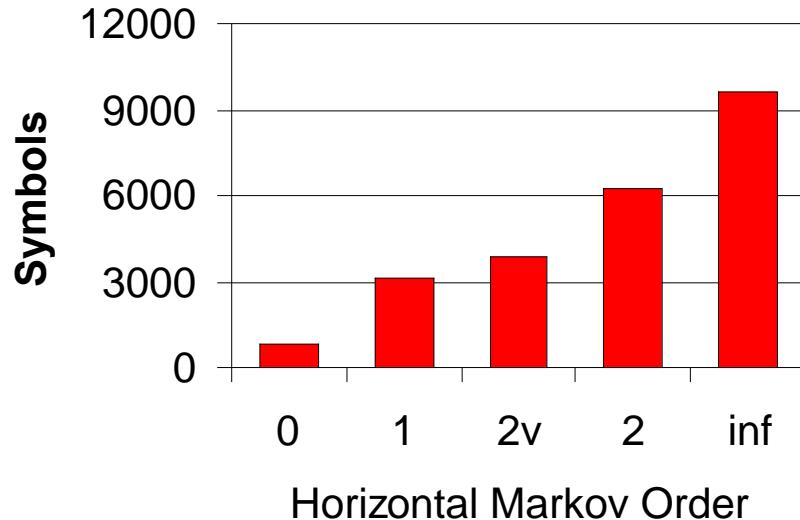
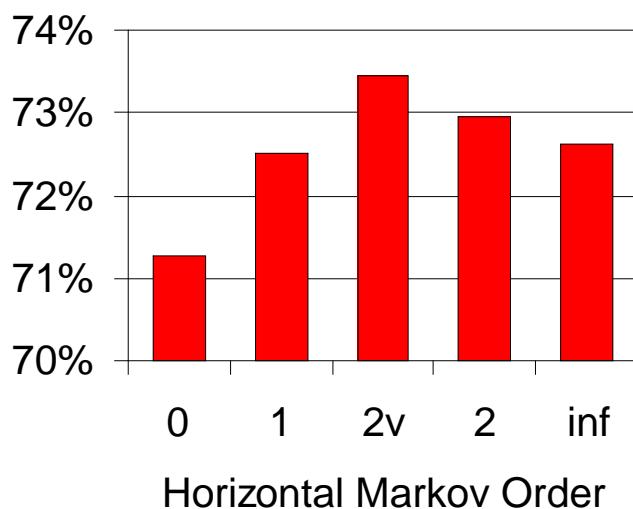
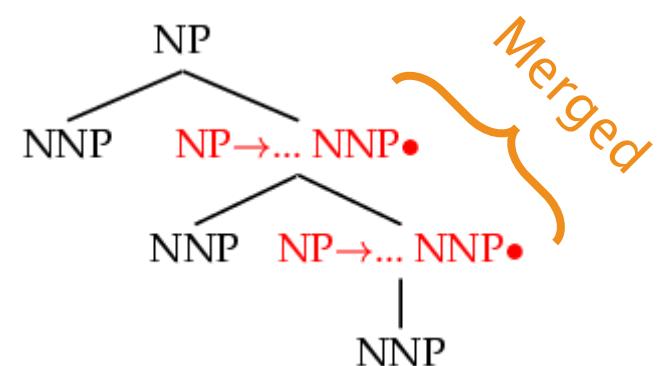
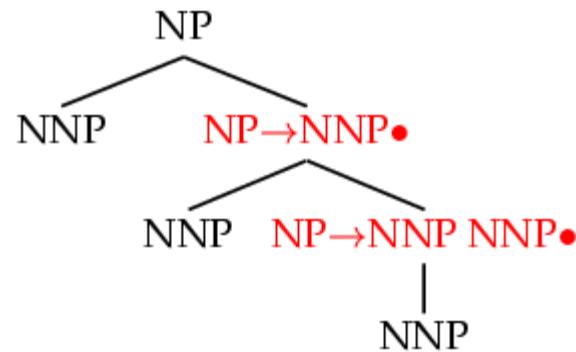
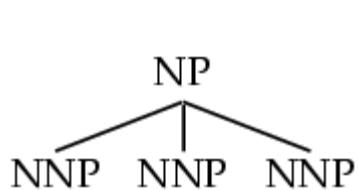


In the PTB, this construction is for possessives

# Advanced Unlexicalized Parsing

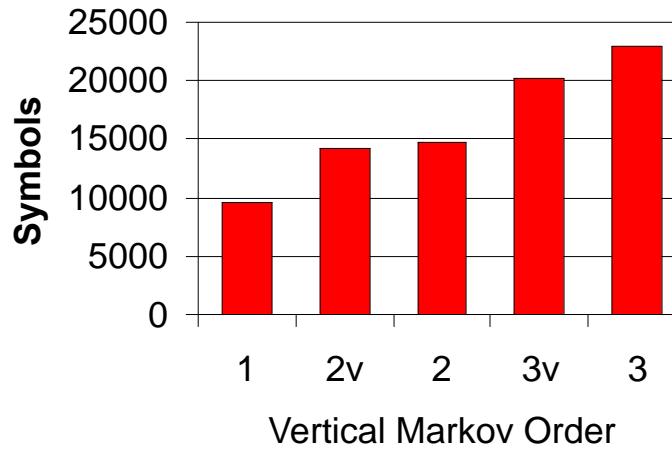
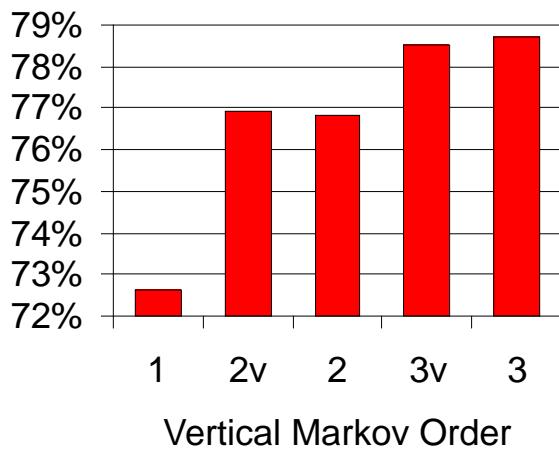
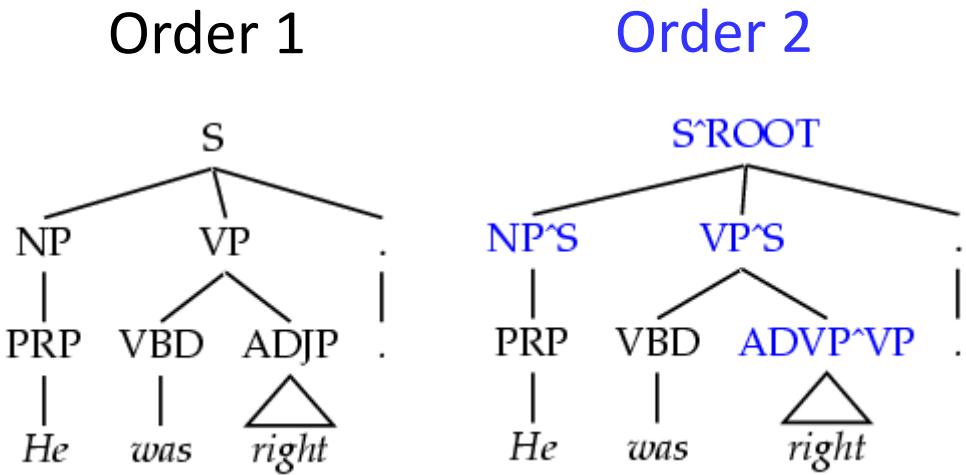
# Horizontal Markovization

- Horizontal Markovization: Merges States



# Vertical Markovization

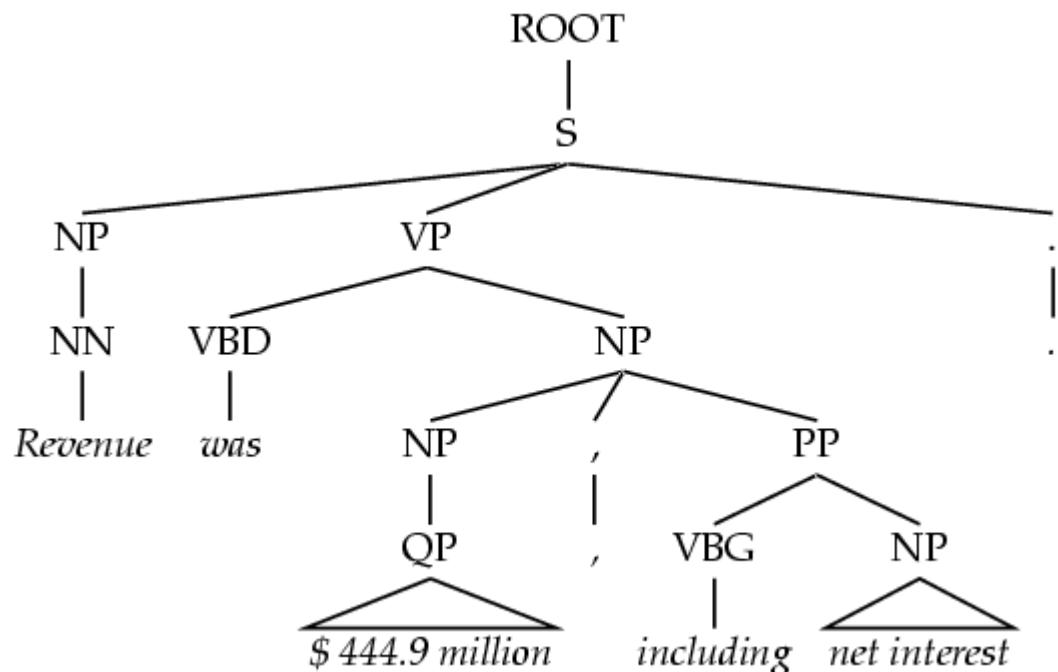
- Vertical Markov order:  
rewrites depend on past  
 $k$  ancestor nodes.  
(i.e., parent annotation)



| Model  | F1   | Size |
|--------|------|------|
| v=h=2v | 77.8 | 7.5K |

# Unary Splits

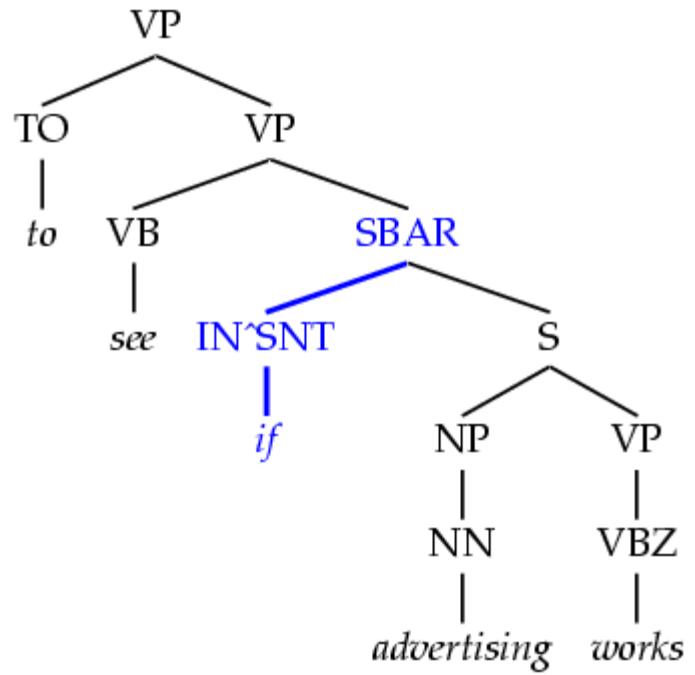
- Problem: unary rewrites are used to transmute categories so a high-probability rule can be used.
- Solution: Mark unary rewrite sites with -U



| Annotation | F1   | Size |
|------------|------|------|
| Base       | 77.8 | 7.5K |
| UNARY      | 78.3 | 8.0K |

# Tag Splits

- Problem: Treebank tags are too coarse.
- Example: SBAR sentential complementizers (*that*, *whether*, *if*), subordinating conjunctions (*while*, *after*), and true prepositions (*in*, *of*, *to*) are all tagged IN.
- Partial Solution:
  - Subdivide the IN tag.



| Annotation | F1   | Size |
|------------|------|------|
| Previous   | 78.3 | 8.0K |
| SPLIT-IN   | 80.3 | 8.1K |

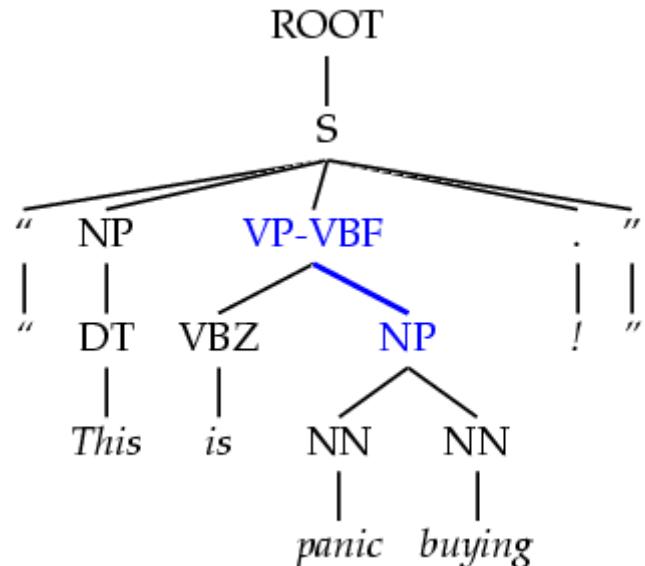
## Other Tag Splits

- UNARY-DT: mark demonstratives as DT<sup>U</sup> (“the X” vs. “those”)
- UNARY-RB: mark phrasal adverbs as RB<sup>U</sup> (“quickly” vs. “very”)
- TAG-PA: mark tags with non-canonical parents (“not” is an RB<sup>VP</sup>)
- SPLIT-AUX: mark auxiliary verbs with –AUX [cf. Charniak 97]
- SPLIT-CC: separate “but” and “&” from other conjunctions
- SPLIT-%: “%” gets its own tag.

| F1   | Size |
|------|------|
| 80.4 | 8.1K |
| 80.5 | 8.1K |
| 81.2 | 8.5K |
| 81.6 | 9.0K |
| 81.7 | 9.1K |
| 81.8 | 9.3K |

# Yield Splits

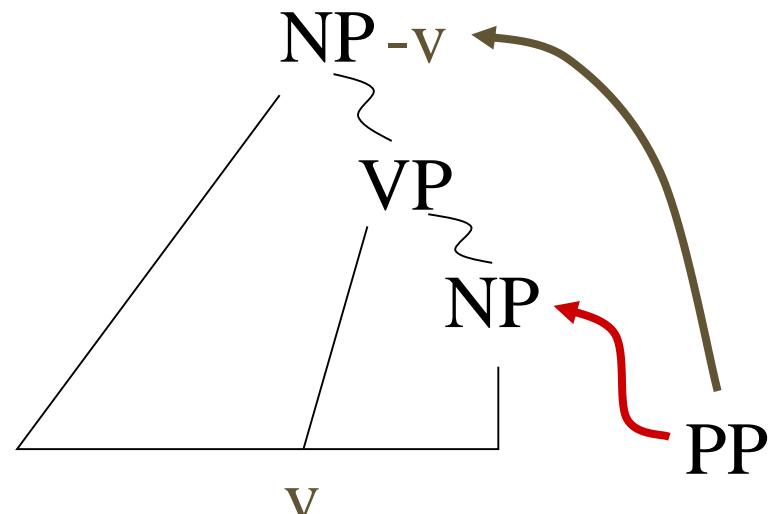
- Problem: sometimes the behavior of a category depends on something inside its future yield.
- Examples:
  - Possessive NPs
  - Finite vs. infinite VPs
  - Lexical heads!
- Solution: annotate future elements into nodes.



| Annotation | F1   | Size  |
|------------|------|-------|
| tag splits | 82.3 | 9.7K  |
| POSS-NP    | 83.1 | 9.8K  |
| SPLIT-VP   | 85.7 | 10.5K |

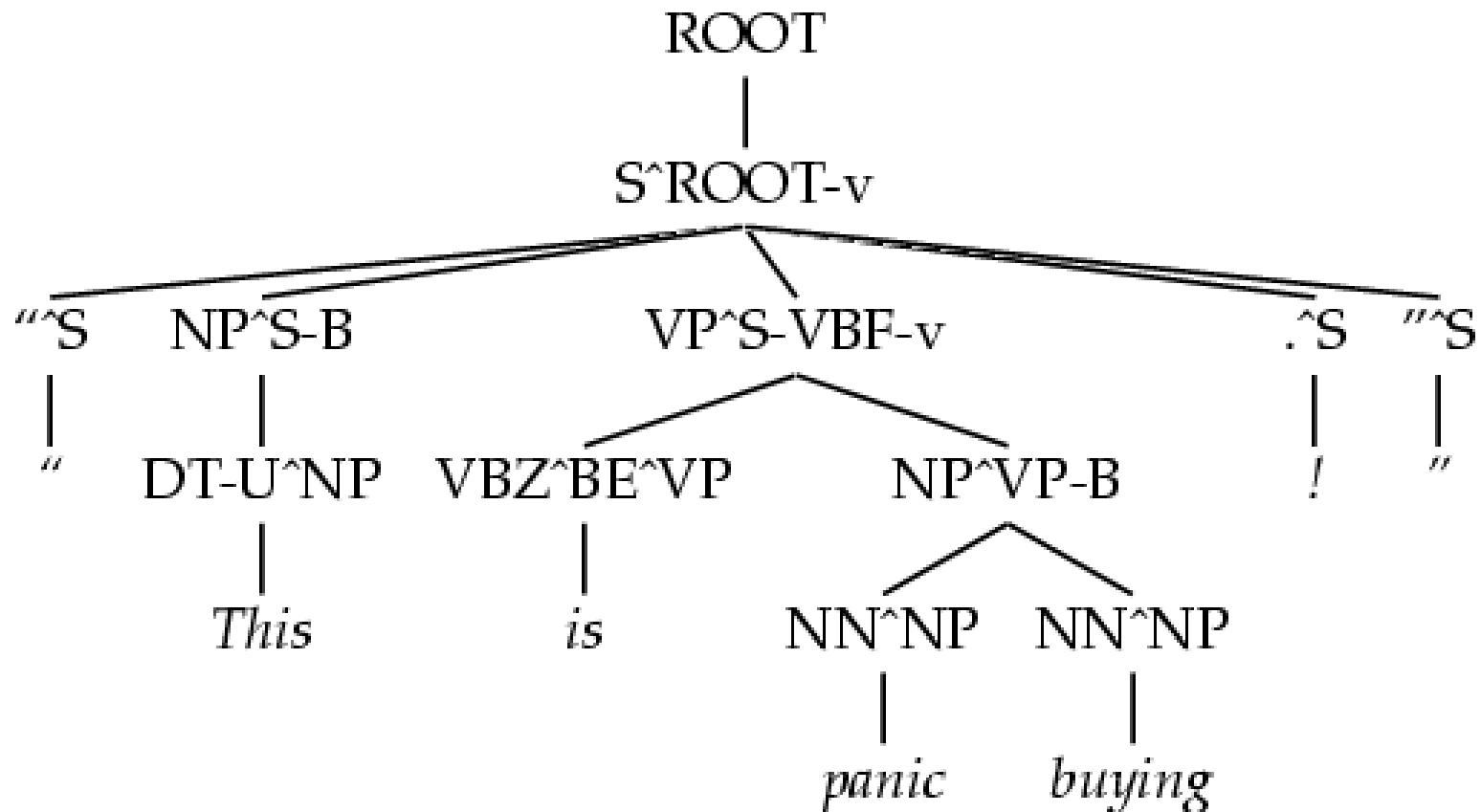
# Distance / Recursion Splits

- Problem: vanilla PCFGs cannot distinguish attachment heights.
- Solution: mark a property of higher or lower sites:
  - Contains a verb.
  - Is (non)-recursive.
    - Base NPs [cf. Collins 99]
    - Right-recursive NPs



| Annotation   | F1   | Size  |
|--------------|------|-------|
| Previous     | 85.7 | 10.5K |
| BASE-NP      | 86.0 | 11.7K |
| DOMINATES-V  | 86.9 | 14.1K |
| RIGHT-REC-NP | 87.0 | 15.2K |

# A Fully Annotated Tree



# Final Test Set Results

| Parser             | LP   | LR   | F1          |
|--------------------|------|------|-------------|
| Magerman 95        | 84.9 | 84.6 | <b>84.7</b> |
| Collins 96         | 86.3 | 85.8 | <b>86.0</b> |
| Klein & Manning 03 | 86.9 | 85.7 | <b>86.3</b> |
| Charniak 97        | 87.4 | 87.5 | <b>87.4</b> |
| Collins 99         | 88.7 | 88.6 | <b>88.6</b> |

- Beats “first generation” lexicalized parsers

# Lexicalised PCFGs

# Heads in Comtext Free Rules

Add annotations specifying the “**head**” of each rule:

|    |               |    |    |
|----|---------------|----|----|
| S  | $\Rightarrow$ | NP | VP |
| VP | $\Rightarrow$ | Vi |    |
| VP | $\Rightarrow$ | Vt | NP |
| VP | $\Rightarrow$ | VP | PP |
| NP | $\Rightarrow$ | DT | NN |
| NP | $\Rightarrow$ | NP | PP |
| PP | $\Rightarrow$ | IN | NP |

|    |               |           |
|----|---------------|-----------|
| Vi | $\Rightarrow$ | sleeps    |
| Vt | $\Rightarrow$ | saw       |
| NN | $\Rightarrow$ | man       |
| NN | $\Rightarrow$ | woman     |
| NN | $\Rightarrow$ | telescope |
| DT | $\Rightarrow$ | the       |
| IN | $\Rightarrow$ | with      |
| IN | $\Rightarrow$ | in        |

# Heads

- ▶ Each context-free rule has one “special” child that is the head of the rule. e.g.,

$$\begin{array}{lll} S & \Rightarrow & NP \quad VP \\ & & \quad \quad \quad VP \text{ is the head} \\ VP & \Rightarrow & Vt \quad NP \\ & & \quad \quad \quad (Vt \text{ is the head}) \\ NP & \Rightarrow & DT \quad NN \quad NN \\ & & \quad \quad \quad (NN \text{ is the head}) \end{array}$$

- ▶ A core idea in syntax  
(e.g., see X-bar Theory, Head-Driven Phrase Structure Grammar)
- ▶ Some intuitions:
  - ▶ The central sub-constituent of each rule.
  - ▶ The semantic predicate in each rule.

# Rules to Recover Heads: An Example for NPs

**If** the rule contains NN, NNS, or NNP:

Choose the rightmost NN, NNS, or NNP

**Else If** the rule contains an NP: Choose the leftmost NP

**Else If** the rule contains a JJ: Choose the rightmost JJ

**Else If** the rule contains a CD: Choose the rightmost CD

**Else** Choose the rightmost child

e.g.,

|    |               |    |     |     |
|----|---------------|----|-----|-----|
| NP | $\Rightarrow$ | DT | NNP | NN  |
| NP | $\Rightarrow$ | DT | NN  | NNP |
| NP | $\Rightarrow$ | NP | PP  |     |
| NP | $\Rightarrow$ | DT | JJ  |     |
| NP | $\Rightarrow$ | DT |     |     |

## Rules to Recover Heads: An Example for VPs

**If** the rule contains  $V_i$  or  $V_t$ : Choose the leftmost  $V_i$  or  $V_t$

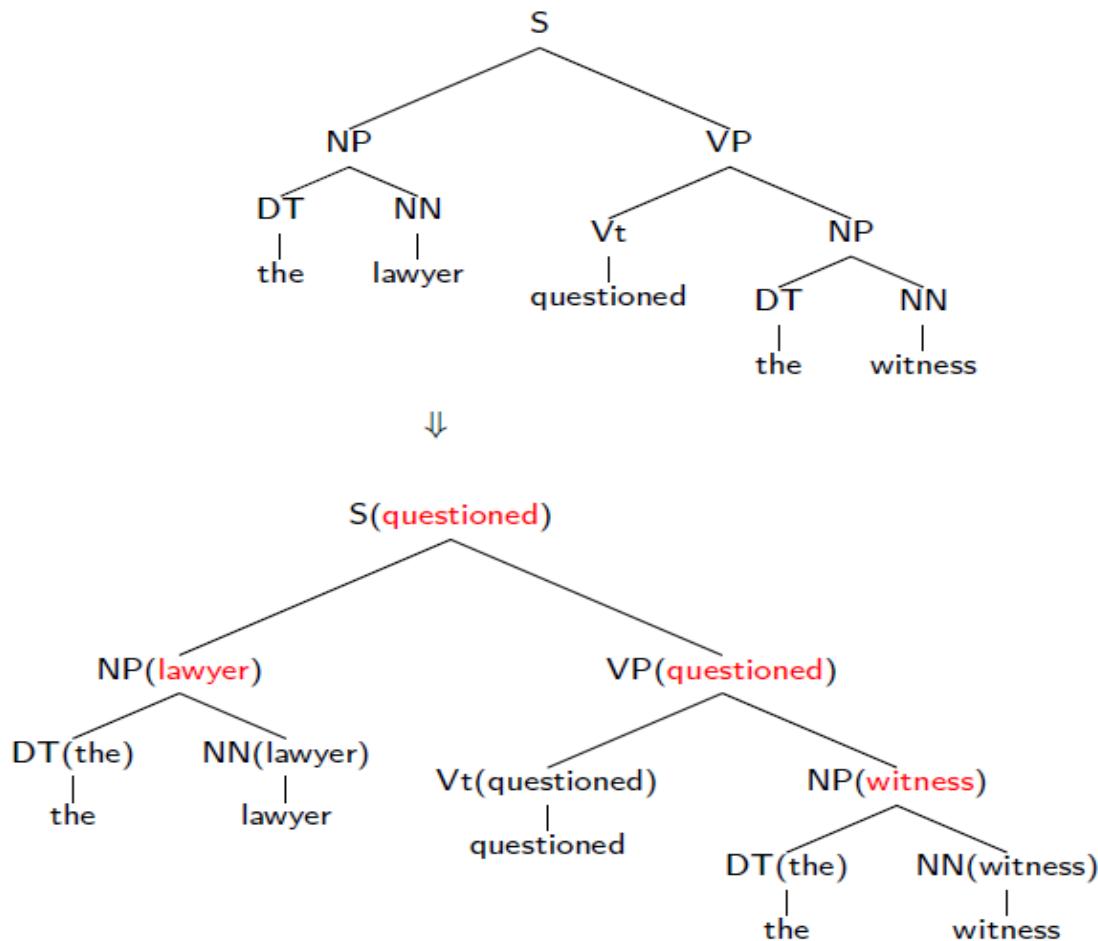
**Else If** the rule contains an VP: Choose the leftmost VP

**Else** Choose the leftmost child

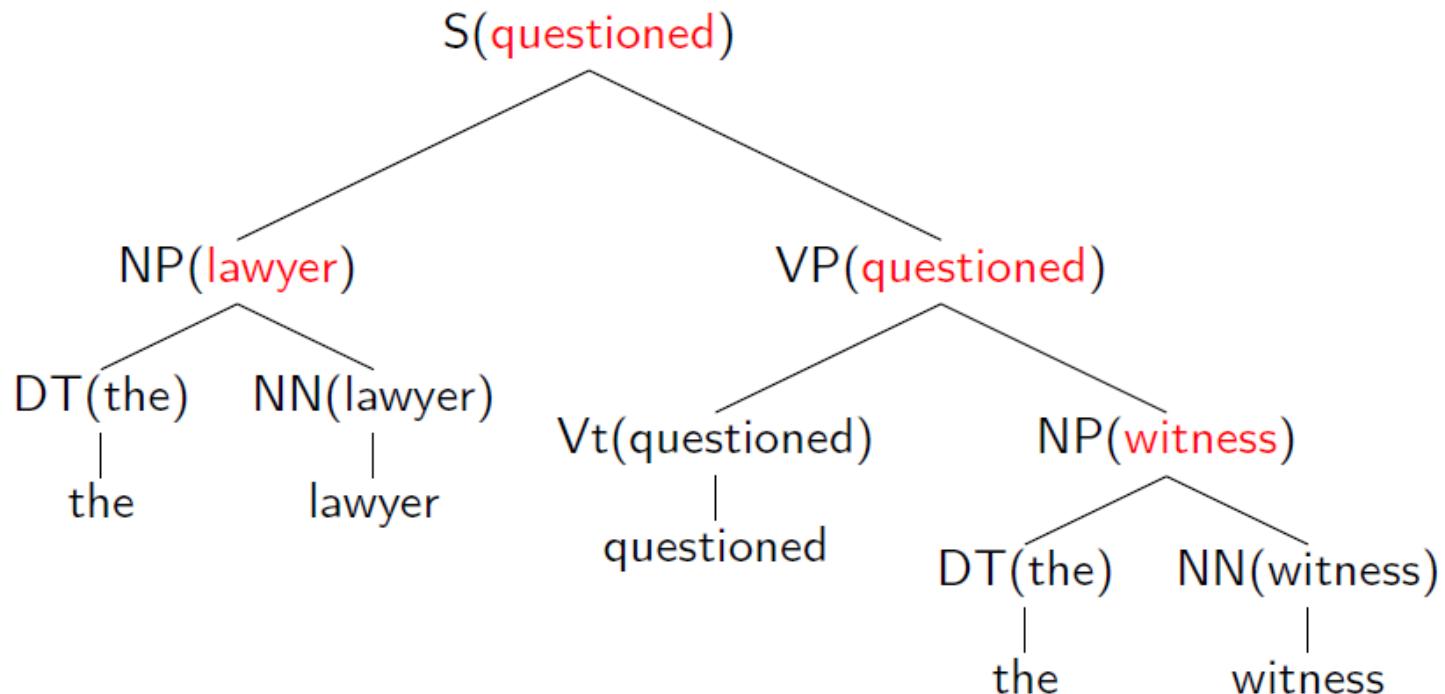
e.g.,

$$\begin{array}{lll} \text{VP} & \Rightarrow & \text{Vt} \quad \text{NP} \\ \text{VP} & \Rightarrow & \text{VP} \quad \text{PP} \end{array}$$

# Adding Headwords to Trees



# Adding Headwords to Trees



- ▶ A constituent receives its **headword** from its **head child**.

$S \Rightarrow NP \quad VP$   
 $VP \Rightarrow Vt \quad NP$   
 $NP \Rightarrow DT \quad NN$

( $S$  receives headword from  $VP$ )  
( $VP$  receives headword from  $Vt$ )  
( $NP$  receives headword from  $NN$ )

# Lexicalized CFGs in Chomsky Normal Form

- ▶  $N$  is a set of non-terminal symbols
- ▶  $\Sigma$  is a set of terminal symbols
- ▶  $R$  is a set of rules which take one of three forms:
  - ▶  $X(h) \rightarrow_1 Y_1(h) Y_2(w)$  for  $X \in N$ , and  $Y_1, Y_2 \in N$ , and  $h, w \in \Sigma$
  - ▶  $X(h) \rightarrow_2 Y_1(w) Y_2(h)$  for  $X \in N$ , and  $Y_1, Y_2 \in N$ , and  $h, w \in \Sigma$
  - ▶  $X(h) \rightarrow h$  for  $X \in N$ , and  $h \in \Sigma$
- ▶  $S \in N$  is a distinguished start symbol

## Example

|         |                 |         |         |
|---------|-----------------|---------|---------|
| S(saw)  | $\rightarrow_2$ | NP(man) | VP(saw) |
| VP(saw) | $\rightarrow_1$ | Vt(saw) | NP(dog) |
| NP(man) | $\rightarrow_2$ | DT(the) | NN(man) |
| NP(dog) | $\rightarrow_2$ | DT(the) | NN(dog) |
| Vt(saw) | $\rightarrow$   | saw     |         |
| DT(the) | $\rightarrow$   | the     |         |
| NN(man) | $\rightarrow$   | man     |         |
| NN(dog) | $\rightarrow$   | dog     |         |

# Lexicalized CKY

$(VP \rightarrow VBD \dots NP) [saw]$

$(VP \rightarrow \quad VBD[saw] \quad NP[her])$

```
bestScore(X, i, j, h)
```

```
if (j = i)
```

```
    return score(X, s[i])
```

```
else
```

```
    return
```

```
        max max score(X[h] -> Y[h] Z[w]) *
```

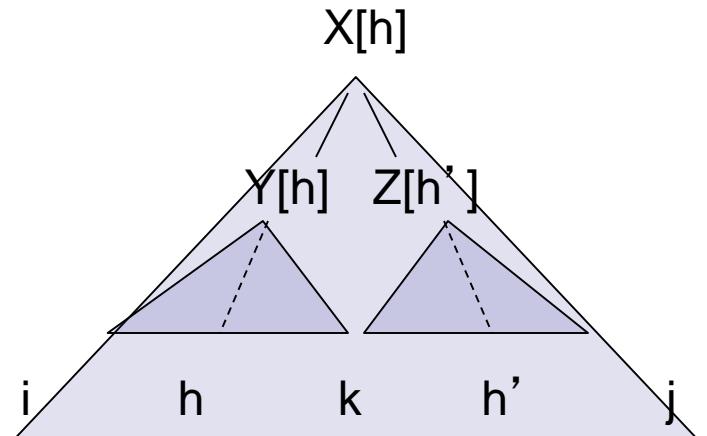
```
        x->yz bestScore(Y, i, k, h) *
```

```
                bestScore(Z, k, j, w)
```

```
        max max score(X[h] -> Y[w] Z[h]) *
```

```
        x->yz bestScore(Y, i, k, w) *
```

```
                bestScore(Z, k, j, h)
```

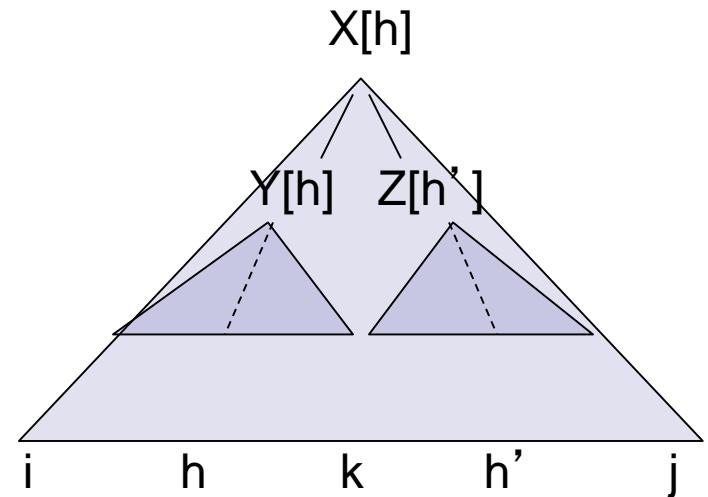


# Parsing with Lexicalized CFGs

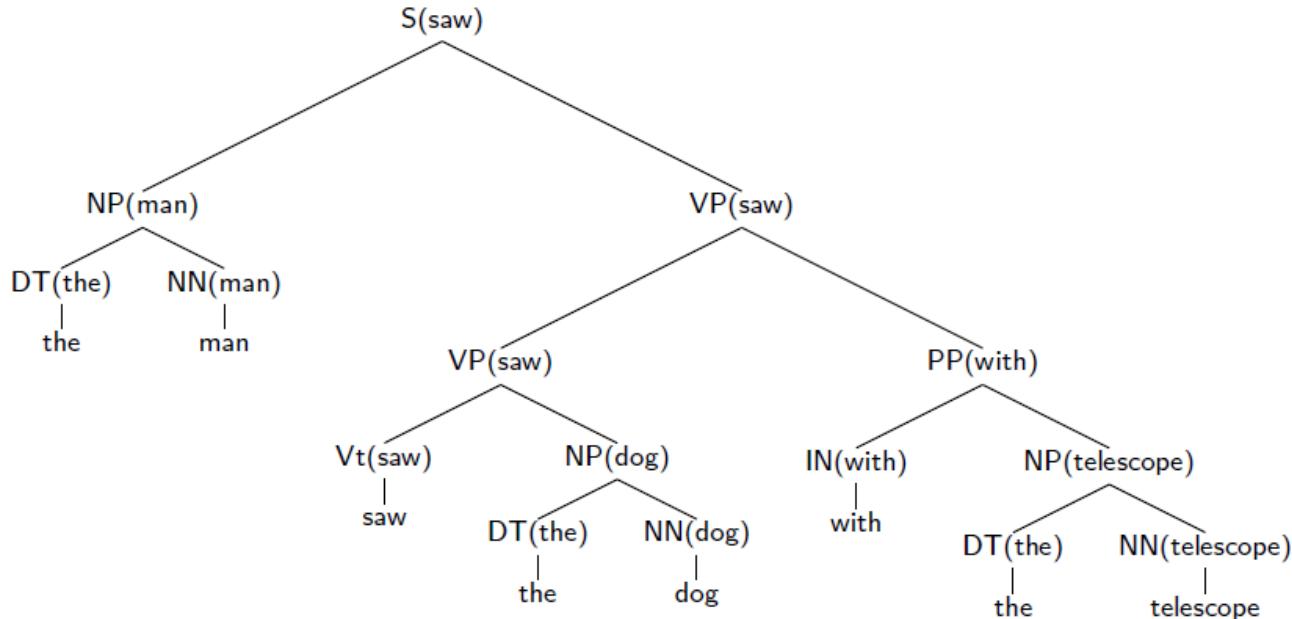
- ▶ The new form of grammar looks just like a Chomsky normal form CFG, but with potentially  $O(|\Sigma|^2 \times |N|^3)$  possible rules.
- ▶ Naively, parsing an  $n$  word sentence using the dynamic programming algorithm will take  $O(n^3|\Sigma|^2|N|^3)$  time. **But  $|\Sigma|$  can be huge!!**
- ▶ Crucial observation: at most  $O(n^2 \times |N|^3)$  rules can be applicable to a given sentence  $w_1, w_2, \dots, w_n$  of length  $n$ . This is because any rules which contain a lexical item that is not one of  $w_1 \dots w_n$ , can be safely discarded.
- ▶ The result: we can parse in  $O(n^5|N|^3)$  time.

# Pruning with Beams

- The Collins parser prunes with per-cell beams [Collins 99]
  - Essentially, run the  $O(n^5)$  CKY
  - Remember only a few hypotheses for each span  $\langle i, j \rangle$ .
  - If we keep  $K$  hypotheses at each span, then we do at most  $O(nK^2)$  work per span (why?)
  - Keeps things more or less cubic
- Also: certain spans are forbidden entirely on the basis of punctuation (crucial for speed)



# Parameter Estimation



$$\begin{aligned} p(t) &= q(S(\text{saw}) \rightarrow_2 \text{NP}(\text{man}) \text{ VP}(\text{saw})) \\ &\quad \times q(\text{NP}(\text{man}) \rightarrow_2 \text{DT}(\text{the}) \text{ NN}(\text{man})) \\ &\quad \times q(\text{VP}(\text{saw}) \rightarrow_1 \text{VP}(\text{saw}) \text{ PP}(\text{with})) \\ &\quad \times q(\text{VP}(\text{saw}) \rightarrow_1 \text{Vt}(\text{saw}) \text{ NP}(\text{dog})) \\ &\quad \times q(\text{PP}(\text{with}) \rightarrow_1 \text{IN}(\text{with}) \text{ NP}(\text{telescope})) \\ &\quad \times \dots \end{aligned}$$

## A Model from Charniak (1997)

- ▶ An example parameter in a Lexicalized PCFG:

$$q(S(\text{saw}) \rightarrow_2 NP(\text{man}) VP(\text{saw}))$$

- ▶ First step: decompose this parameter into a product of two parameters

$$\begin{aligned} & q(S(\text{saw}) \rightarrow_2 NP(\text{man}) VP(\text{saw})) \\ = & q(S \rightarrow_2 NP VP | S, \text{saw}) \times q(\text{man} | S \rightarrow_2 NP VP, \text{saw}) \end{aligned}$$

# A Model from Charniak (1997)

$$\begin{aligned} & q(S(\text{saw}) \rightarrow_2 \text{NP}(\text{man}) \text{ VP}(\text{saw})) \\ = & q(S \rightarrow_2 \text{NP VP}|S, \text{saw}) \times q(\text{man}|S \rightarrow_2 \text{NP VP}, \text{saw}) \end{aligned}$$

- ▶ Second step: use smoothed estimation for the two parameter estimates

$$\begin{aligned} & q(S \rightarrow_2 \text{NP VP}|S, \text{saw}) \\ = & \lambda_1 \times q_{ML}(S \rightarrow_2 \text{NP VP}|S, \text{saw}) + \lambda_2 \times q_{ML}(S \rightarrow_2 \text{NP VP}|S) \end{aligned}$$

$$\begin{aligned} & q(\text{man}|S \rightarrow_2 \text{NP VP}, \text{saw}) \\ = & \lambda_3 \times q_{ML}(\text{man}|S \rightarrow_2 \text{NP VP}, \text{saw}) + \lambda_4 \times q_{ML}(\text{man}|S \rightarrow_2 \text{NP VP}) \\ & + \lambda_5 \times q_{ML}(\text{man}|NP) \end{aligned}$$

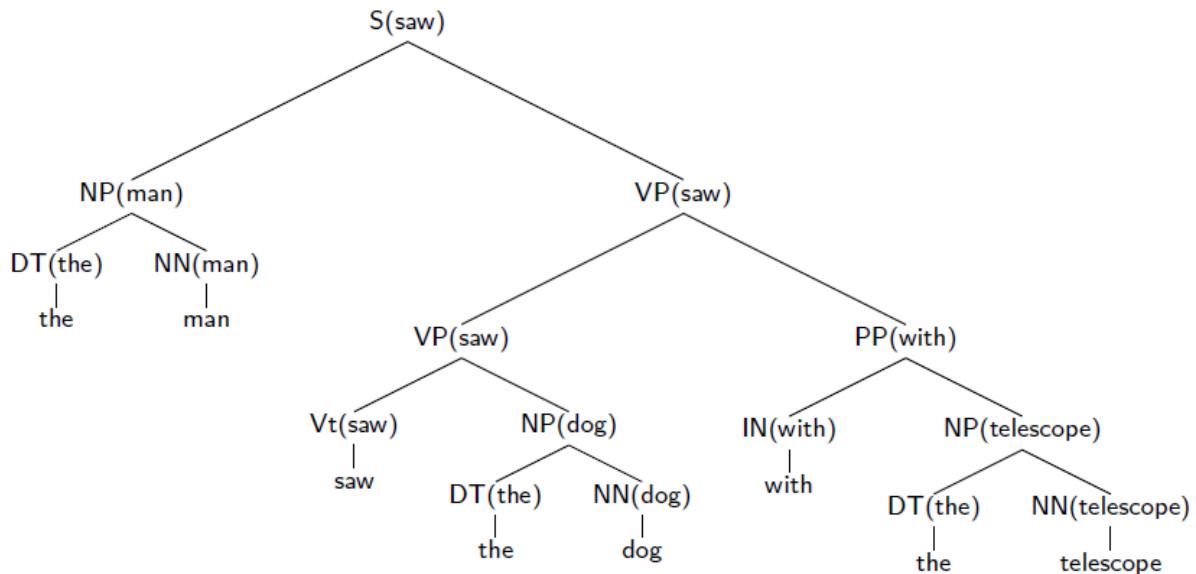
## Other Details

- ▶ Need to deal with rules with more than two children, e.g.,  
 $\text{VP}(\text{told}) \rightarrow \text{V}(\text{told}) \text{ NP}(\text{him}) \text{ PP}(\text{on}) \text{ SBAR}(\text{that})$
- ▶ Need to incorporate parts of speech (useful in smoothing)  
 $\text{VP-V}(\text{told}) \rightarrow \text{V}(\text{told}) \text{ NP-PRP}(\text{him}) \text{ PP-IN}(\text{on}) \text{ SBAR-COMP}(\text{that})$
- ▶ Need to encode preferences for close attachment  
John was believed to have been shot by Bill

# Final Test Set Results

| Parser             | LP   | LR   | F1          |
|--------------------|------|------|-------------|
| Magerman 95        | 84.9 | 84.6 | <b>84.7</b> |
| Collins 96         | 86.3 | 85.8 | <b>86.0</b> |
| Klein & Manning 03 | 86.9 | 85.7 | <b>86.3</b> |
| Charniak 97        | 87.4 | 87.5 | <b>87.4</b> |
| Collins 99         | 88.7 | 88.6 | <b>88.6</b> |

# Analysis/Evaluation (Method 2)



|                            |                          |                           |
|----------------------------|--------------------------|---------------------------|
| { ROOT <sub>0</sub> ,      | saw <sub>3</sub> ,       | ROOT }                    |
| { saw <sub>3</sub> ,       | man <sub>2</sub> ,       | S → <sub>2</sub> NP VP }  |
| { man <sub>2</sub> ,       | the <sub>1</sub> ,       | NP → <sub>2</sub> DT NN } |
| { saw <sub>3</sub> ,       | with <sub>6</sub> ,      | VP → <sub>1</sub> VP PP } |
| { saw <sub>3</sub> ,       | dog <sub>5</sub> ,       | VP → <sub>1</sub> Vt NP } |
| { dog <sub>5</sub> ,       | the <sub>4</sub> ,       | NP → <sub>2</sub> DT NN } |
| { with <sub>6</sub> ,      | telescope <sub>8</sub> , | PP → <sub>1</sub> IN NP } |
| { telescope <sub>8</sub> , | the <sub>7</sub> ,       | NP → <sub>2</sub> DT NN } |

# Dependency Accuracies

- ▶ All parses for a sentence with  $n$  words have  $n$  dependencies  
*Report a single figure, dependency accuracy*
- ▶ Results from Collins, 2003: 88.3% dependency accuracy
- ▶ Can calculate precision/recall on particular dependency **types**  
e.g., look at all subject/verb dependencies  $\Rightarrow$   
all dependencies with label  $S \rightarrow_2 NP VP$

Recall =

$$\frac{\text{number of subject/verb dependencies correct}}{\text{number of subject/verb dependencies in gold standard}}$$

Precision =

$$\frac{\text{number of subject/verb dependencies correct}}{\text{number of subject/verb dependencies in parser's output}}$$

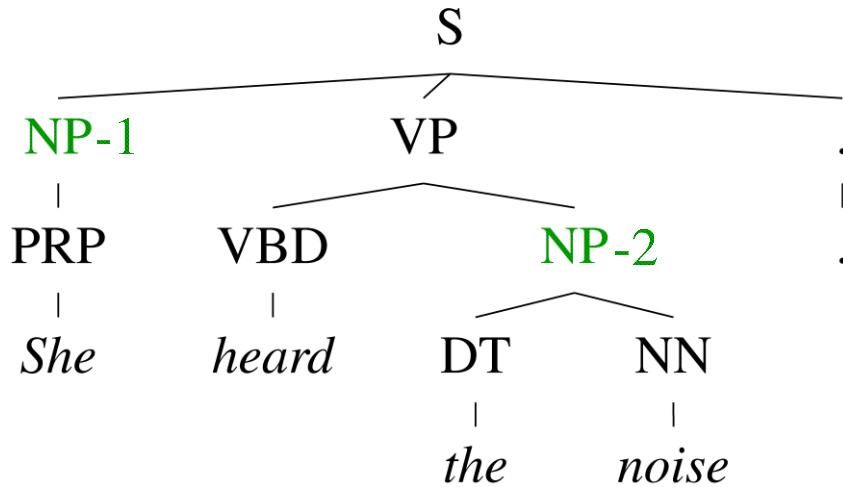
# Strengths and Weaknesses of Modern Parsers

(Numbers taken from Collins (2003))

- ▶ Subject-verb pairs: over 95% recall and precision
- ▶ Object-verb pairs: over 92% recall and precision
- ▶ Other arguments to verbs:  $\approx$  93% recall and precision
- ▶ Non-recursive NP boundaries:  $\approx$  93% recall and precision
- ▶ PP attachments:  $\approx$  82% recall and precision
- ▶ Coordination ambiguities:  $\approx$  61% recall and precision

# Modern Parsers

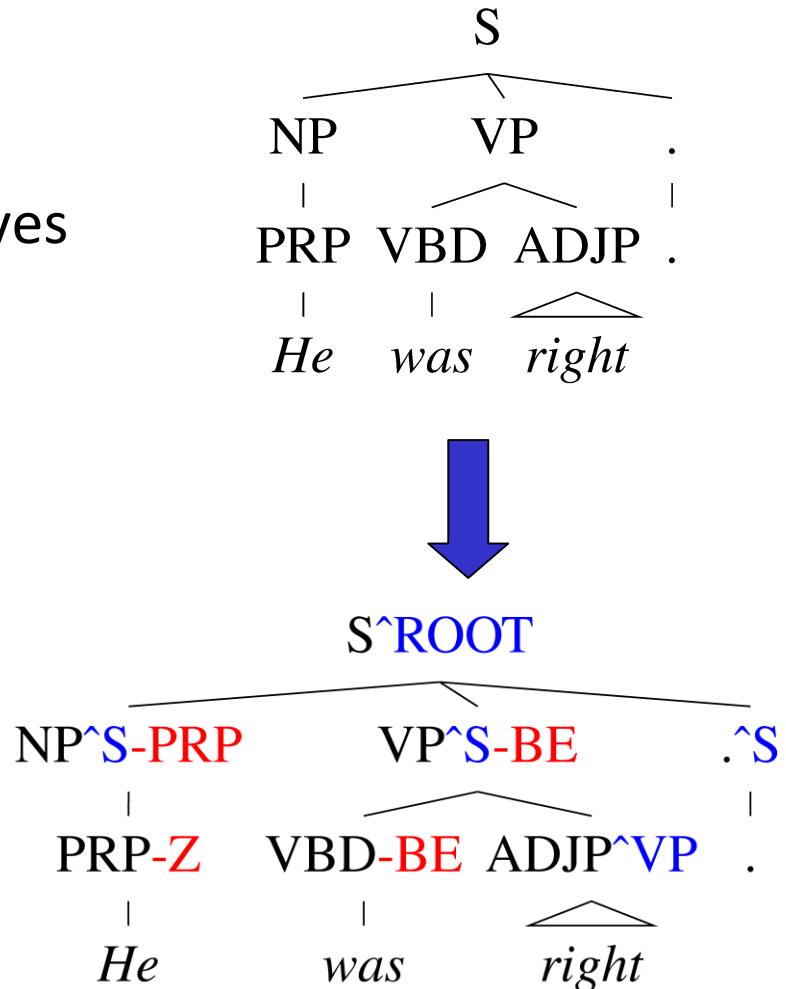
# The Game of Designing a Grammar



- Annotation refines base treebank symbols to improve statistical fit of the grammar
- Parent annotation [Johnson '98]
- Head lexicalization [Collins '99, Charniak '00]
- Automatic clustering?

# Manual Splits

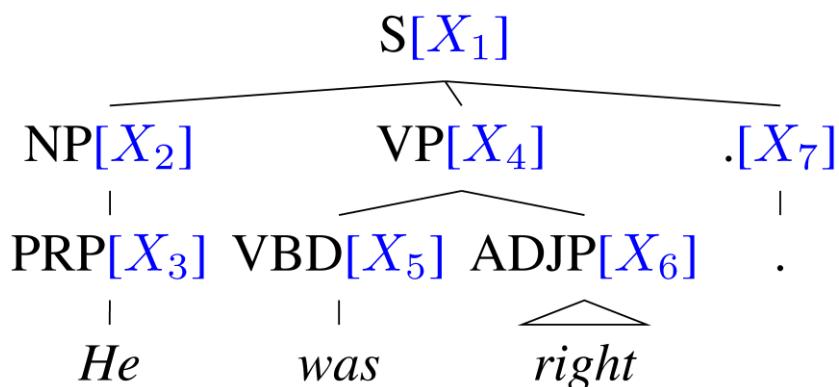
- Manually split categories
  - NP: subject vs object
  - DT: determiners vs demonstratives
  - IN: sentential vs prepositional
- Advantages:
  - Fairly compact grammar
  - Linguistic motivations
- Disadvantages:
  - Performance leveled out
  - Manually annotated



# Learning Latent Annotations

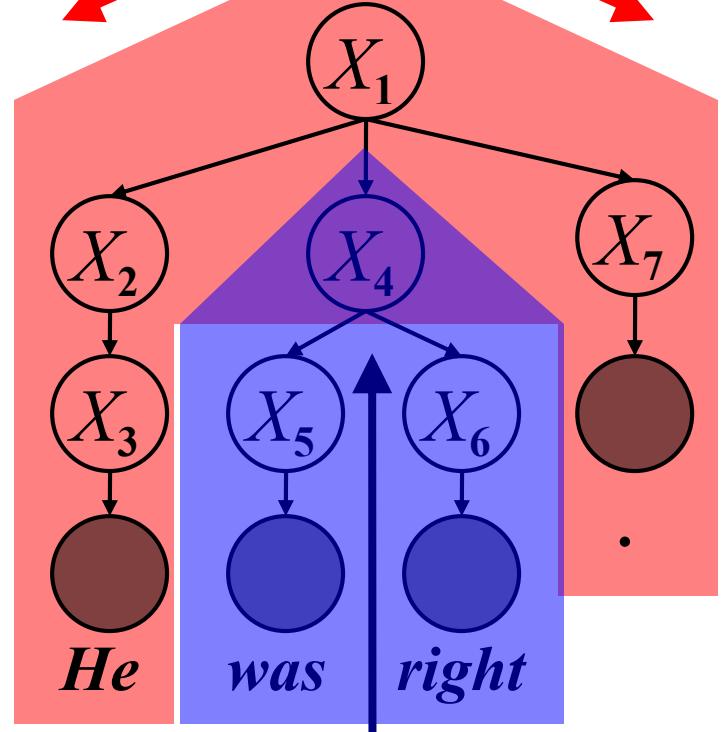
Latent Annotations:

- Brackets are known
- Base categories are known
- Hidden variables for subcategories



Can learn with EM: like Forward-Backward for HMMs.

Forward/Outside



Backward/Inside

# Automatic Annotation Induction

- Advantages:

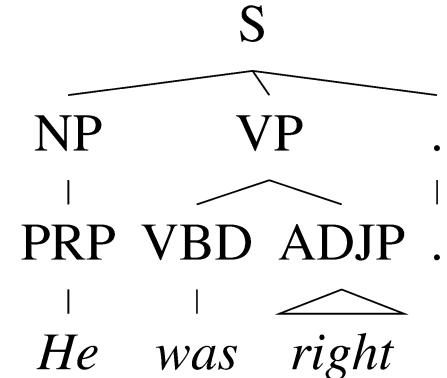
- Automatically learned:

- Label *all* nodes with latent variables.

- Same number  $k$  of subcategories for all categories.

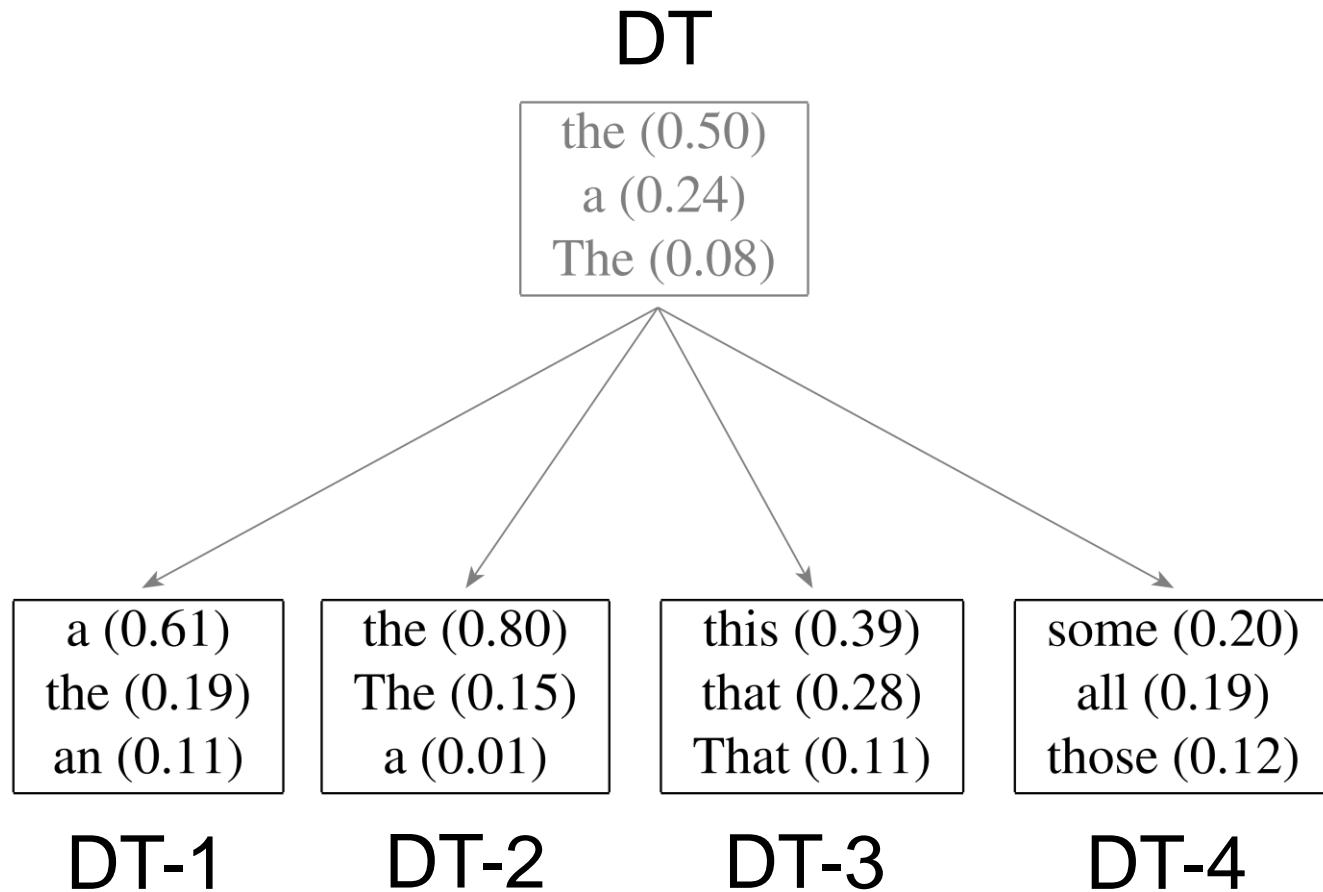
- Disadvantages:

- Grammar gets too large
  - Most categories are oversplit while others are undersplit.



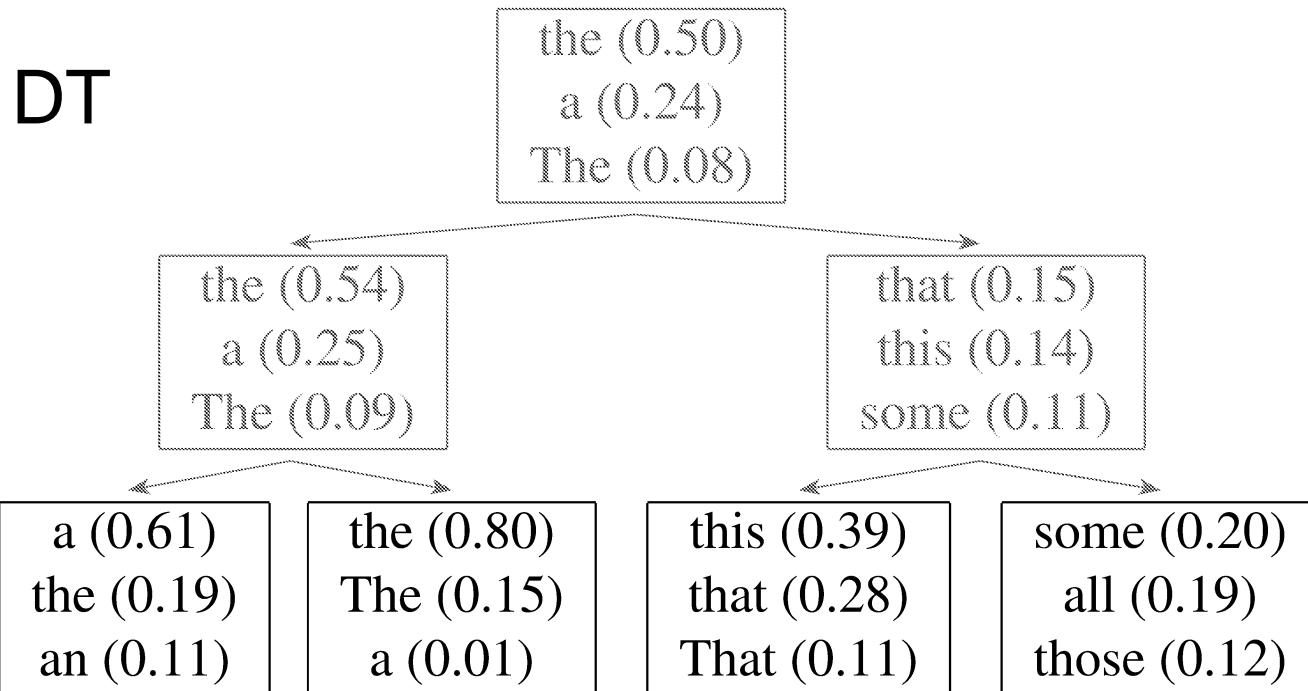
| Model                | F1   |
|----------------------|------|
| Klein & Manning '03  | 86.3 |
| Matsuzaki et al. '05 | 86.7 |

# Refinement of the DT tag



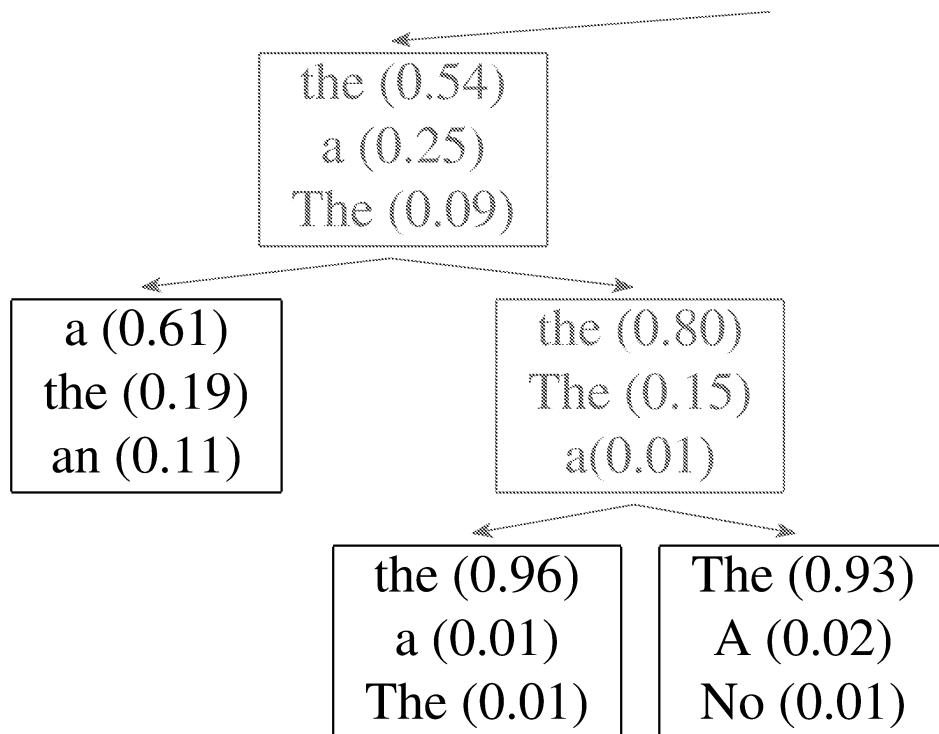
# Hierarchical refinement

- Repeatedly learn more fine-grained subcategories
- start with two (per non-terminal), then keep splitting
- initialize each EM run with the output of the last



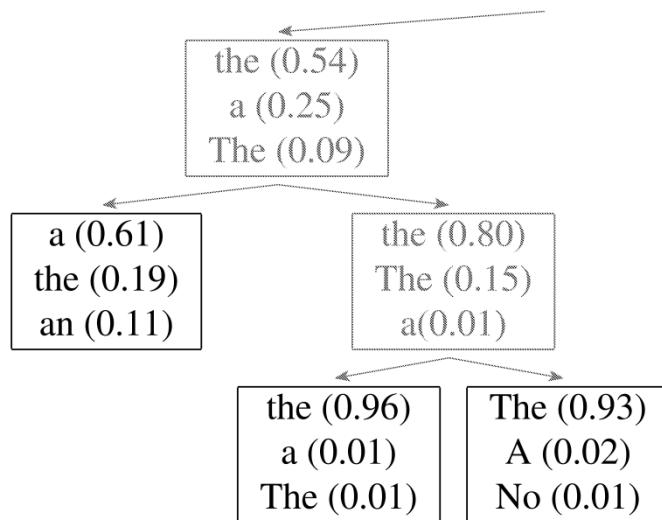
# Adaptive Splitting

- Want to split complex categories more
- Idea: split everything, roll back splits which were least useful

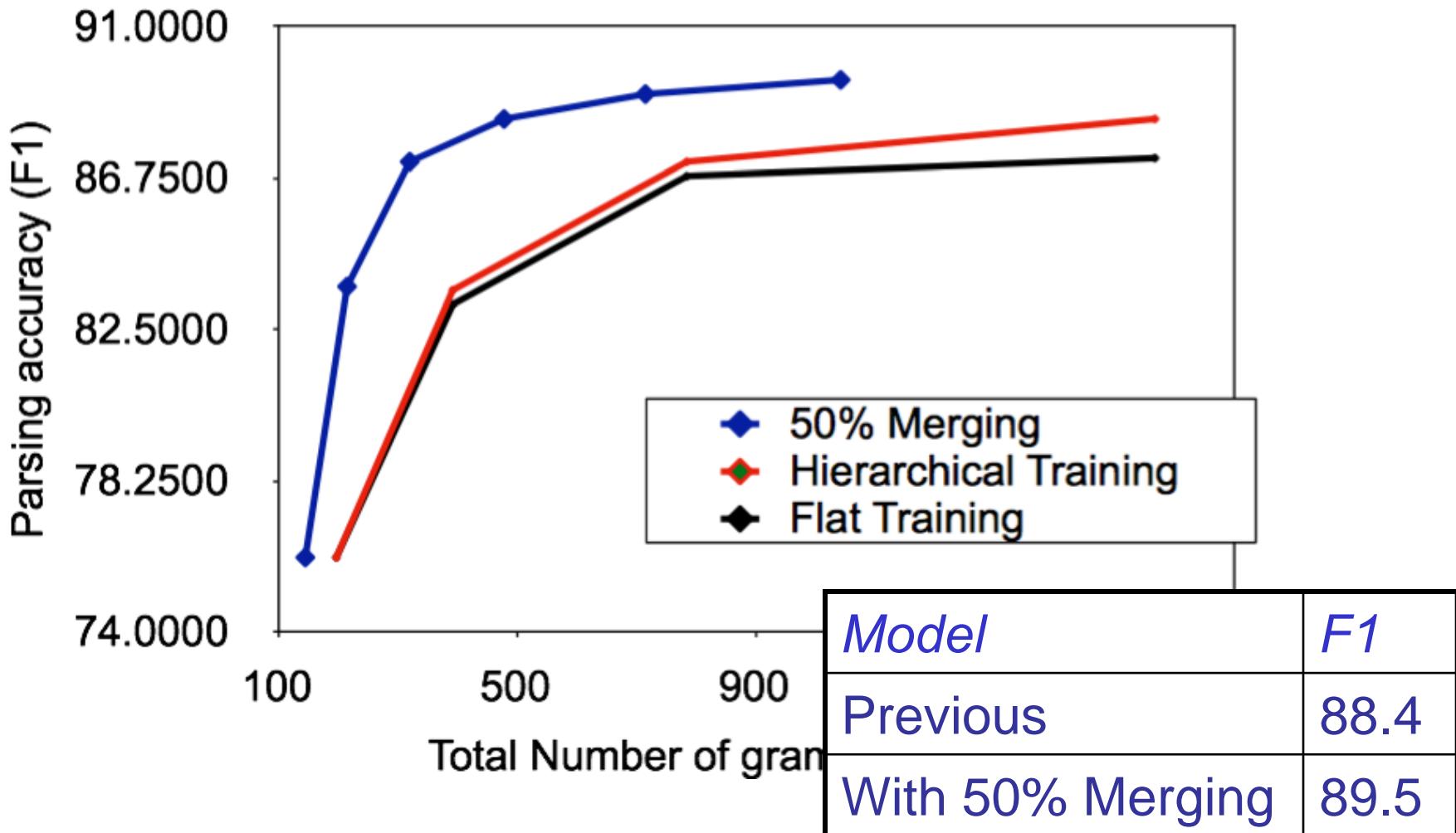


# Adaptive Splitting

- Evaluate loss in likelihood from removing each split =  
$$\frac{\text{Data likelihood with split reversed}}{\text{Data likelihood with split}}$$
- No loss in accuracy when 50% of the splits are reversed.

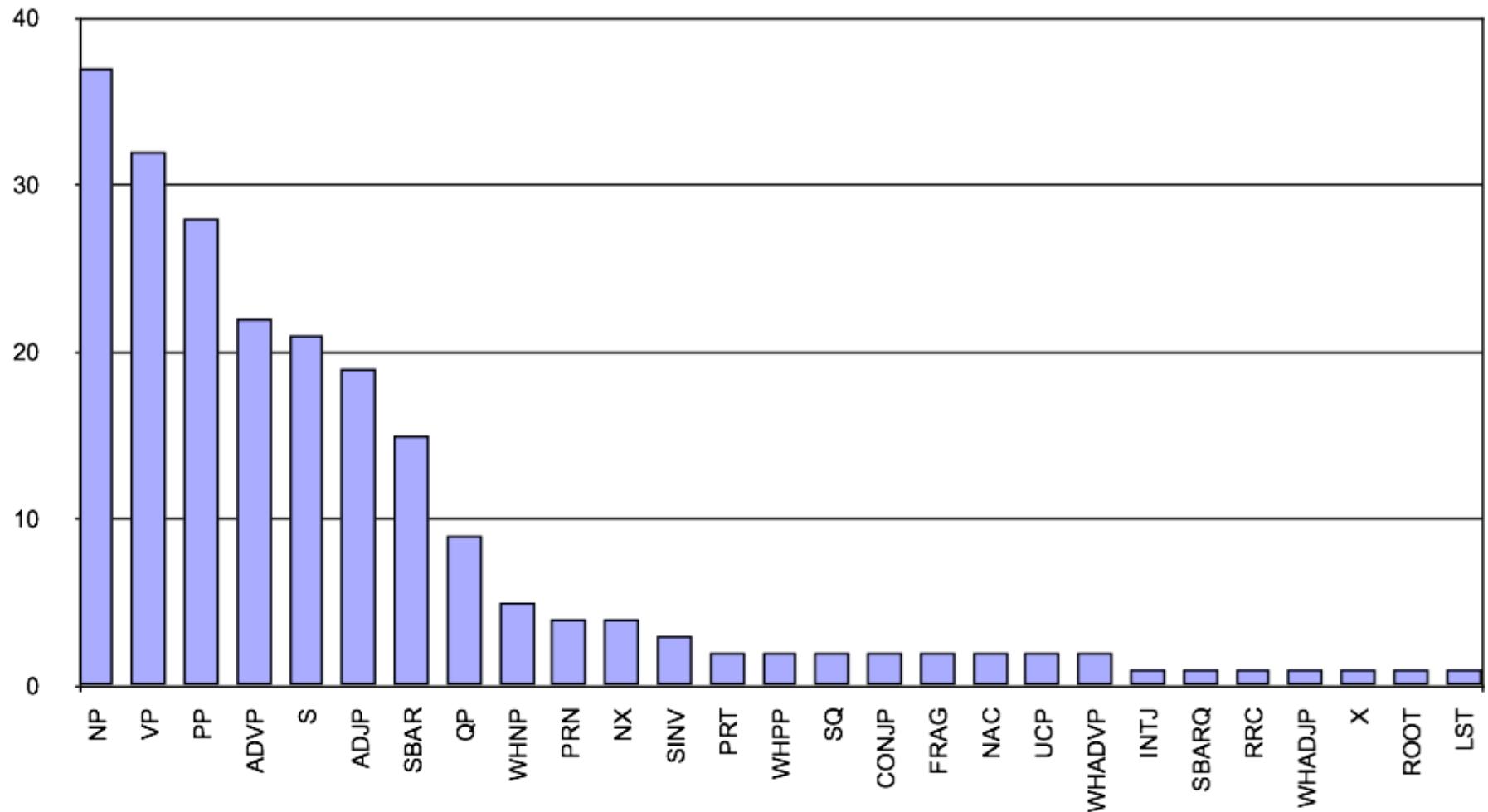


# Adaptive Splitting Results



# Number of Phrasal Subcategories

---



# Final Results

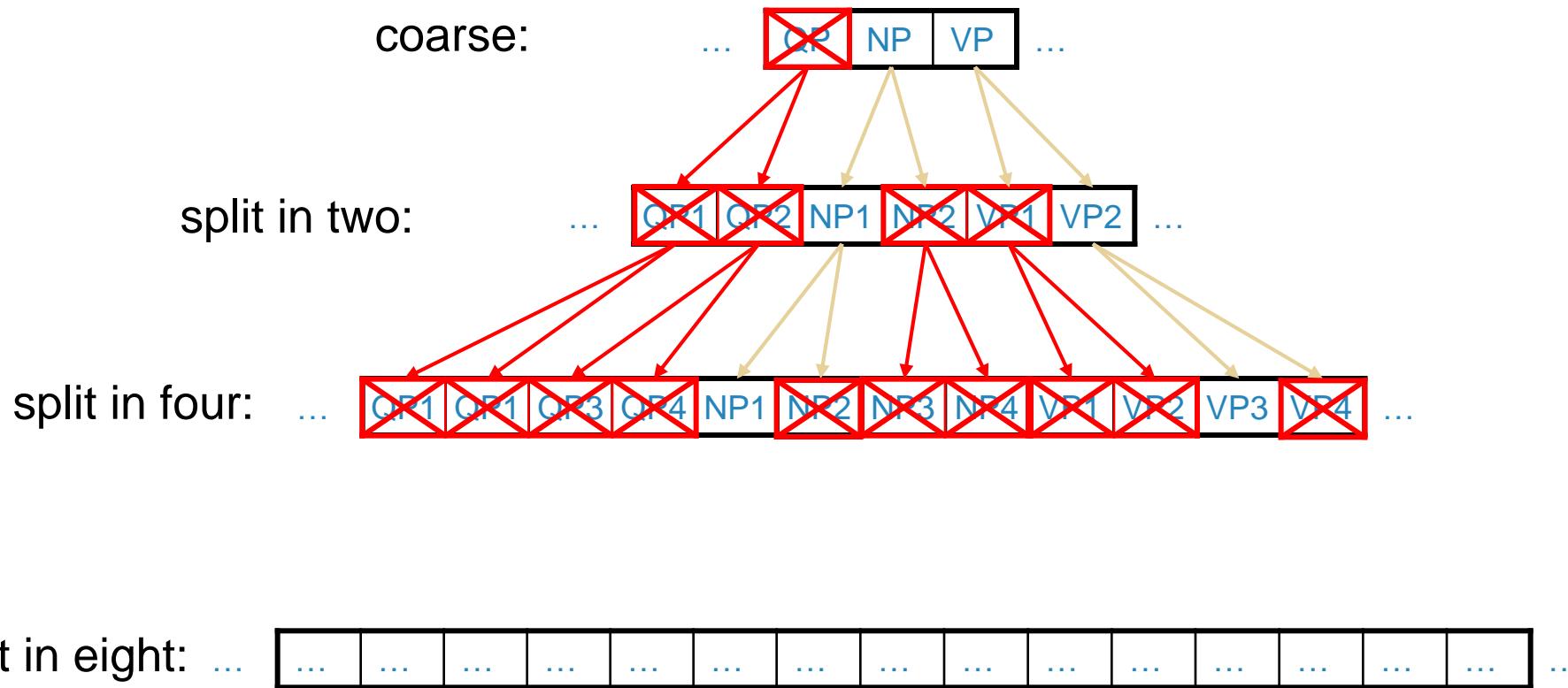
---

| <i>Parser</i>            | <i>F1</i><br>$\leq 40$ words | <i>F1</i><br>all words |
|--------------------------|------------------------------|------------------------|
| Klein & Manning '03      | 86.3                         | 85.7                   |
| Matsuzaki et al. '05     | 86.7                         | 86.1                   |
| Collins '99              | 88.6                         | 88.2                   |
| Charniak & Johnson '05   | 90.1                         | 89.6                   |
| <b>Petrov et. al. 06</b> | <b>90.2</b>                  | <b>89.7</b>            |

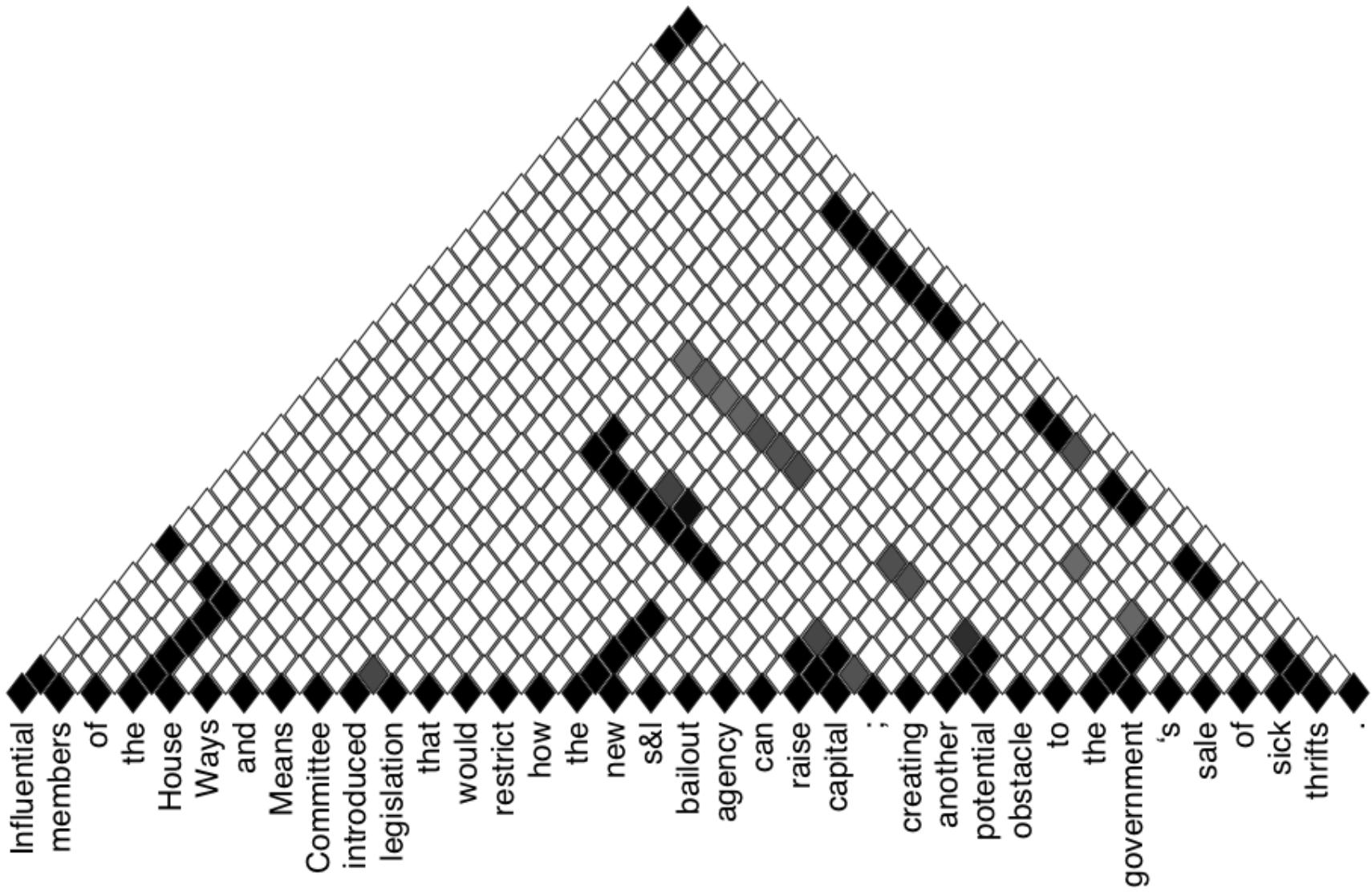
# Hierarchical Pruning

## Pruning with

Parse multiple times, with grammars at different levels of granularity!



# Bracket Posteriors



**1621 min**

**111 min**

**35 min**

**15 min [91.2 F1]**  
**(no search error)**