Statistical NLP
Winter 2011

Lecture 11: Parsing, Part II

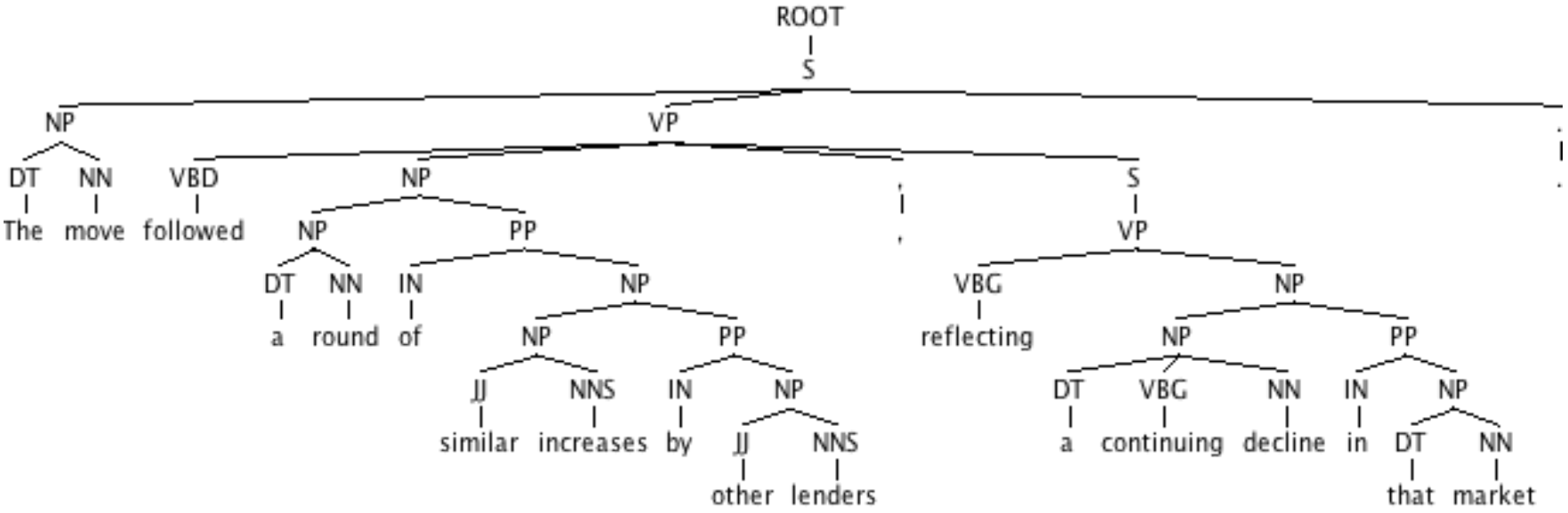
Luke Zettlemoyer - University of Washington

[Most slides from Dan Klein]

Topics

- [Recap] Probabilistic Context Free Grammars
- Agenda-based chart parsing algorithms
- Treebank Parsing (English, edited text)
  - grammar refinement (three ways to figure out what the non-terminals should be...)
  - comparison and evaluation of approaches

Parse Trees



The move followed a round of similar increases by other lenders, reflecting a continuing decline in that market

Probabilistic Context-Free Grammars

- A context-free grammar is a tuple <*N, T, S, R*>
  - *N* : the set of non-terminals
    - Phrasal categories: S, NP, VP, ADJP, etc.
    - Parts-of-speech (pre-terminals): NN, JJ, DT, VB
  - *T* : the set of terminals (the words)
  - *S* : the start symbol
    - Often written as ROOT or TOP
    - *Not* usually the sentence non-terminal S
  - *R* : the set of rules
    - Of the form $X \rightarrow Y_1\ Y_2 \ldots Y_k$, with $X, Y_i \in N$
    - Examples: S → NP VP, VP → VP CC VP
    - Also called rewrites, productions, or local trees
- A PCFG adds:
  - A top-down production probability per rule $P(X \rightarrow Y_1\ Y_2 \ldots Y_k)$

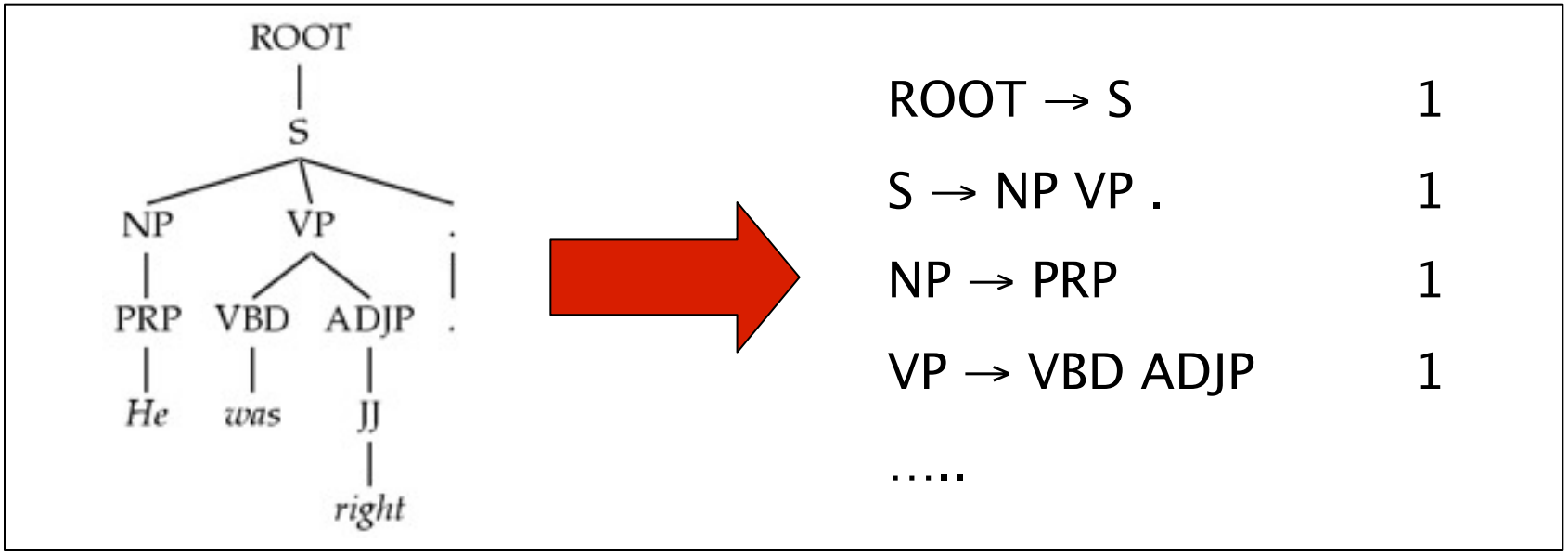
PCFG Joint Distribution

- A PCFG for parse tree T and a sentence $w_{0:n}$:
  - P(T, $w_{0:n}$) = $\prod_{X \to Y_1\, Y_2 \ldots Y_k}$ P($X \to Y_1\, Y_2 \ldots Y_k$)
  - where the rules $X \to Y_1\, Y_2 \ldots Y_k$ range over all nodes in T and the leaves of T define the sentence $w_{0:n}$

- Is a generative model: to sample a pair $(T, w_{0:n})$
  - Initialize: Let T contain the root/start non-terminal S
  - Repeat: Select a non-terminal X in T that is in a leaf position (while one exists)
    - Sample re-write rule with prob. P($X \to Y_1\, Y_2 \ldots Y_k$)
    - Add the nodes $Y_1\, Y_2 \ldots Y_k$ as children of X in T

Treebank Grammars

- Need a PCFG for broad coverage parsing.
- Can take a grammar right off the trees (doesn't work well):

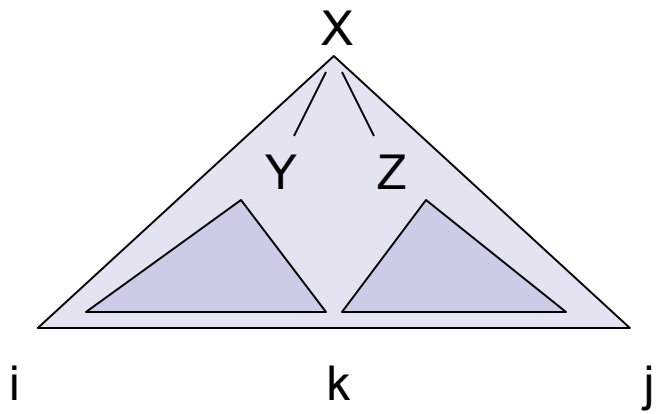


- Still have to deal with all of the standard machine learning problems
  - smoothing, data sparsity, etc.
- We will also talk later about other challenges, specific to parsing

A Bottom-Up Parser (CKY)

- Can also organize things bottom-up

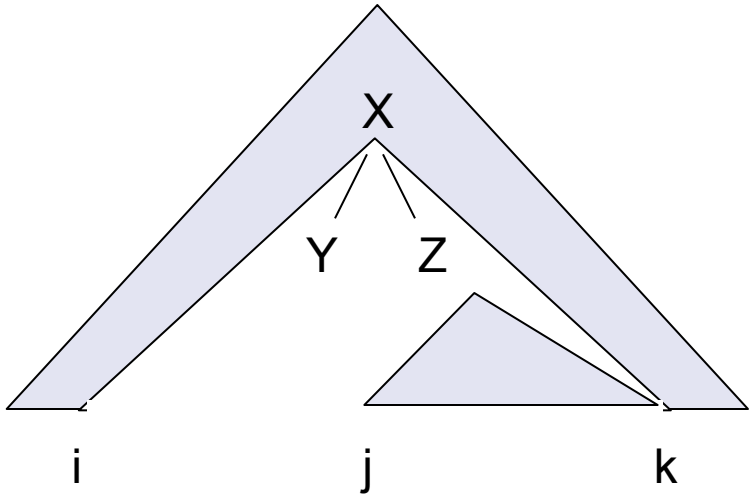
```
bestScore(s)
  for (i : [0,n-1])
    for (X : tags[s[i]])
      score[X][i][i+1] =
          tagScore(X,s[i])
  for (diff : [2,n])
    for (i : [0,n-diff])
      j = i + diff
      for (X->YZ : rule)
        for (k : [i+1, j-1])
          score[X][i][j] = max score[X][i][j],
                               score(X->YZ) *
                               score[Y][i][k] *
                               score[Z][k][j]
```

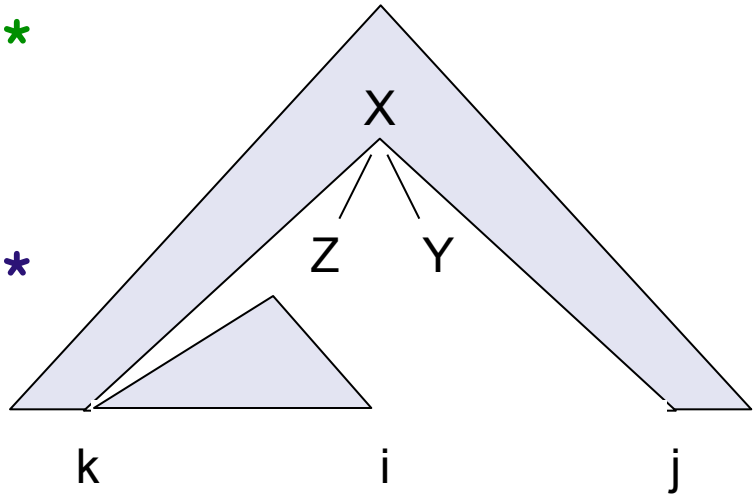


*CKY/CYK: Cocker–Kasami–Younger algorithm [1970,1965,1967]

Best Outside Scores

```
bestOutside(Y,i,j,s)
 if (i==0 && j==n)
   return 1.0
 else
   return max
    max     score(X->YZ) *
  X->YZ,k   bestOutside(X,i,k,s) *
            bestScore(Z,j,k,s)
    max     score(X->ZY) *
  X->ZY,k   bestOutside(X,k,j,s) *
            bestScore(Z,k,i,s)
```





Agenda-Based Parsing

- Agenda-based parsing is like graph search (but over a hypergraph)
- Concepts:
  - Numbering: indices go between words
  - “Edges” or items: spans with labels, e.g. PP[3,5], represent the sets of trees over those words rooted at that label (cf. search states)
    - in probabilistic setting each edge has a score (real number) and, optionally, a backpointer for finding the best tree
  - A chart: records edges we’ve expanded (cf. closed set)
  - An agenda: a queue which holds edges (cf. a fringe or open set)

Word Items

- Building an item for the first time is called discovery. Items go into the agenda on discovery.
- To initialize, we discover all word items (with score 1.0).

AGENDA

critics[0,1], write[1,2], reviews[2,3], with[3,4], computers[4,5]

CHART [EMPTY]

Unary Projection

- When we pop a word item, the lexicon tells us the tag item successors (and scores) which go on the agenda

critics[0,1]	write[1,2]	reviews[2,3]	with[3,4]	computers[4,5]
NNS[0,1]	VBP[1,2]	NNS[2,3]	IN[3,4]	NNS[4,5]

critics write reviews with computers

Item Successors

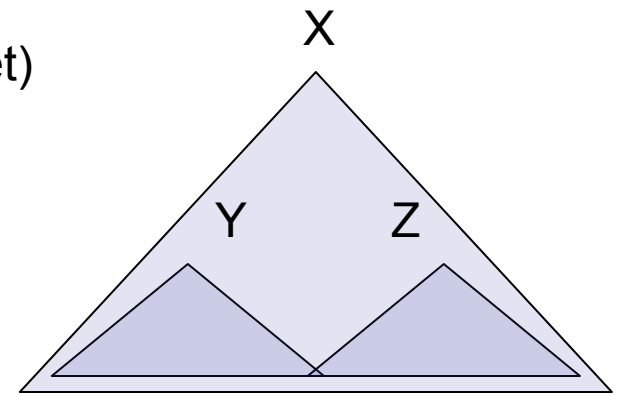
- When we pop items off of the agenda:
  - Graph successors: unary projections (NNS → critics, NP → NNS)

$$Y[i,j] \text{ with } X \rightarrow Y \text{ forms } X[i,j]$$

  - Hypergraph successors: combine with items already in our chart

$$Y[i,j] \text{ and } Z[j,k] \text{ with } X \rightarrow Y\ Z \text{ form } X[i,k]$$

  - Enqueue / promote resulting items (if not in chart already)
  - Record backtraces as appropriate
  - Stick the popped edge in the chart (closed set)

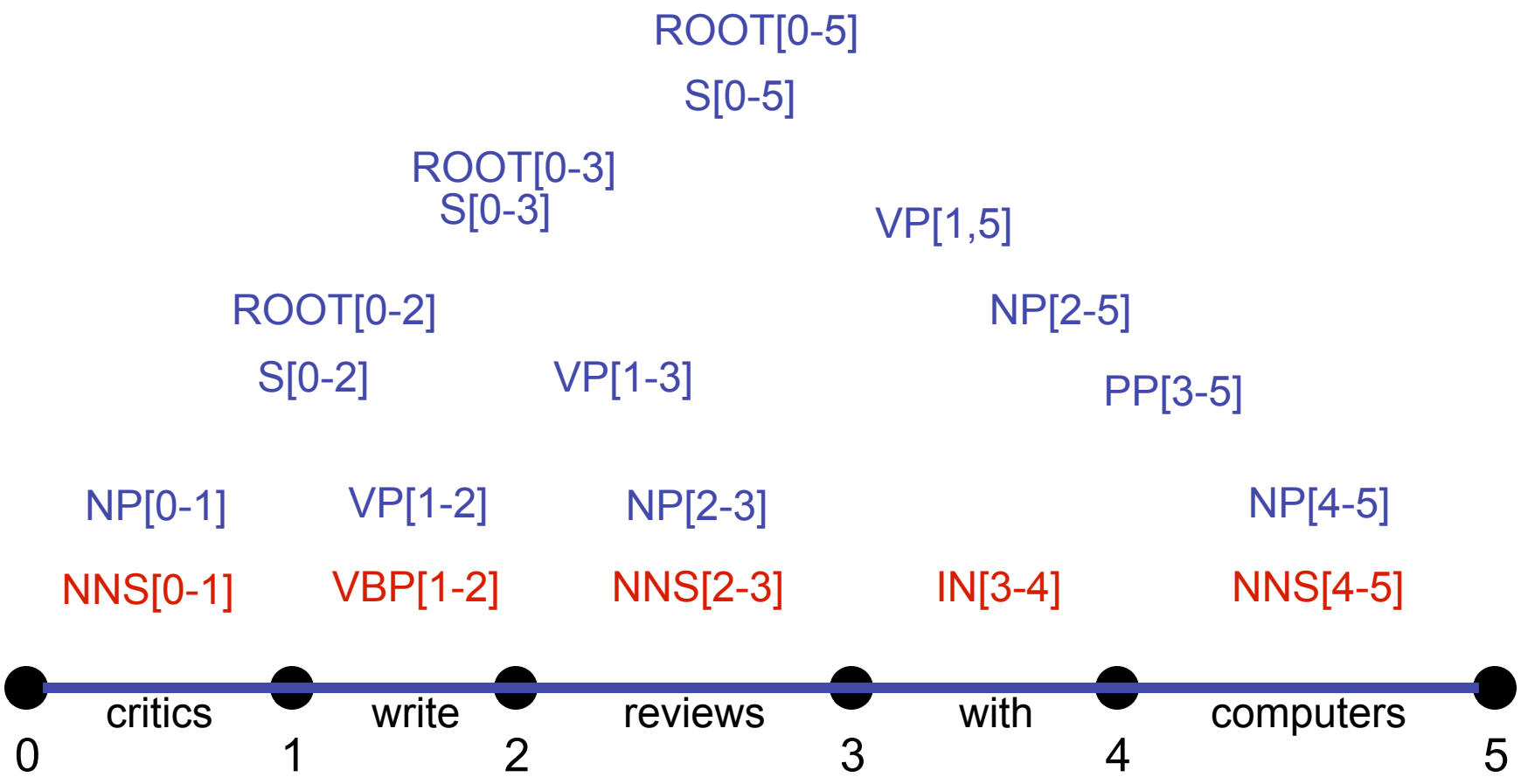


- Queries a chart must support:
  - Is edge X:[i,j] in the chart? (What score?)
  - What edges with label Y end at position j?
  - What edges with label Z start at position i?

An Example

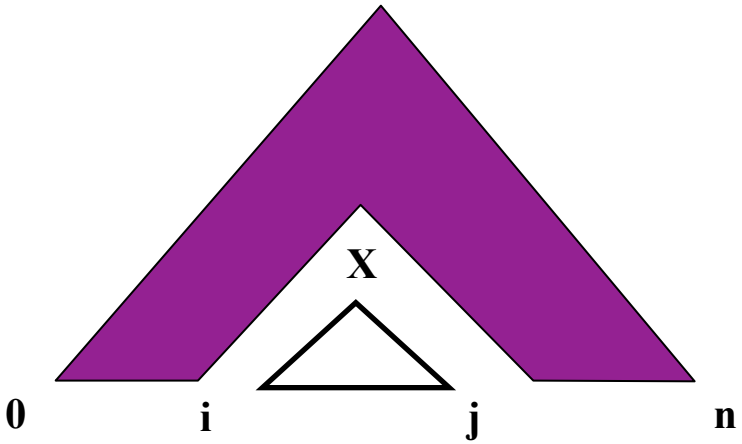
NNS[0,1] VBP[1,2] NNS[2,3] IN[3,4] NNS[3,4] NP[0,1] VP[1,2] NP[2,3] NP[4,5] S[0,2]

VP[1,3] PP[3,5] ROOT[0,2] S[0,3] VP[1,5] NP[2,5] ROOT[0,3] S[0,5] ROOT[0,5]



UCS / A*

- With weighted edges, order matters
  - Must expand optimal parse from bottom up (subparses first)
  - CKY does this by processing smaller spans before larger ones
  - UCS pops items off the agenda in order of decreasing inside (Viterbi) score
  - A* search also well defined



- You can also speed up the search without sacrificing optimality
  - Can select which items to process first
  - Can do with any “figure of merit” [Charniak 98]
  - If your figure-of-merit is a valid A* heuristic, no loss of optimiality [Klein and Manning 03]

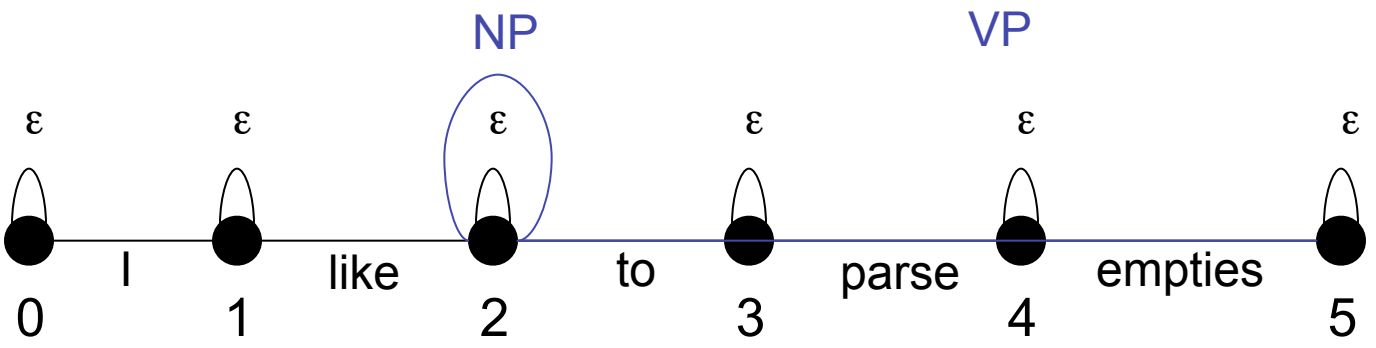
Empty Elements

- Sometimes we want to posit nodes in a parse tree that don't contain any pronounced words:

I want you to parse this sentence

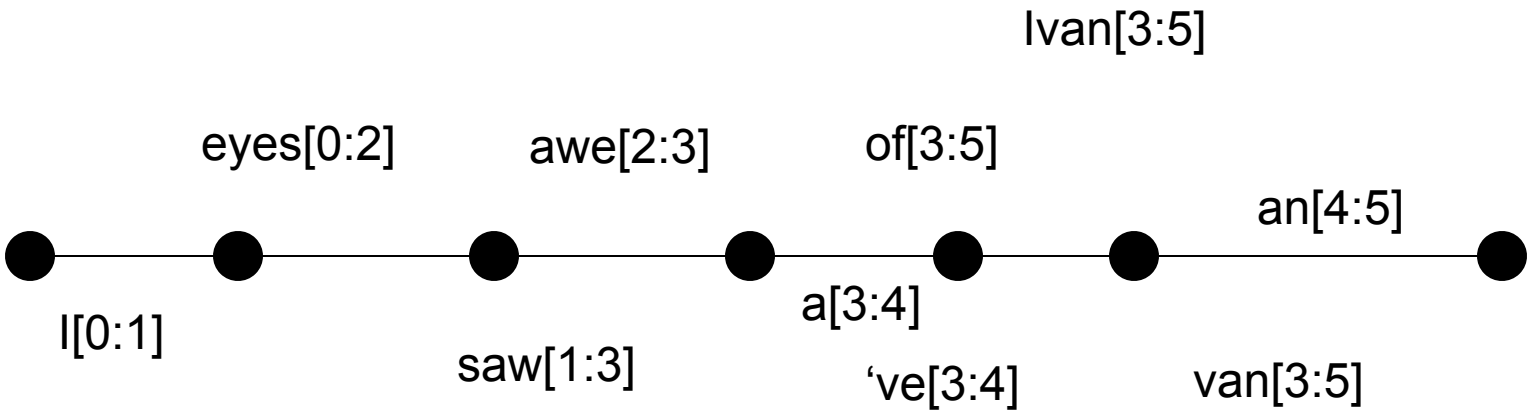
I want [ ] to parse this sentence

- These are easy to add to a chart parser!
  - For each position i, add the "word" edge ε:[i,i]
  - Add rules like NP → ε to the grammar
  - That's it!



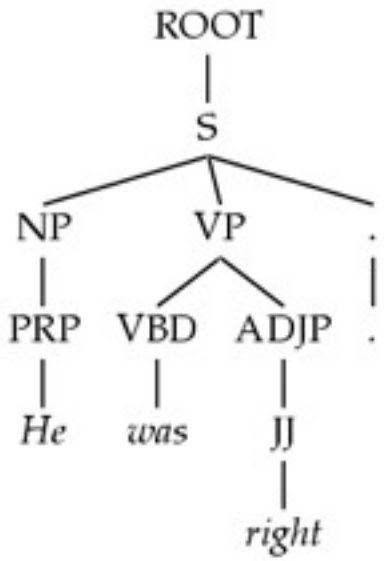
(Speech) Lattices

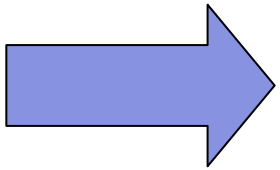
- When working with speech, we generally don't know how many words there are, or where they break.
- There was nothing magical about words spanning exactly one position.
- We can represent the possibilities as a lattice and parse these just as easily.



Treebank PCFGs [Charniak 96]

- Use PCFGs for broad coverage parsing
- Can take a grammar right off the trees (doesn't work well):





ROOT → S 1

S → NP VP . 1

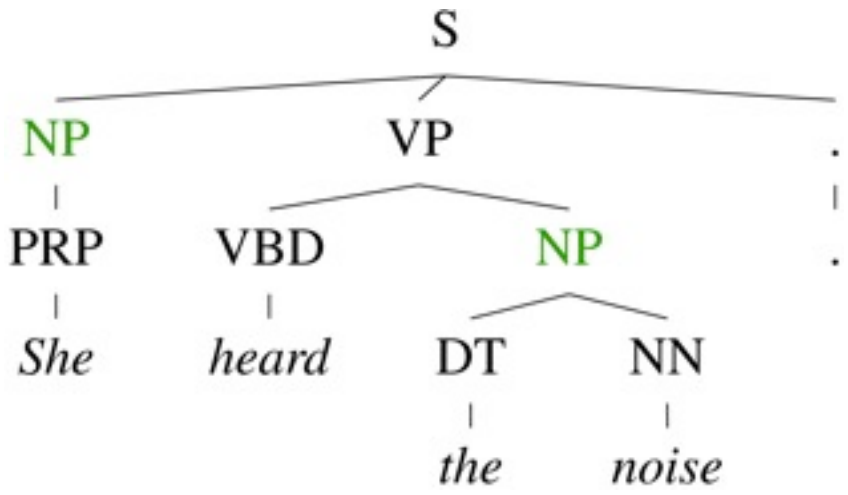
NP → PRP 1

VP → VBD ADJP 1

…..

Model	*F1*
Baseline	72.0

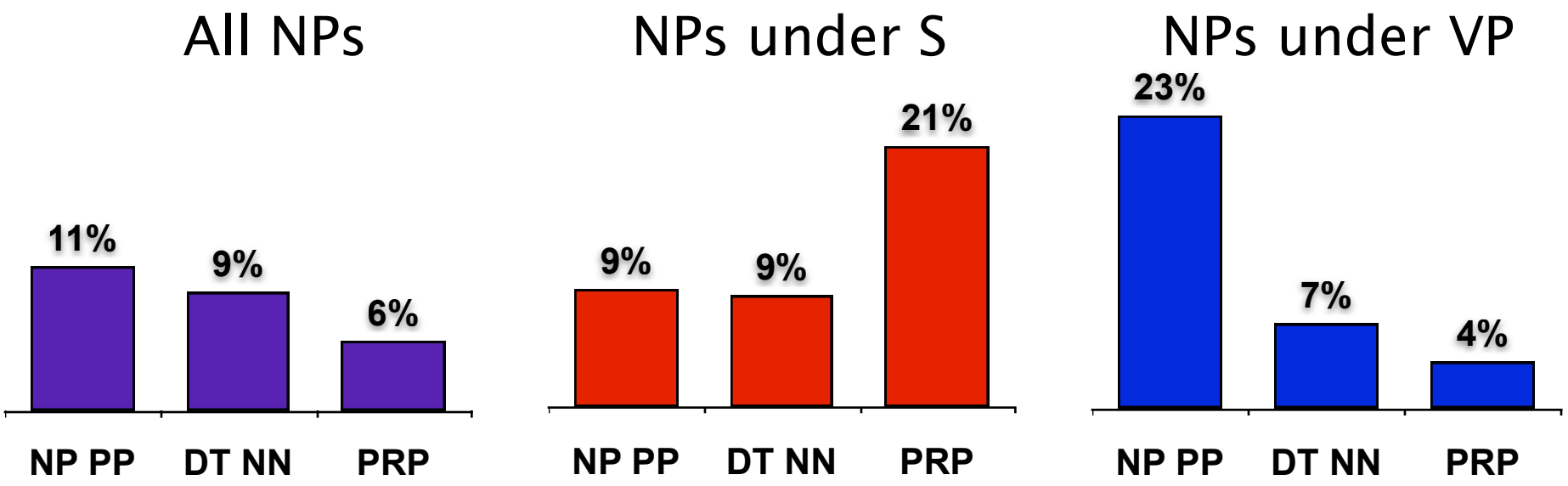
Conditional Independence?



- Not every NP expansion can fill every NP slot
  - A grammar with symbols like “NP” won’t be context-free
  - Statistically, conditional independence too strong

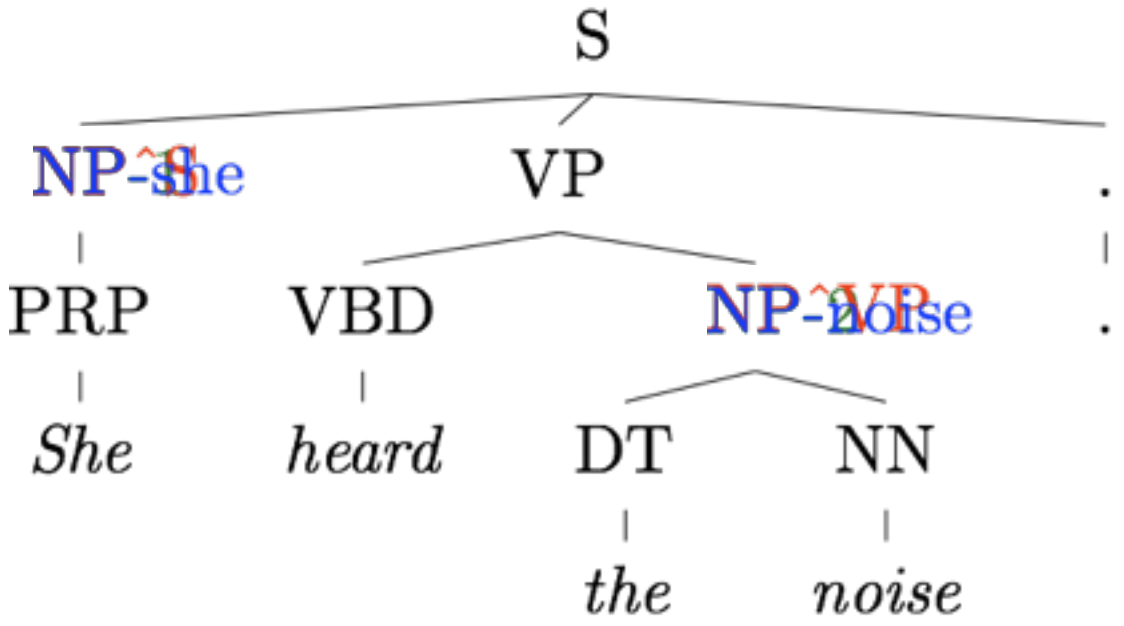
Non-Independence

- Independence assumptions are often too strong.



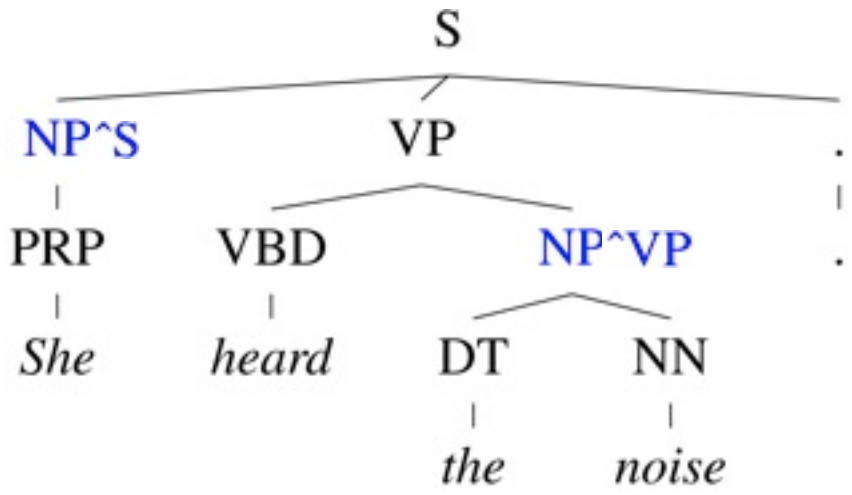
- Example: the expansion of an NP is highly dependent on the parent of the NP (i.e., subjects vs. objects).
- Also: the subject and object expansions are correlated!

Grammar Refinement



- Structure Annotation [Johnson ’98, Klein&Manning ’03]
- Lexicalization [Collins ’99, Charniak ’00]
- Latent Variables [Matsuzaki et al. 05, Petrov et al. ’06]

The Game of Designing a Grammar



- Annotation refines base treebank symbols to improve statistical fit of the grammar
  - Structural annotation

Typical Experimental Setup

- Corpus: Penn Treebank, WSJ

Training:	sections	02-21
Development:	section	22 (here, first 20 files)
Test:	section	23

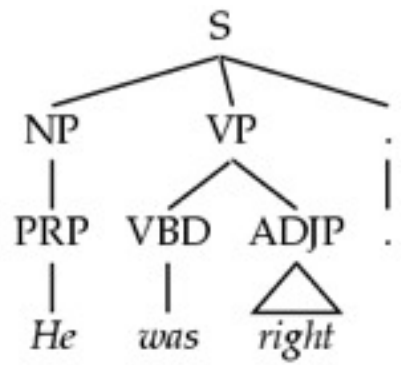
- Accuracy – F1: harmonic mean of per-node labeled precision and recall.
- Here: also size – number of symbols in grammar.
  - Passive / complete symbols: NP, NP^S
  - Active / incomplete symbols: NP → NP CC •

Vertical Markovization

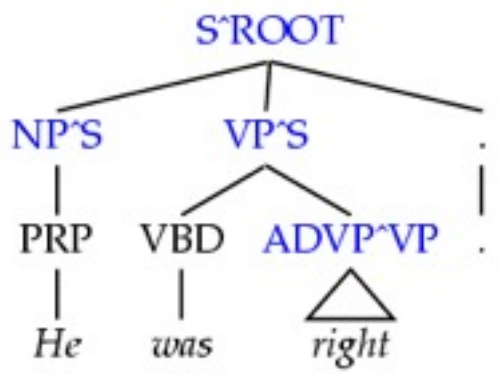
- Vertical Markov order: rewrites depend on past k ancestor nodes.

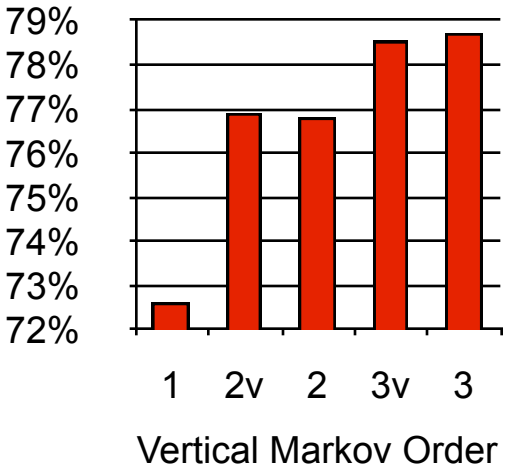
  (cf. parent annotation)

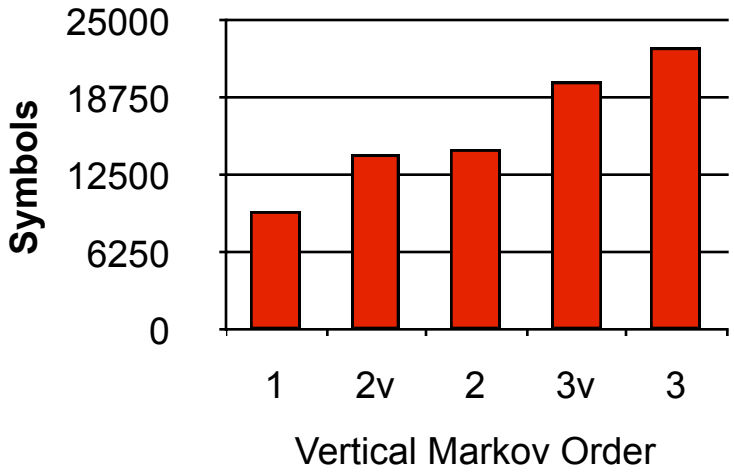
Order 1



Order 2

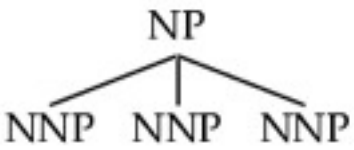


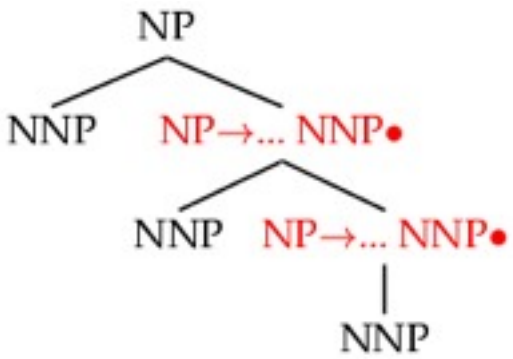




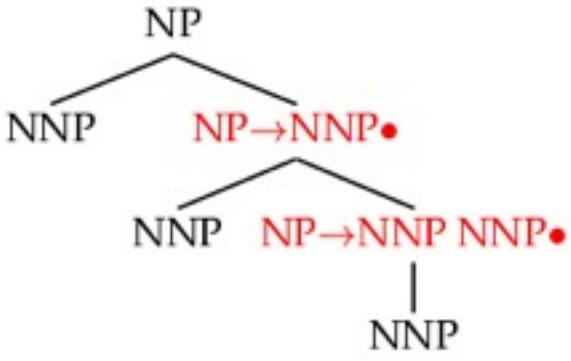
Horizontal Markovization

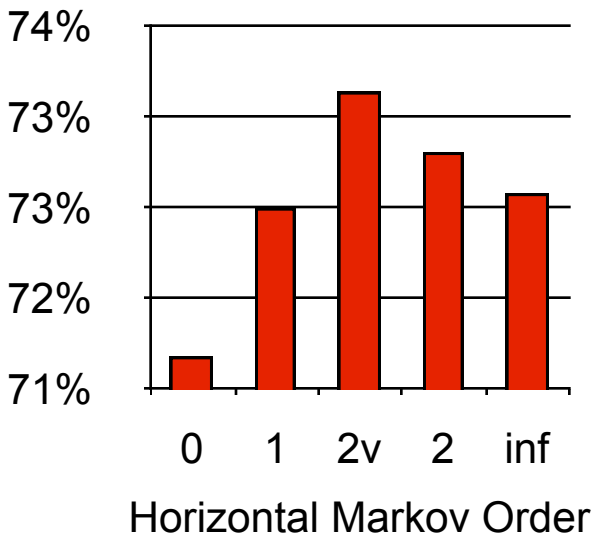
Order 1

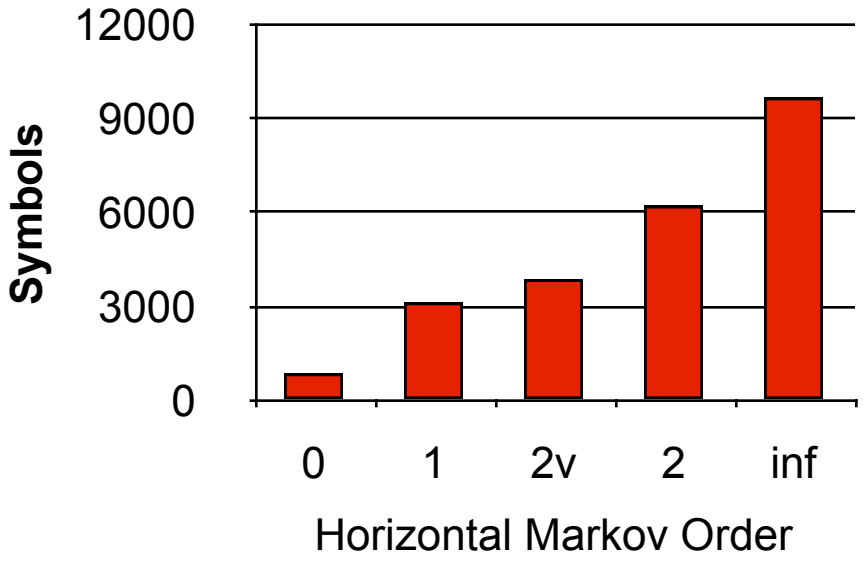




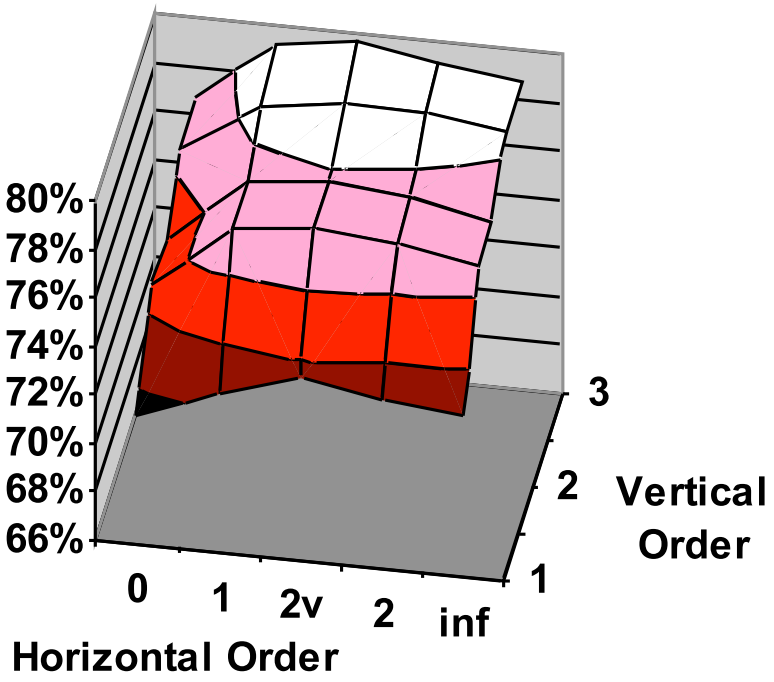
Order ∞

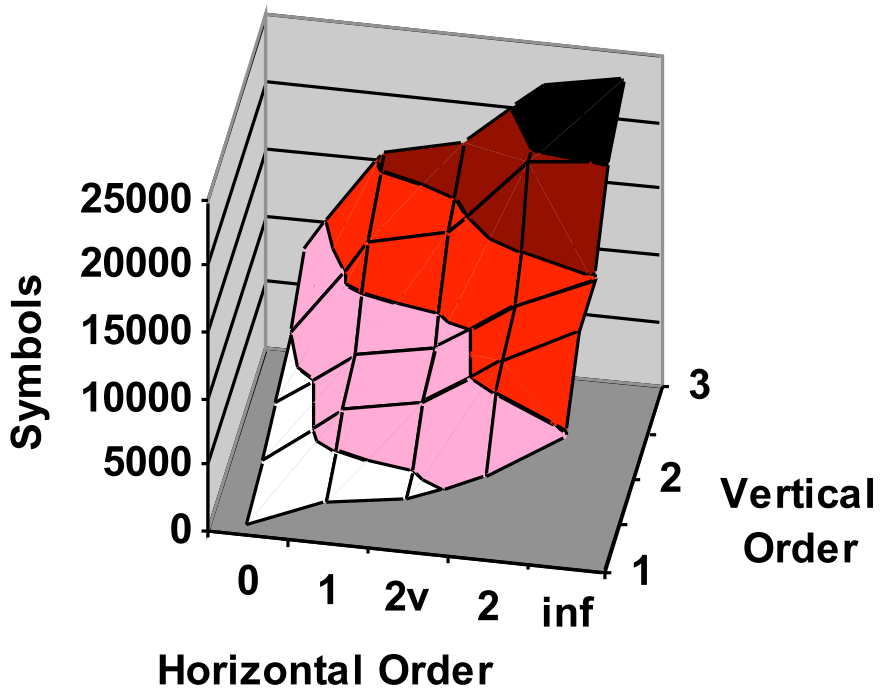






Vertical and Horizontal



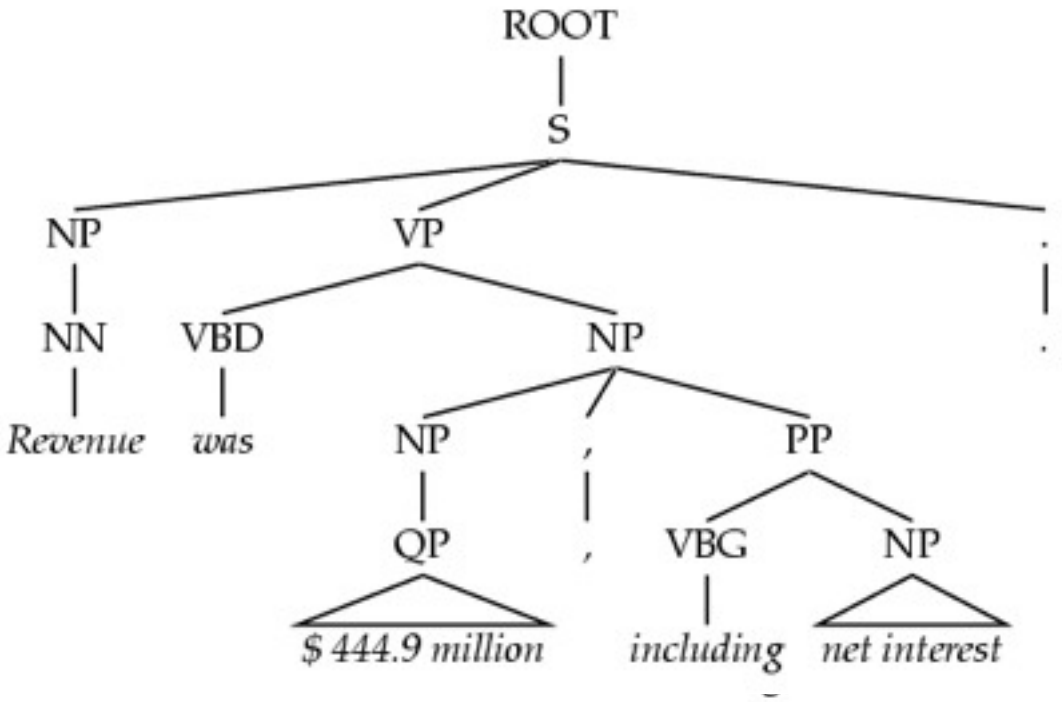


- Examples:
  - Raw treebank: v=1, h=∞
  - Johnson 98: v=2, h=∞
  - Collins 99: v=2, h=2
  - Best F1: v=3, h=2v

Model	F1	Size
Base: v=h=2v	77.8	7.5K

Unary Splits

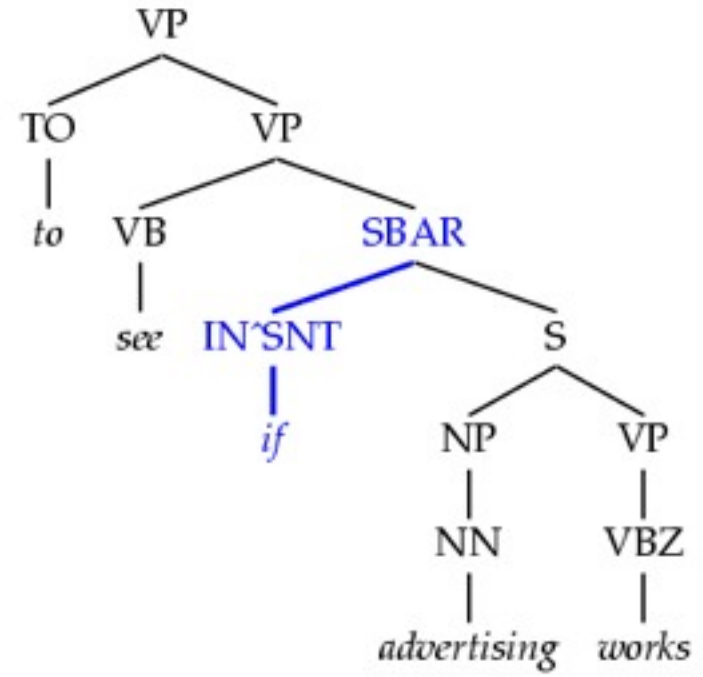
- Problem: unary rewrites used to transmute categories so a high-probability rule can be used.
- Solution: Mark unary rewrite sites with -U



Annotation	F1	Size
Base	77.8	7.5K
UNARY	78.3	8.0K

Tag Splits

- Problem: Treebank tags are too coarse.
- Example: Sentential, PP, and other prepositions are all marked IN.
- Partial Solution:
  - Subdivide the IN tag.



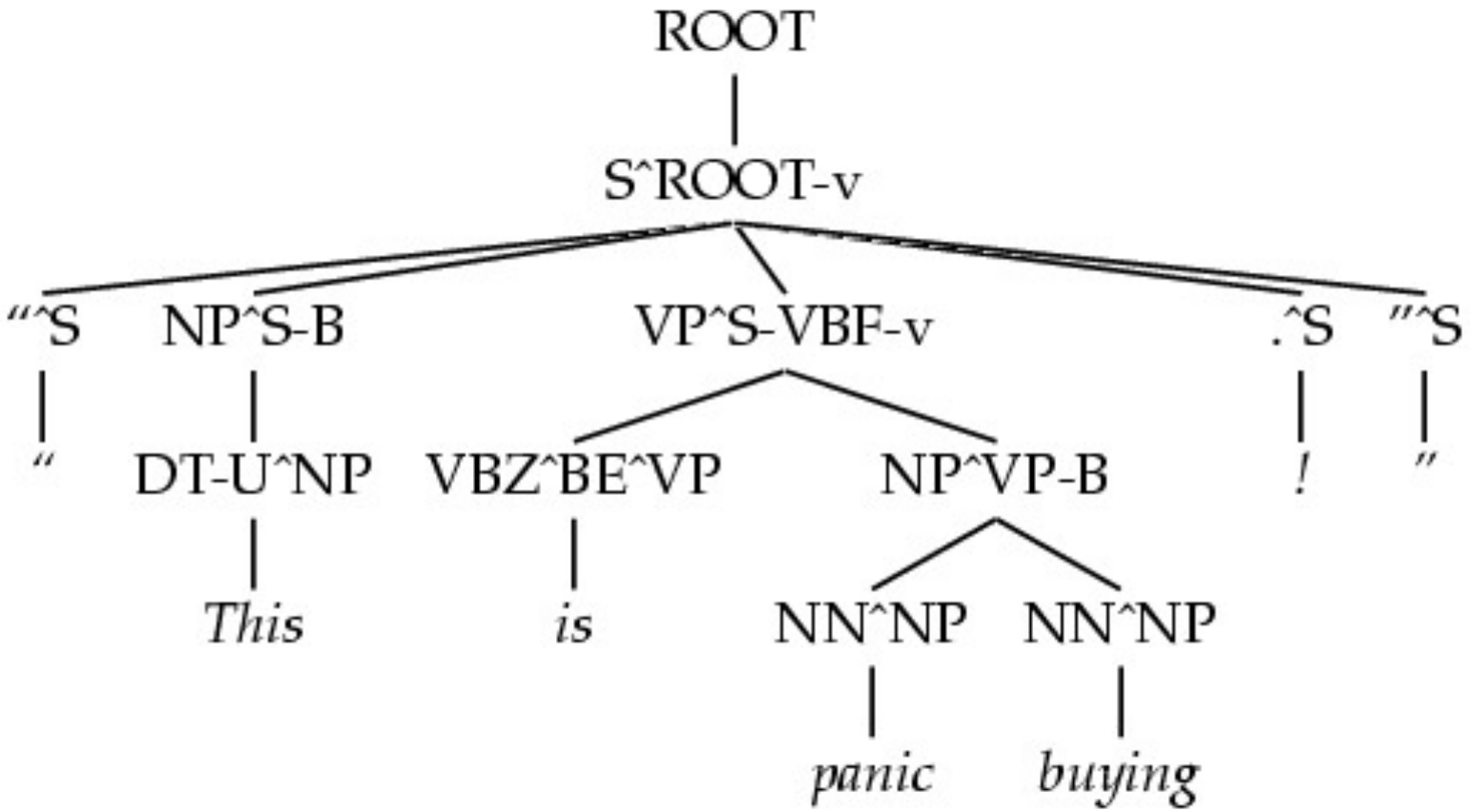
Annotation	F1	Size
Previous	78.3	8.0K
SPLIT-IN	80.3	8.1K

Other Tag Splits

- UNARY-DT: mark demonstratives as DT^U ("the X" vs. "those")
- UNARY-RB: mark phrasal adverbs as RB^U ("quickly" vs. "very")
- TAG-PA: mark tags with non-canonical parents ("not" is an RB^VP)
- 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

A Fully Annotated (Unlex) Tree

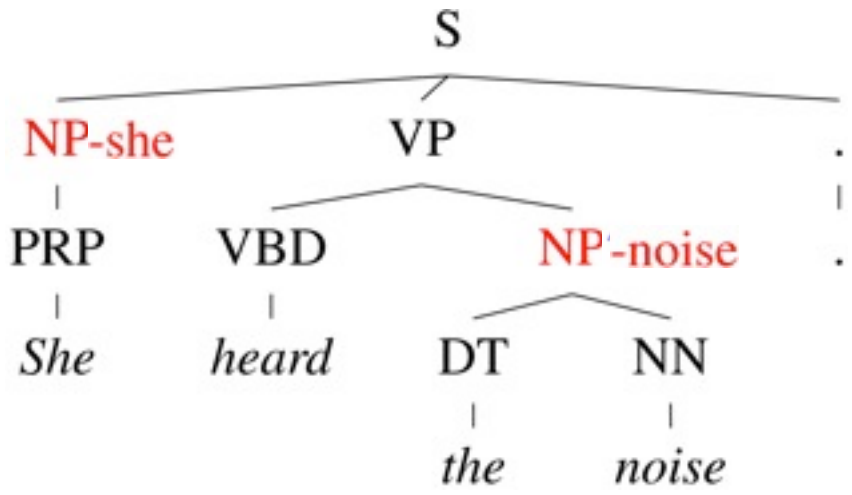


Some Test Set Results

Parser	LP	LR	**F1**	CB	0 CB
Magerman 95	84.9	84.6	**84.7**	1.26	56.6
Collins 96	86.3	85.8	**86.0**	1.14	59.9
Unlexicalized	86.9	85.7	**86.3**	1.10	60.3
Charniak 97	87.4	87.5	**87.4**	1.00	62.1
Collins 99	88.7	88.6	**88.6**	0.90	67.1

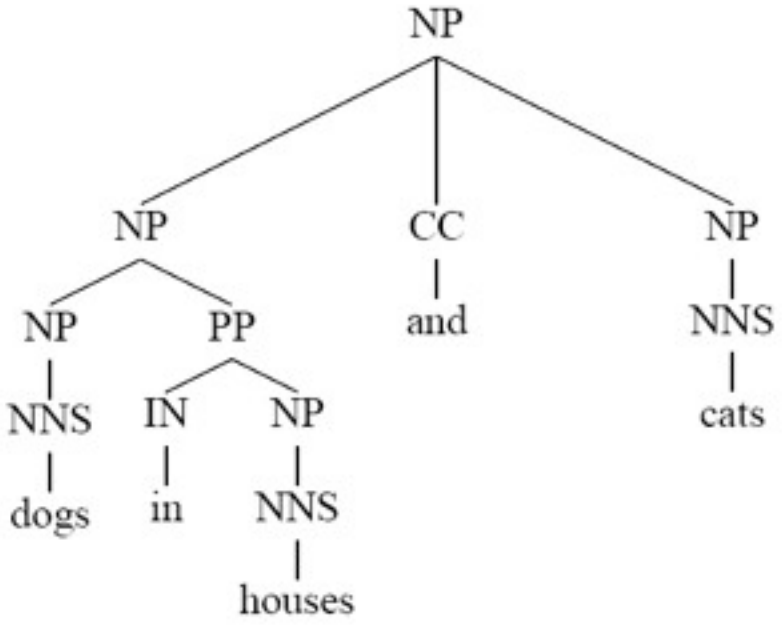
- Beats “first generation” lexicalized parsers.
- Lots of room to improve – more complex models next.

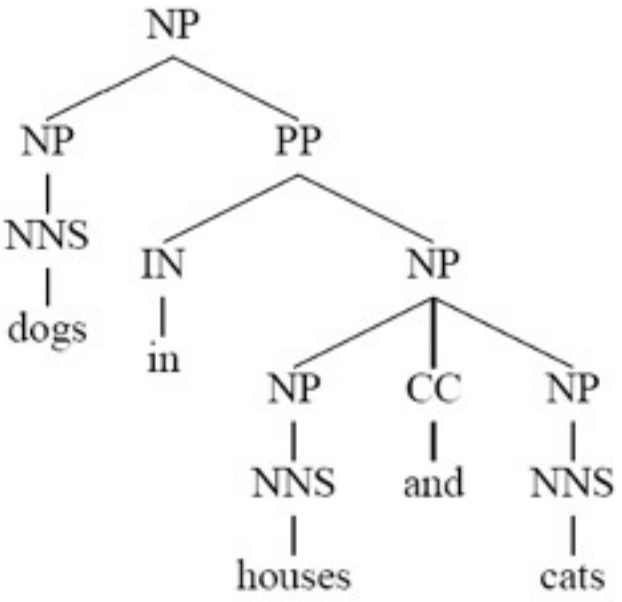
The Game of Designing a Grammar



- Annotation refines base treebank symbols to improve statistical fit of the grammar
  - Structural annotation [Johnson '98, Klein and Manning 03]
  - Head lexicalization [Collins '99, Charniak '00]

Problems with PCFGs

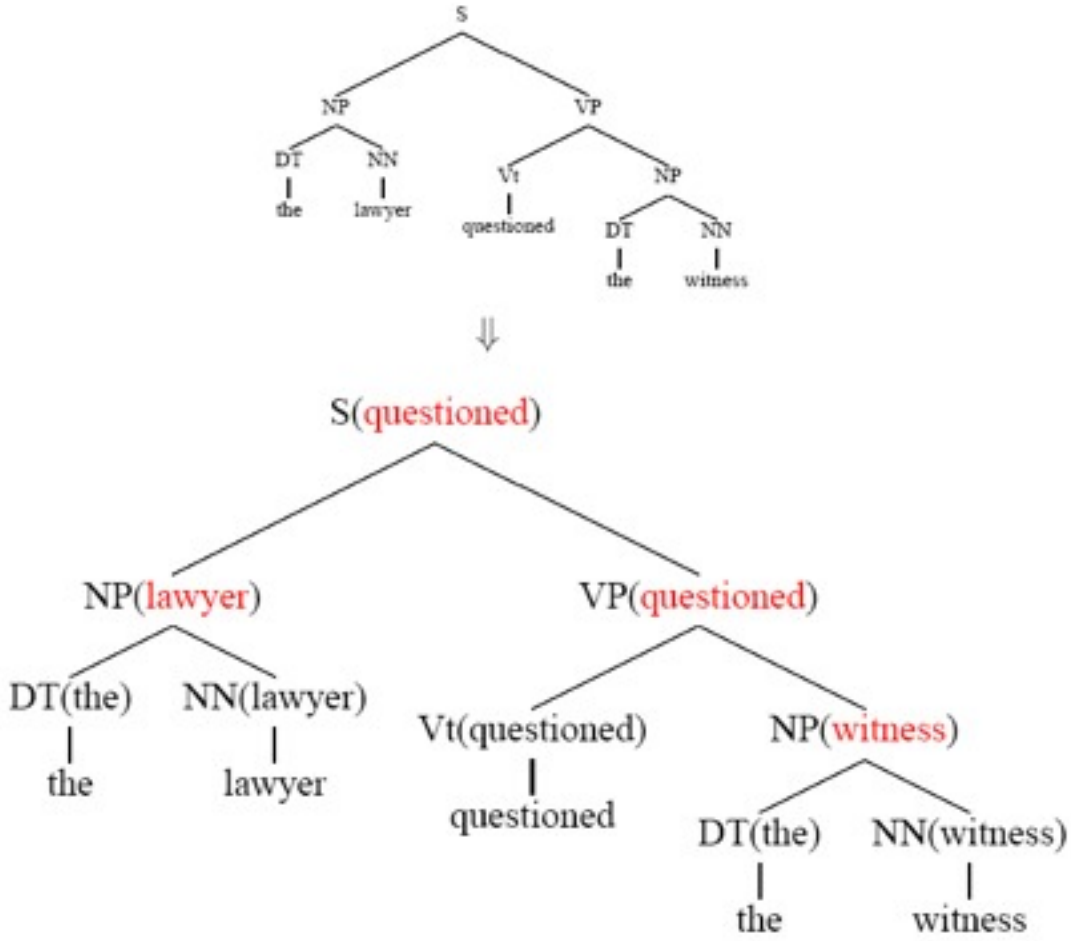




- What's different between basic PCFG scores here?
- What (lexical) correlations need to be scored?

Lexicalized Trees

- Add "headwords" to each phrasal node
  - Headship not in (most) treebanks
  - Usually *use head rules*, e.g.:
    - NP:
      - Take leftmost NP
      - Take rightmost N*
      - Take rightmost JJ
      - Take right child
    - VP:
      - Take leftmost VB*
      - Take leftmost VP
      - Take left child



Lexicalized PCFGs?

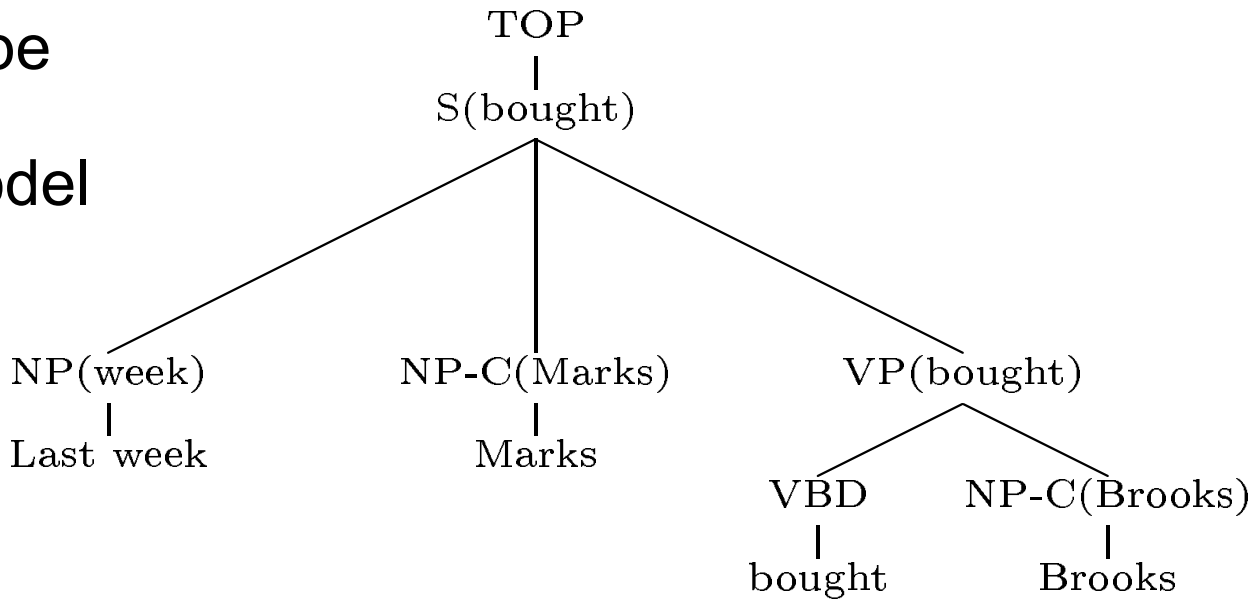
- Problem: we now have to estimate probabilities like

```
VP(saw) -> VBD(saw) NP-C(her) NP(today)
```

  - Never going to get these atomically off of a treebank
  - Solution: break up derivation into smaller steps

Complement / Adjunct Distinction

- **warning** - can be tricky, and most parsers don't model the distinction

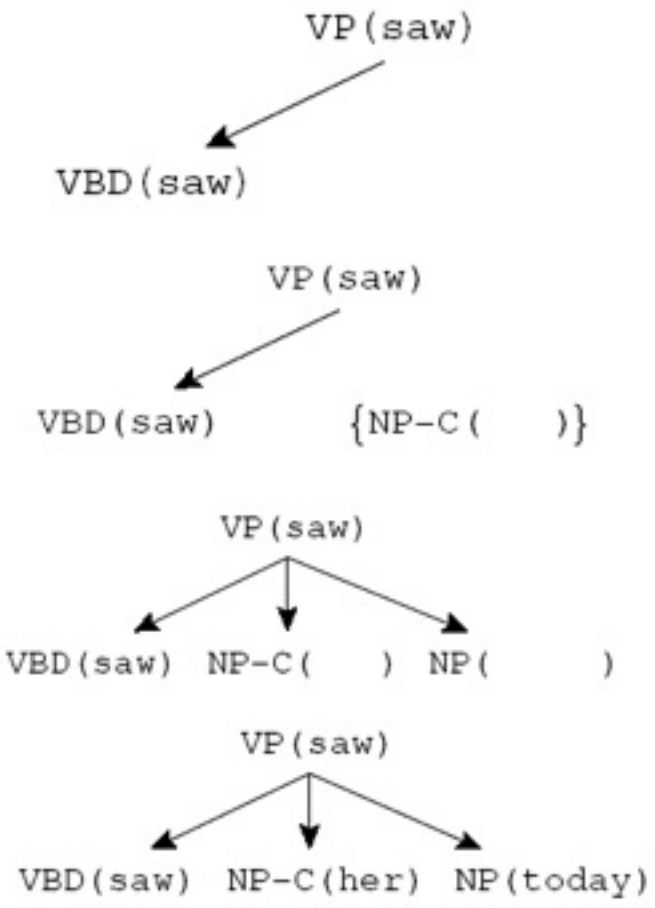


- Complement: defines a property/argument (often obligatory), ex: [[capitol] [of Rome]]
- Adjunct: modifies / describes something (always optional), ex: [quickly [ran]]
- A Test for Adjuncts: [X Y] --> can claim X and Y
  - [they ran and it happened quickly] vs. [capitol and it was of Rome]

[Collins 99]

Lexical Derivation Steps

- Main idea: define a linguistically-motivated Markov process for generating children given the parent



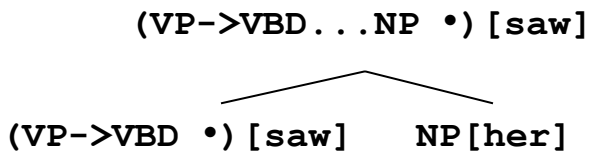
Step 1: Choose a head tag and word

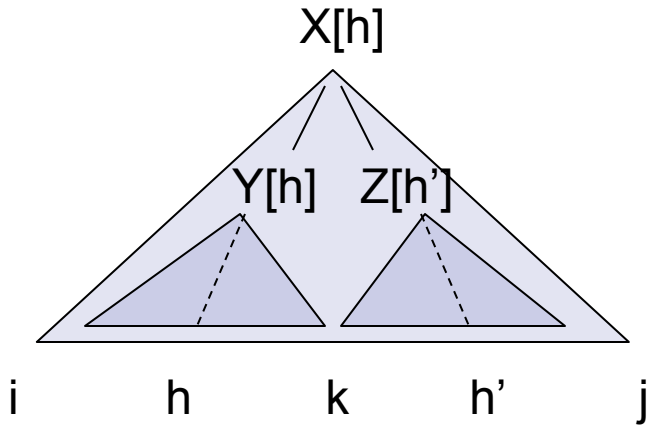
Step 2: Choose a complement bag

Step 3: Generate children (incl. adjuncts)

Step 4: Recursively derive children

Lexicalized CKY

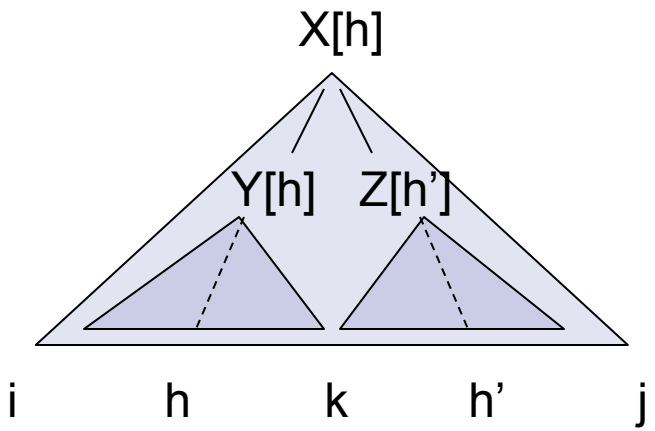





```
bestScore(X,i,j,h)
  if (j = i+1)
    return tagScore(X,s[i])
  else
    return
      max   max   score(X[h]->Y[h] Z[h']) *
        k,h',     bestScore(Y,i,k,h) *
        X->YZ     bestScore(Z,k,j,h')
          max     score(X[h]->Y[h'] Z[h]) *
        k,h,      bestScore(Y,i,k,h') *
        X->YZ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

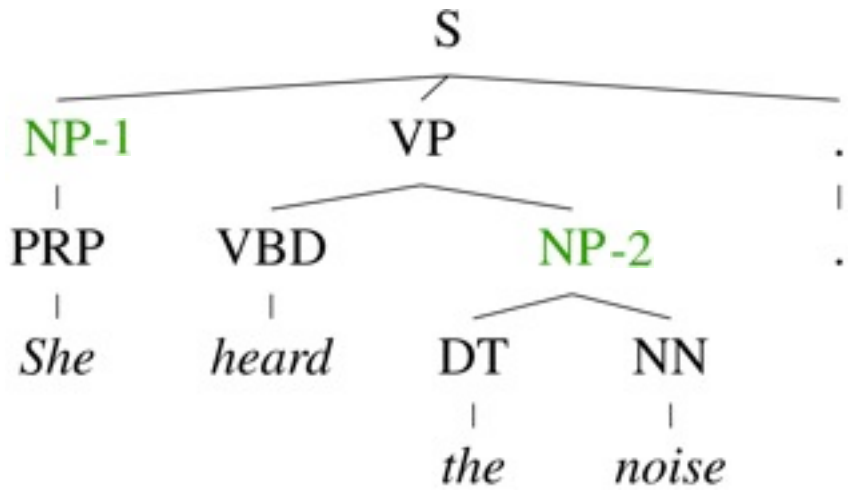
Pruning with Beams

- 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 <i,j>.
  - 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)

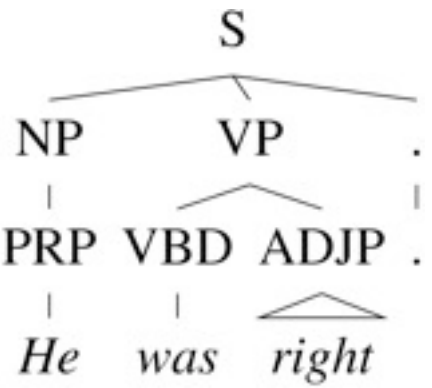
The Game of Designing a Grammar



- 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 Annotation

- 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

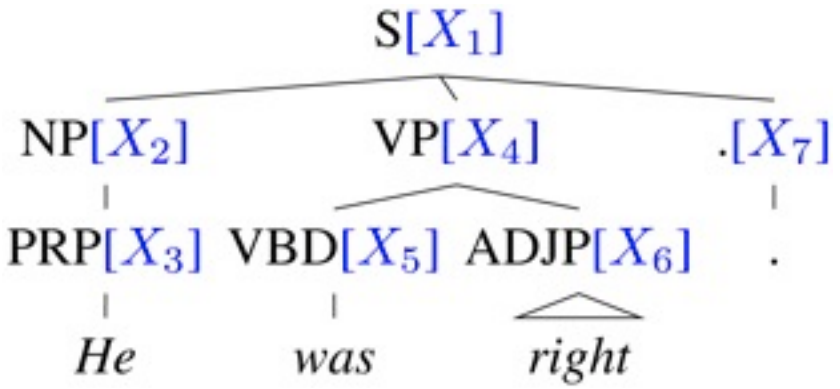


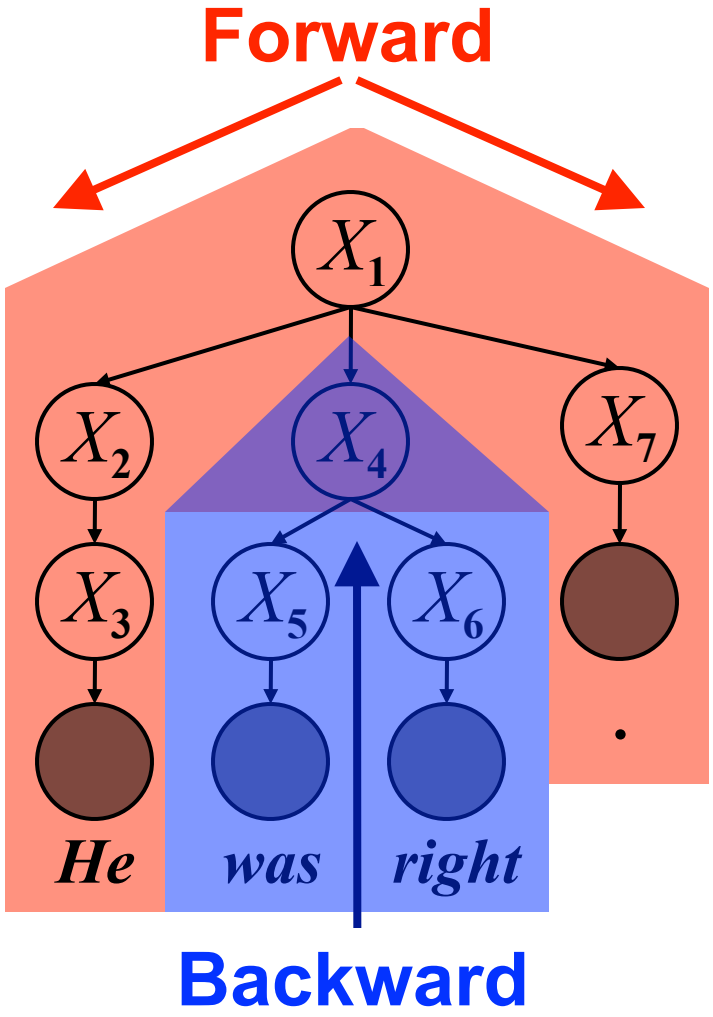
Model	*F1*
Naïve Treebank Grammar	72.6
Klein & Manning '03	86.3

Learning Latent Annotations

Latent Annotations:

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





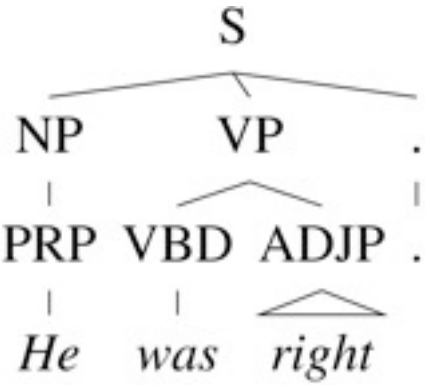
Can learn with EM: like Forward-Backward for HMMs.

Automatic Annotation Induction

- Advantages:
  - Automatically learned:

    Label *all* nodes with latent variables.

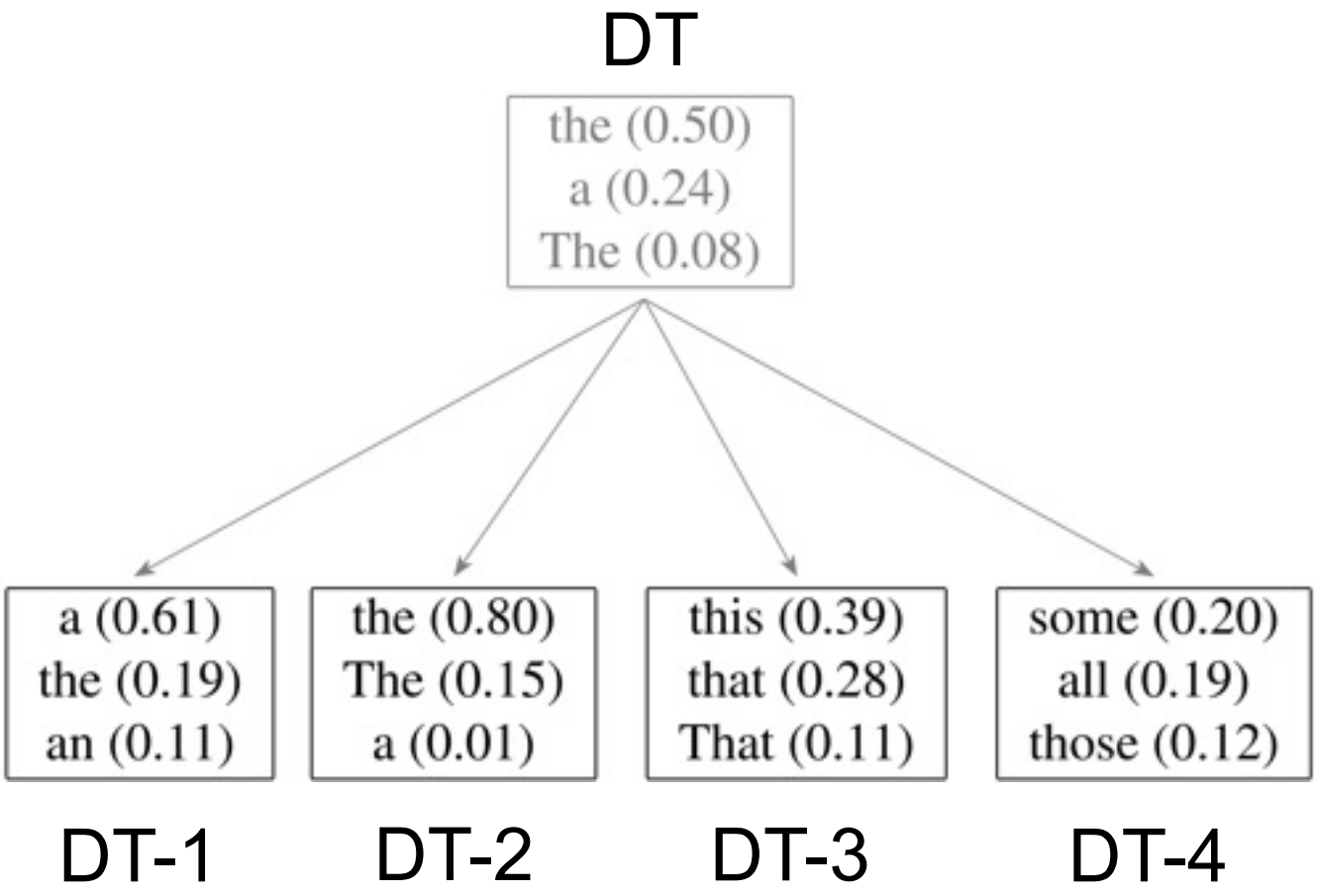
    Same number k of subcategories for all categories.



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

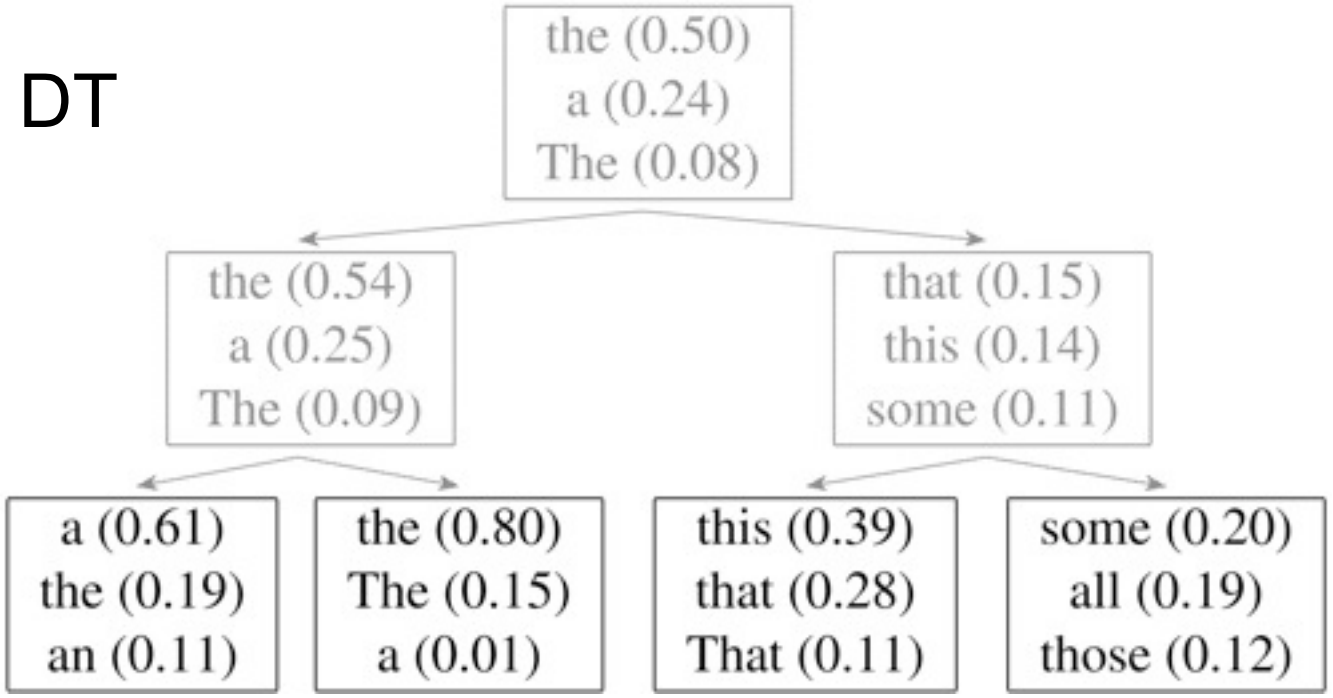
Model	*F1*
Klein & Manning '03	86.3
Matsuzaki et al. '05	86.7

Refinement of the DT tag



Hierarchical refinement

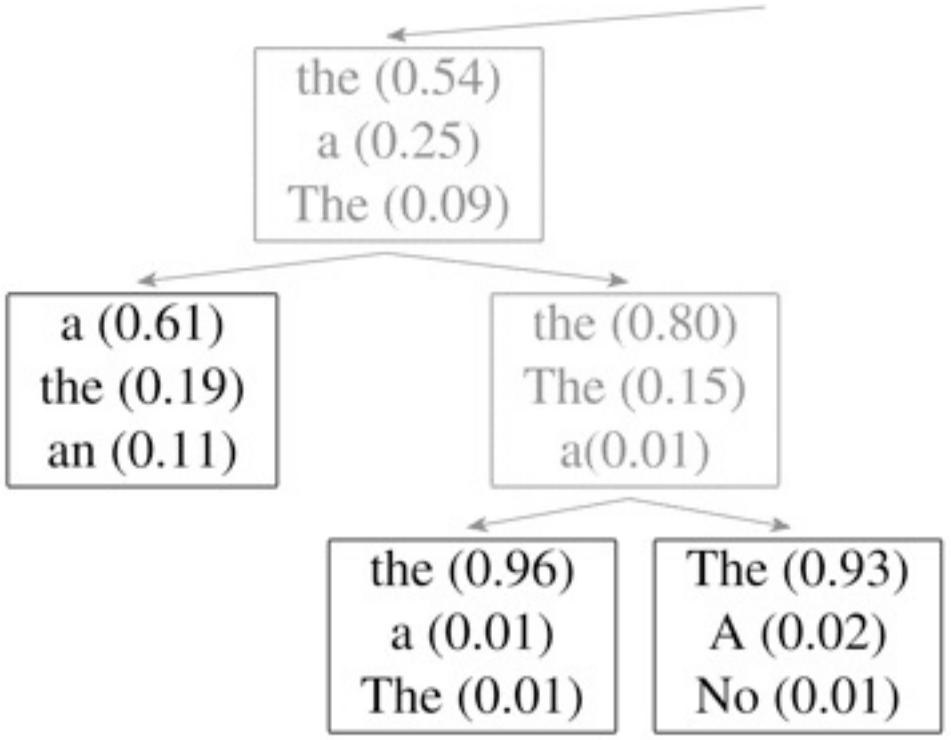
- Repeatedly learn more fine-grained subcategories
  - start with two (per non-terminal), then keep splitting
  - initialize each EM run with the output of the last



[Petrov et al. 06]

Adaptive Splitting

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

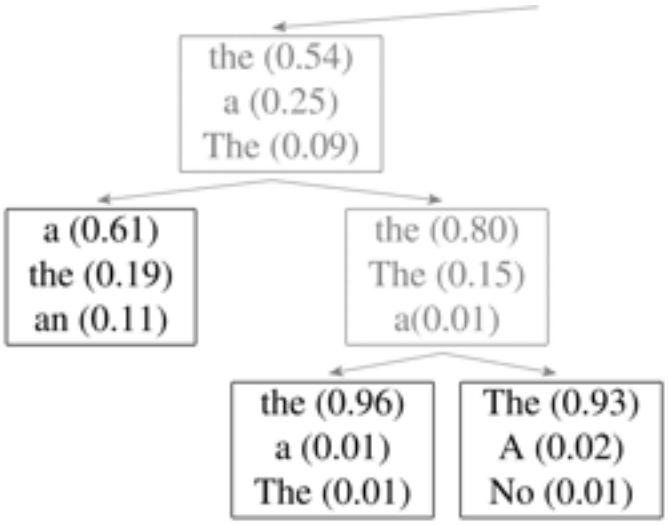


Adaptive Splitting

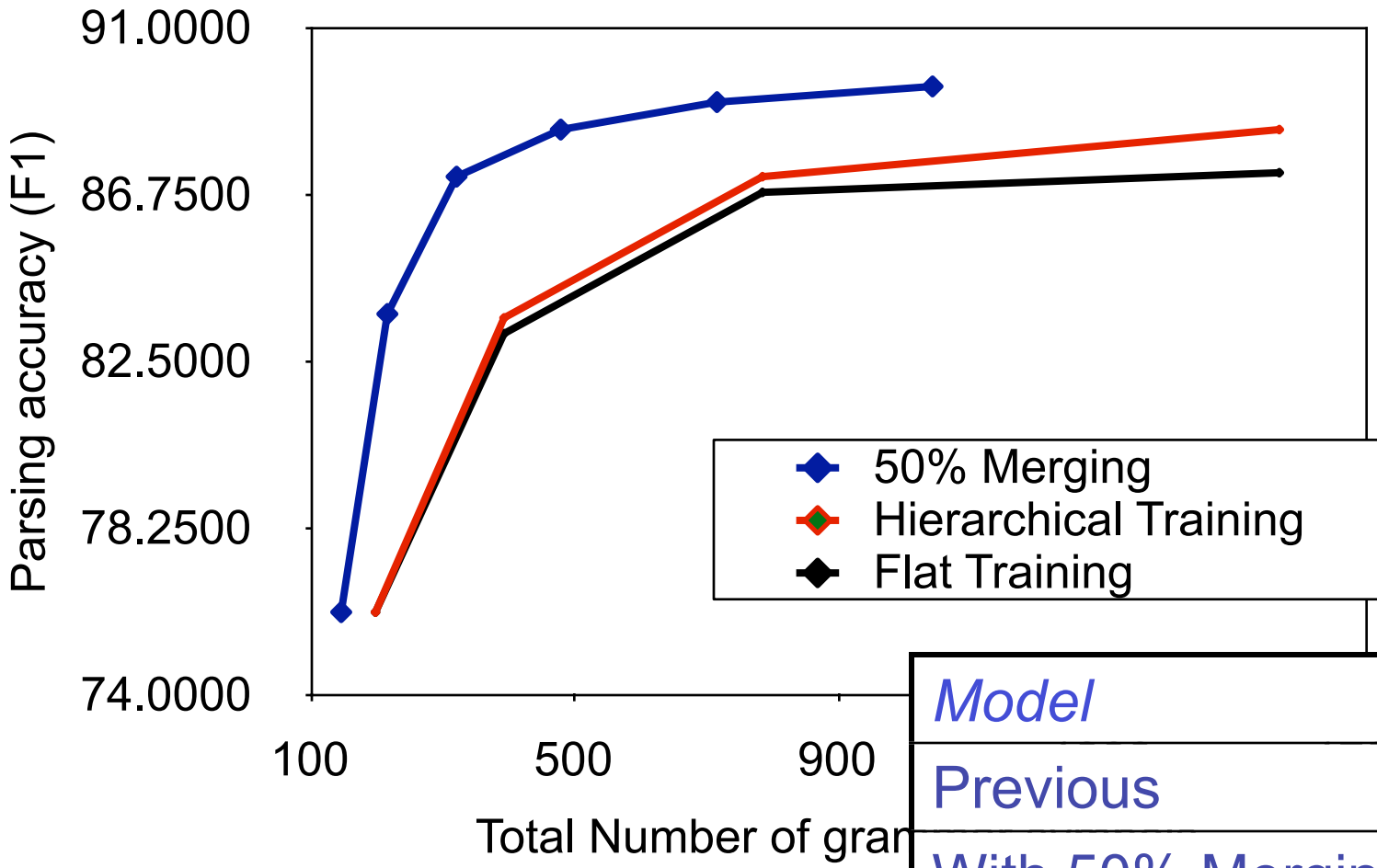
- Evaluate loss in likelihood from removing each split =

$$\frac{\textit{Data likelihood with split reversed}}{\textit{Data likelihood with split}}$$

- No loss in accuracy when 50% of the splits are reversed.

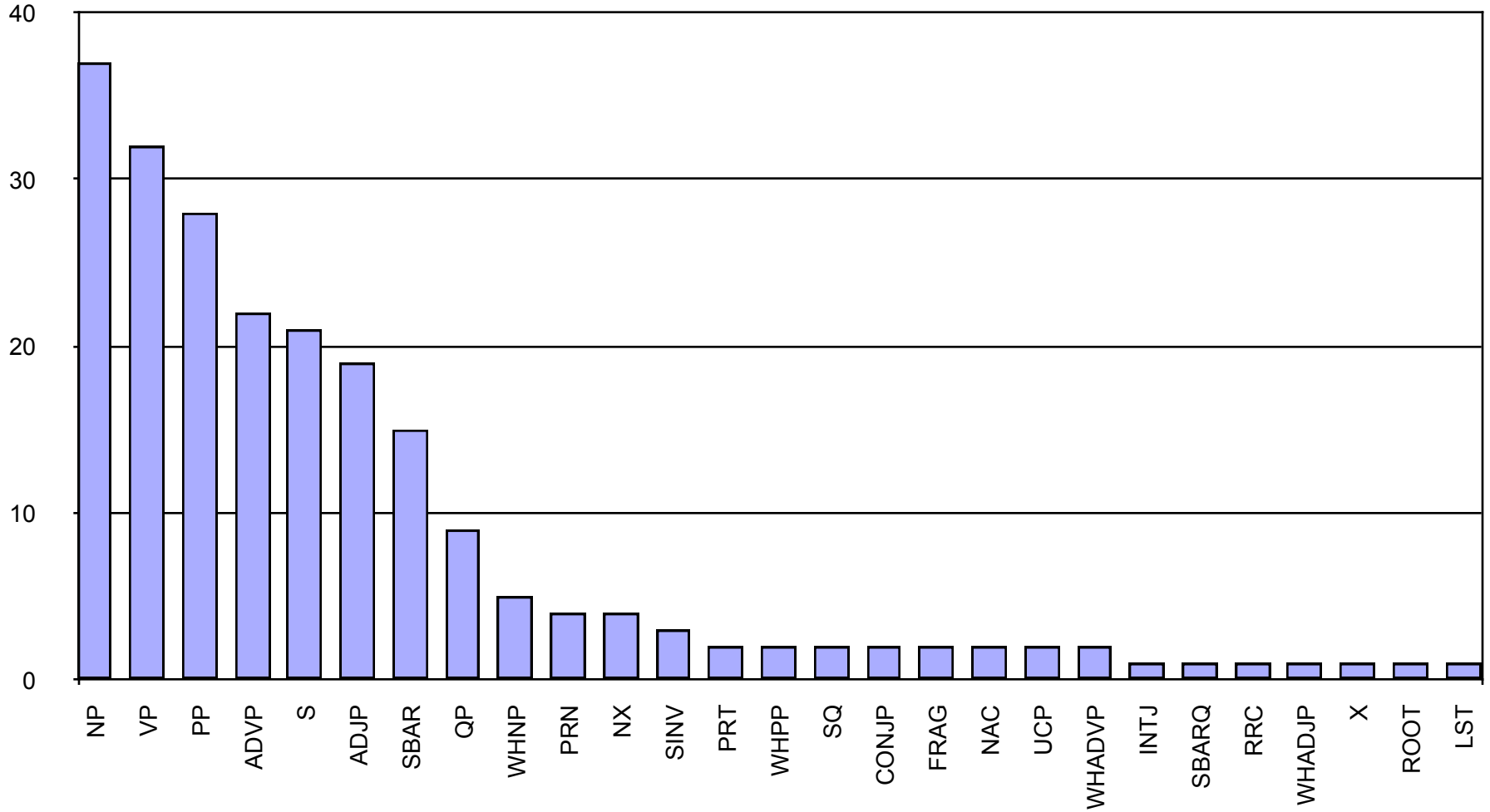


Adaptive Splitting Results



Model	*F1*
Previous	88.4
With 50% Merging	89.5

Number of Phrasal Subcategories



Final Results

Parser	*F1* ≤ *40 words*	*F1* *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
Petrov et. al. 06	**90.2**	**89.7**

Learned Splits

- Proper Nouns (NNP):

NNP-14	Oct.	Nov.	Sept.
NNP-12	John	Robert	James
NNP-2	J.	E.	L.
NNP-1	Bush	Noriega	Peters
NNP-15	New	San	Wall
NNP-3	York	Francisco	Street

- Personal pronouns (PRP):

PRP-0	It	He	I
PRP-1	it	he	they
PRP-2	it	them	him

Learned Splits

- Relative adverbs (RBR):

RBR-0	further	lower	higher
RBR-1	more	less	More
RBR-2	earlier	Earlier	later

- Cardinal Numbers (CD):

CD-7	one	two	Three
CD-4	1989	1990	1988
CD-11	million	billion	trillion
CD-0	1	50	100
CD-3	1	30	31
CD-9	78	58	34