ANTONIS ANASTASOPOULOS
CS499 INTRODUCTION TO NLP

PROBABILISTIC CFGS

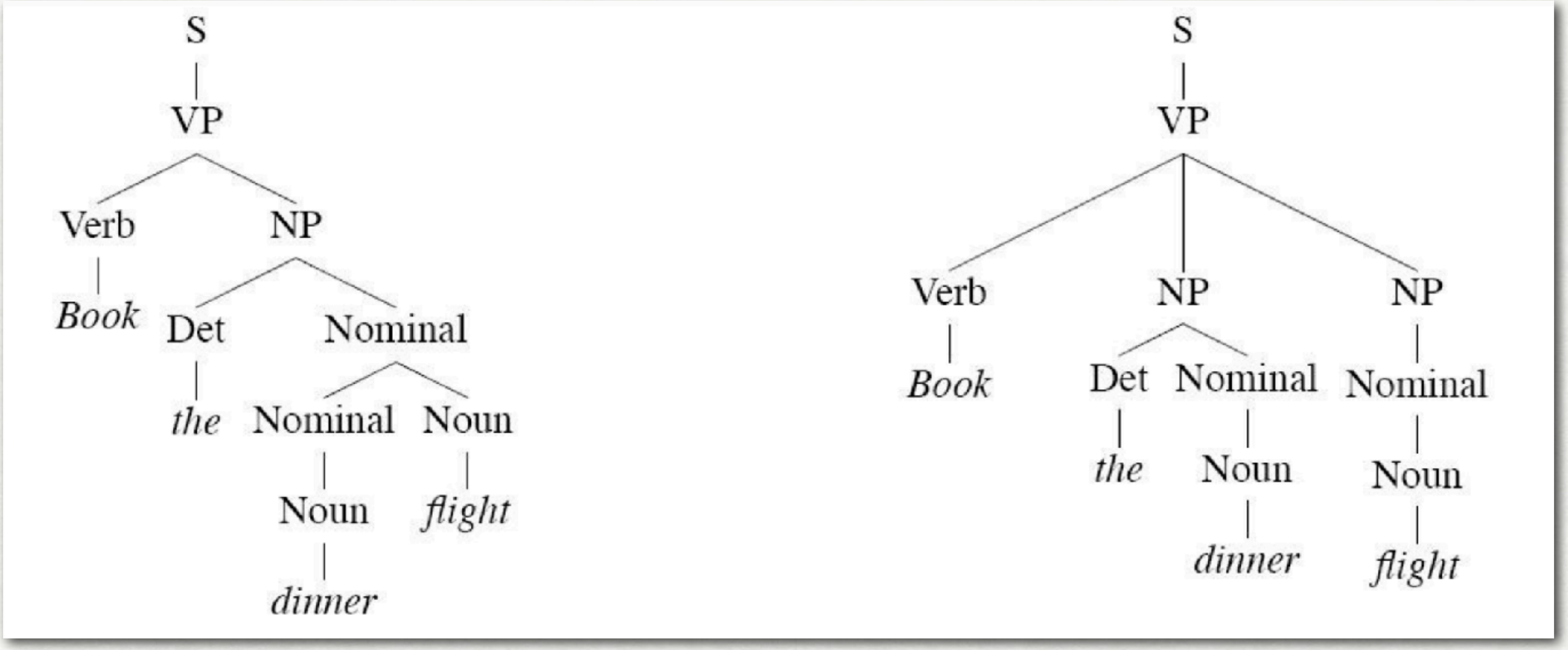
https://cs.gmu.edu/~antonis/course/cs499-spring21/

With adapted slides by David Mortensen and Alan Black

STRUCTURE OF THIS LECTURE

1. Probabilistic Parsing
2. Parsing Algorithms
3. Chomsky Normal Form
4. CKY Algorithm

EXAMPLE AMBIGUOUS PARSE



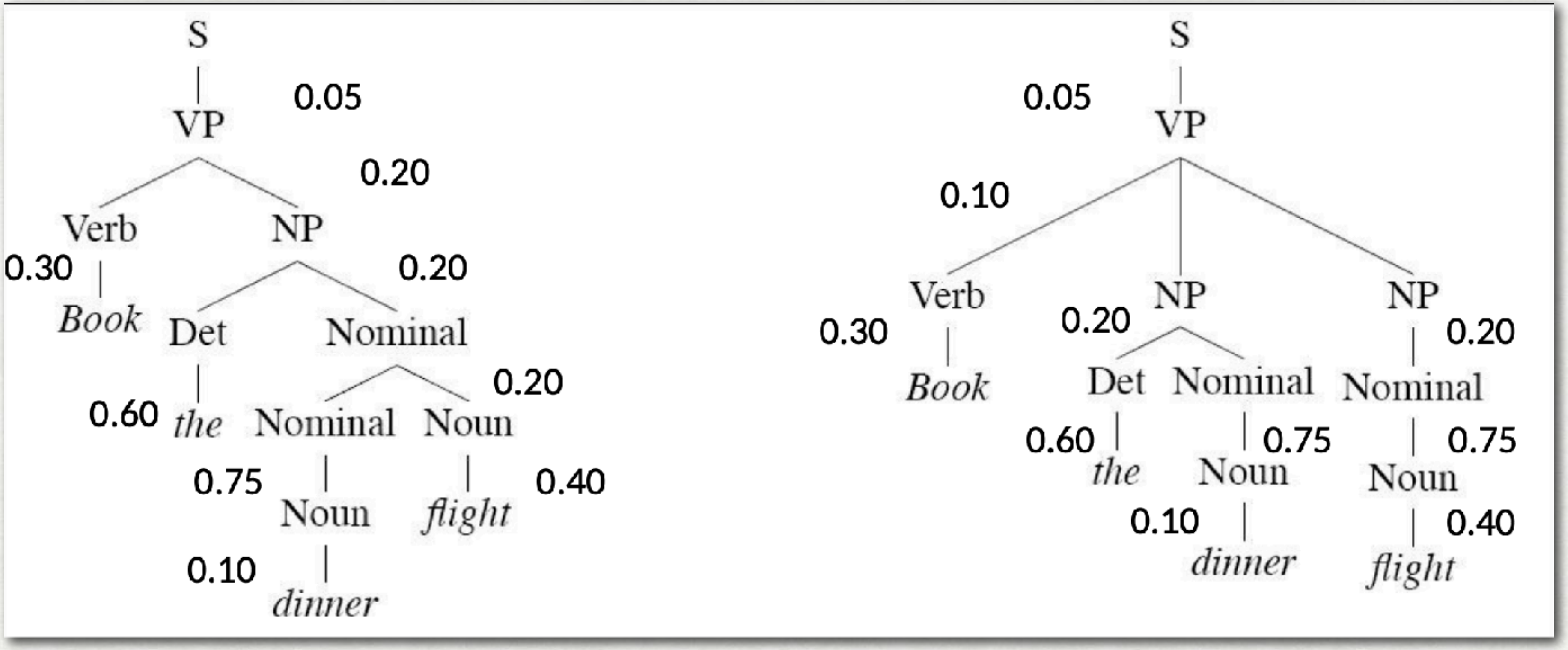
PROBABILISTIC CFG

Grammar	
S → *NP VP*	[.80]
S → *Aux NP VP*	[.15]
S → *VP*	[.05]
NP → *Pronoun*	[.35]
NP → *Proper-Noun*	[.30]
NP → *Det Nominal*	[.20]
NP → *Nominal*	[.15]
Nominal → *Noun*	[.75]
Nominal → *Nominal Noun*	[.20]
Nominal → *Nominal PP*	[.05]
VP → *Verb*	[.35]
VP → *Verb NP*	[.20]
VP → *Verb NP PP*	[.10]
VP → *Verb PP*	[.15]
VP → *Verb NP NP*	[.05]
VP → *VP PP*	[.15]
PP → *Preposition NP*	[1.0]

Lexicon

Det → *that* [.10] | *a* [.30] | *the* [.60]
Noun → *book* [.10] | *flight* [.30]
| *meal* [.15] | *money* [.05]
| *flights* [.40] | *dinner* [.10]
Verb → *book* [.30] | *include* [.30]
| *prefer*; [.40]
Pronoun → *I* [.40] | *she* [.05]
| *me* [.15] | *you* [.40]
Proper-Noun → *Houston* [.60]
| *NWA* [.40]
Aux → *does* [.60] | *can* [40]
Preposition → *from* [.30] | *to* [.30]
| *on* [.20] | *near* [.15]
| *through* [.05]

AMBIGUOUS PARSE WITH PROBABILITIES



$p(\text{left}) = 2.2 \times 10^{-6}$

$p(\text{right}) = 6.1 \times 10^{-7}$

THE PROBABILITY OF A PARSE TREE

The joint probability of a particular parse T and a sentence S, is defined as the product of the probabilities of all the rules r used to expand each node n in the parse tree:

$$P(T, S) = \Pi_{n \in T} \ p(r(n))$$

REVIEW: CONTEXT-FREE GRAMMARS

Vocabulary of terminal symbols: Σ

Set of non-terminal symbols (aka variables): N

Special start symbols: $S \in N$

Production rules of the form $X \to \alpha$, where

$X \in N$

$\alpha \in (N \cup \Sigma)^*$ (in CNF: $\alpha \in N^2 \cup \Sigma$)

PROBABILISTIC CONTEXT-FREE GRAMMARS

Vocabulary of terminal symbols: Σ

Set of non-terminal symbols (aka variables): N

Special start symbols: $S \in N$

Production rules of the form $X \to \alpha$, each with a politic weight $p(X \to \alpha)$, where

$X \in N$

$\alpha \in (N \cup \Sigma)^*$ (in CNF: $\alpha \in N^2 \cup \Sigma$)

$\forall X \in N,\ \Sigma_\alpha\ p(X \to \alpha) = 1$

WHERE TO THE PCFG PROBABILITIES COME FROM?

a) From a **tree bank**

$$P(\alpha \to \beta \mid \alpha) = \frac{\textbf{Count}(\alpha \to \beta)}{\Sigma_\gamma \textbf{Count}(\alpha \to \gamma)} = \frac{\textbf{Count}(\alpha \to \beta)}{\textbf{Count}(\alpha)}$$

b) From a **corpus**

- Parse the corpus with your CFG
- Count the rules for each parse
- Normalize
- ***But wait, most sentences are ambiguous!***
  - *"Keep a separate count for each parse of a sentence and weight each partial count by the probability of the parse it appears in".*

CKY ALGORITHM: REVIEW

For $i = [1 \ \ \ldots \ \ n]$

$C[i-1,\ i] = \{V \mid V \to w_i\}$

For $l = 2 \ \ \ldots \ \ n$: // width

For $i = 0 \ \ \ldots \ \ n - l$: // left boundary

$k = i + l$ // right boundary

For $j = i + 1 \ \ \ldots \ \ k - 1$: // midpoint

$C[i,k] = C[i,k] \cup \{V \mid V \to YZ,\ Y \in C[i,j],\ Z \in C[j,k]\}$

Return true if $S \in C[0,n]$

WEIGHTED CKY ALGORITHM

For $i = [1 \;\ldots\; n]$

$C[V, i-1, i] = p(V \to w_i)$

For $l = 2 \;\ldots\; n$: // width of span

For $i = 0 \;\ldots\; n - l$: // left boundary

$k = i + l$ // right boundary

For $j = i + 1 \;\ldots\; k - 1$: // midpoint

For each binary rule $V \to Y\, Z$:

$$C[V, i, k] = \max\{C[V, i, k],\; C[Y, i, j] \times C[Z, j, k] \times p(V \to YZ)\}$$

Return true if $S \in C[\,\cdot\,, 0, n]$

CKY EQUATIONS: REVIEW

$$C[i-1, i, w_i] = \text{TRUE}$$

$$C[i-1, i, V] = \begin{cases} \text{TRUE} & \text{if } V \to w_i \\ \text{FALSE} & \text{otherwise} \end{cases}$$

$$C[i, j, V] = \begin{cases} \text{TRUE} & \text{if } \exists j, Y, Z \text{ such that} \\ & V \to Y Z \\ & \quad \text{and } C[i, k, Y] \\ & \quad \text{and } C[k, j, Z] \\ & \quad \text{and } i < k < j \\ \text{FALSE} & \text{otherwise} \end{cases}$$

$$\text{goal} = C[0, n, S]$$

WEIGHTED CKY EQUATIONS

$$
\begin{aligned}
\textbf{base case:} & \\
C[X, i-1, i] &= p(X \rightarrow w_i) \\
\textbf{induction:} & \\
C[X, i, k] &= \max_{j, Y, Z} p(X \rightarrow Y \ Z) \times C[Y, i, j] \times C(Z, j, k) \\
\textbf{goal:} & \\
C[S, 0, n] & \text{ where } n = |\boldsymbol{w}| \\
p(\tau^*, w_1, w_2, \ldots, w_n) &= C[S, 0, n]
\end{aligned}
$$

P-CKY ALGORITHM FROM BOOK

```
function PROBABILISTIC-CKY(words,grammar) returns most probable parse
                                                   and its probability
  for j ← from 1 to LENGTH(words) do
     for all { A | A → words[j] ∈ grammar}
        table[j − 1, j,A] ← P(A → words[j])
     for i ← from j − 2 downto 0 do
        for k ← i + 1 to j − 1 do
              for all { A | A → BC ∈ grammar,
                              and table[i,k,B] > 0 and table[k, j,C] > 0 }
                    if (table[i,j,A] < P(A → BC) × table[i,k,B] × table[k,j,C]) then
                       table[i,j,A] ← P(A → BC) × table[i,k,B] × table[k,j,C]
                       back[i,j,A] ← {k,B,C}
     return BUILD_TREE(back[1, LENGTH(words), S]), table[1, LENGTH(words), S]
```

CKY: CHART

For $i = [1 \ \dots \ n]$

$C[V, i-1, i] = p(V \rightarrow w_i)$

For $l = 2 \ \dots \ n$: // width of span

For $i = 0 \ \dots \ n - l$: // left boundary

$k = i + l$ // right boundary

For $j = i + 1 \ \dots \ k - 1$: // midpoint

For each binary rule $V \rightarrow Y\ Z$:

$C[V, i, k] = \max\{C[V, i, k],\ C[Y, i, j] \times C[Z, j, k] \times p(V \rightarrow YZ)\}$

Return true if $S \in C[\cdot, 0, n]$

	[0,1]	[0,2]	[0,3]	[0,4]	[0,5]
The		[1,2]	[1,3]	[1,4]	[1,5]
	Cat		[2,3]	[2,4]	[2,5]
		Sat		[3,4]	[3,5]
			…		[4,5]
				…	

CKY: CHART

For $i = [1 \ \dots \ n]$

$C[V, i-1, i] = p(V \to w_i)$

For $l = 2 \ \dots \ n$: // width of span

For $i = 0 \ \dots \ n - l$: // left boundary

$k = i + l$ // right boundary

For $j = i + 1 \ \dots \ k - 1$: // midpoint

For each binary rule $V \to Y\ Z$:

$C[V, i, k] = \max\{C[V, i, k],\ C[Y, i, j] \times C[Z, j, k] \times p(V \to YZ)\}$

Return true if $S \in C[\ \cdot\ , 0, n]$

	Det: 0.4 [0,1]	[0,2]	[0,3]	[0,4]	[0,5]
The		N: 0.02 [1,2]	[1,3]	[1,4]	[1,5]
	Cat		V: 0.05 [2,3]	[2,4]	[2,5]
		Sat		[3,4]	[3,5]
			…		[4,5]
				…	

CKY: CHART

	Det: 0.4 [0,1]	NP: .3*.4*.02 = 0.0024 [0,2]	[0,3]	[0,4]	[0,5]
The		N: 0.02 [1,2]	[1,3]	[1,4]	[1,5]
	Cat		V: 0.05 [2,3]	[2,4]	[2,5]
		Sat		[3,4]	[3,5]
			…		[4,5]
				…	

For $i = [1 \ \dots \ n]$

$C[V, i-1, i] = p(V \to w_i)$

For $l = 2 \ \dots \ n$: // width of span

For $i = 0 \ \dots \ n - l$: // left boundary

$k = i + l$ // right boundary

For $j = i + 1 \ \dots \ k - 1$: // midpoint

For each binary rule $V \to Y\ Z$:

$C[V, i, k] = \max\{C[V, i, k],\ C[Y, i, j] \times C[Z, j, k] \times p(V \to YZ)\}$

Return true if $S \in C[\cdot, 0, n]$

TREEBANKS

THE PENN TREEBANK (PTB)

The first big treebank, still widely used

Consists of the Brown Corpus, ATIS (Air Travel Information Service corpus), Switchboard corpus, and a corpus drawn from the *Wall Street Journal*

Produced at University of Pennsylvania (thus the name)

About 1 million words

About 17,500 distinct rule types

- PTB rules tend to be “flat” —lots of symbols on the RHS
- Many of the rule types only occur in one tree

TREEBANK TREE EXAMPLE

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

40717 PP → IN NP
33803 S → NP-SBJ VP
22513 NP-SBJ → -NONE-
21877 NP → NP PP
20740 NP → DT NN
14153 S → NP-SBJ VP .
12922 VP → TO VP
11881 PP-LOC → IN NP
11467 NP-SBJ → PRP
11378 NP → -NONE-
11291 NP → NN
...
989 VP → VBG S
985 NP-SBJ → NN
983 PP-MNR → IN NP
983 NP-SBJ → DT
969 VP → VBN VP
...

100 VP → VBD PP-PRD
100 PRN → : NP :
100 NP → DT JJS
100 NP-CLR → NN
99 NP-SBJ-1 → DT NNP
98 VP → VBN NP PP-DIR
98 VP → VBD PP-TMP
98 PP-TMP → VBG NP
97 VP → VBD ADVP-TMP VP
...
10 WHNP-1 → WRB JJ
10 VP → VP CC VP PP-TMP
10 VP → VP CC VP ADVP-MNR
10 VP → VBZ S , SBAR-ADV
10 VP → VBZ S ADVP-TMP

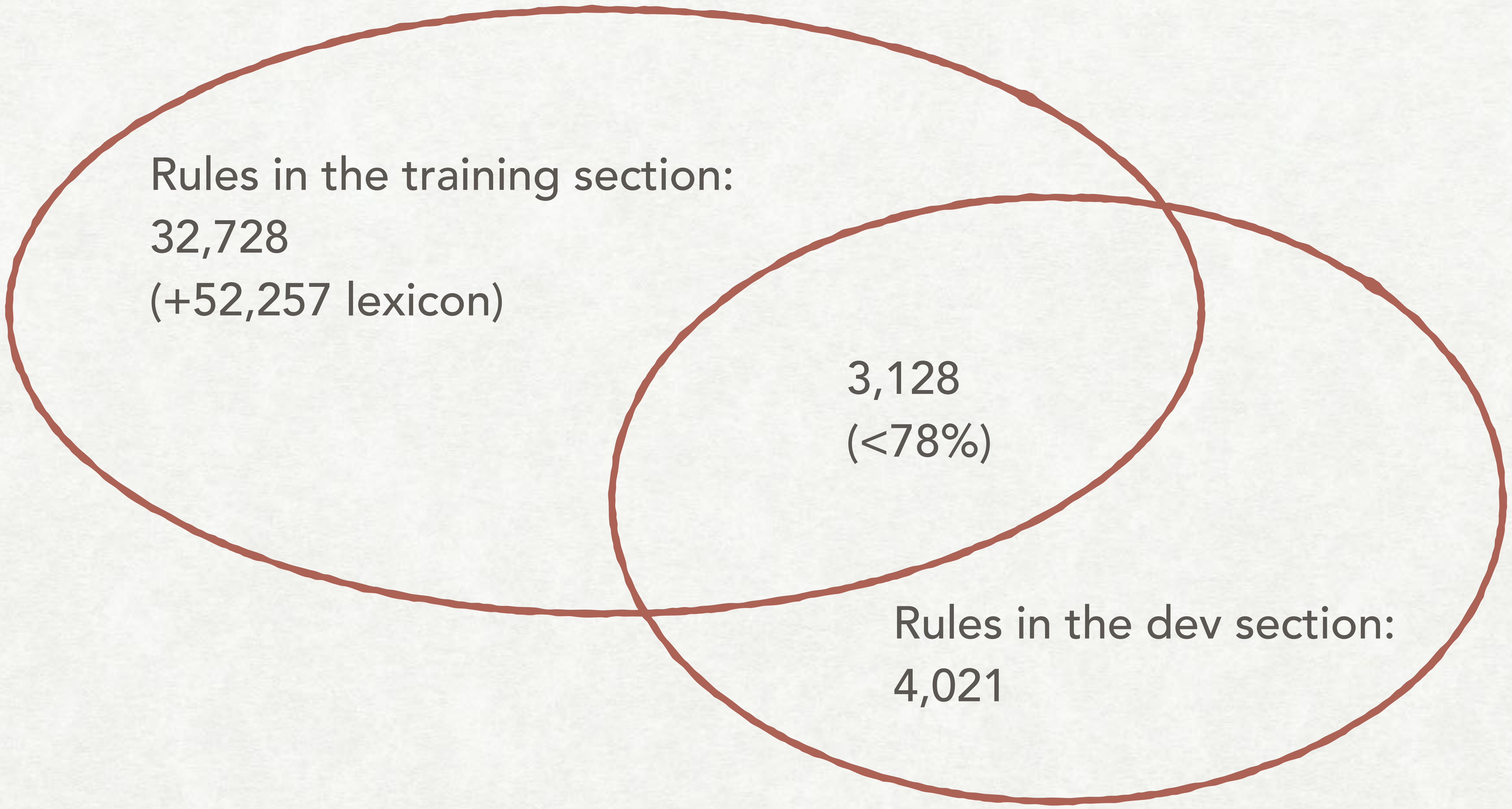
TREEBANK TREE EXAMPLE

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

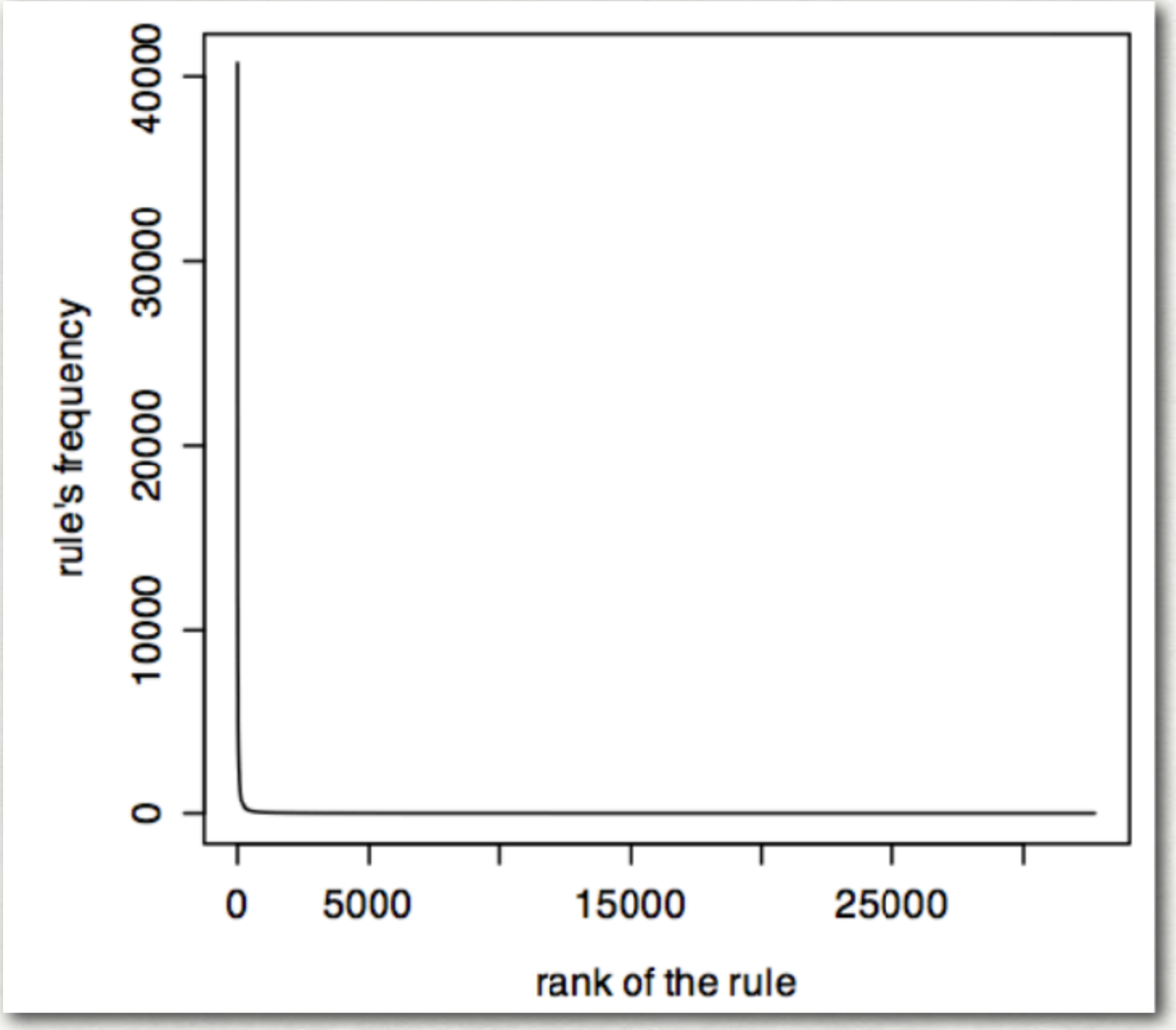
40717 PP → IN NP
33803 S → NP-SBJ VP
22513 NP-SBJ → -NONE-
21877 NP → NP PP
20740 NP → DT NN
14153 S → NP-SBJ VP .
12922 VP → TO VP
11881 PP-LOC → IN NP
11467 NP-SBJ → PRP
11378 NP → -NONE-
11291 NP → NN
...
989 VP → VBG S
985 NP-SBJ → NN
983 PP-MNR → IN NP
983 NP-SBJ → DT
969 VP → VBN VP
...

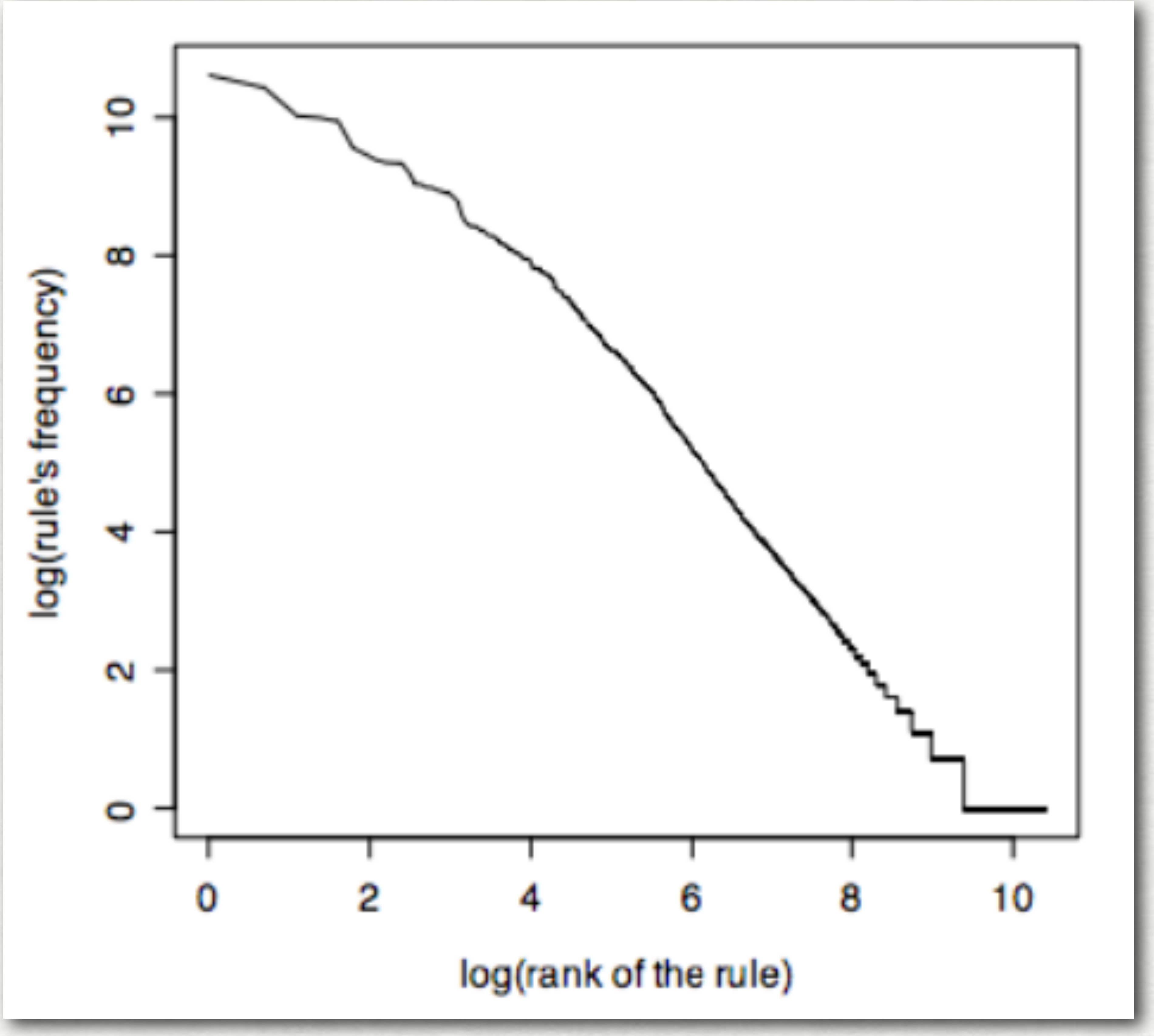
100 VP → VBD PP-PRD
100 PRN → : NP :
100 NP → DT JJS
100 NP-CLR → NN
99 NP-SBJ-1 → DT NNP
98 VP → VBN NP PP-DIR
98 VP → VBD PP-TMP
98 PP-TMP → VBG NP
97 VP → VBD ADVP-TMP VP
...
10 WHNP-1 → WRB JJ
10 VP → VP CC VP PP-TMP
10 VP → VP CC VP ADVP-MNR
10 VP → VBZ S , SBAR-ADV
10 VP → VBZ S ADVP-TMP

RULES IN THE TREEBANK



RULE DISTRIBUTION (TRAINING SET)





OTHER TREEBANKS

PTB is just one, very important, treebank

There are many others, though they are often (a) smaller, (b) dependency treebanks.

However, there are plenty of constituency/phrase structure tree banks in addition to PTB.

UNIVERSAL DEPENDENCIES

Universal dependencies (UD)

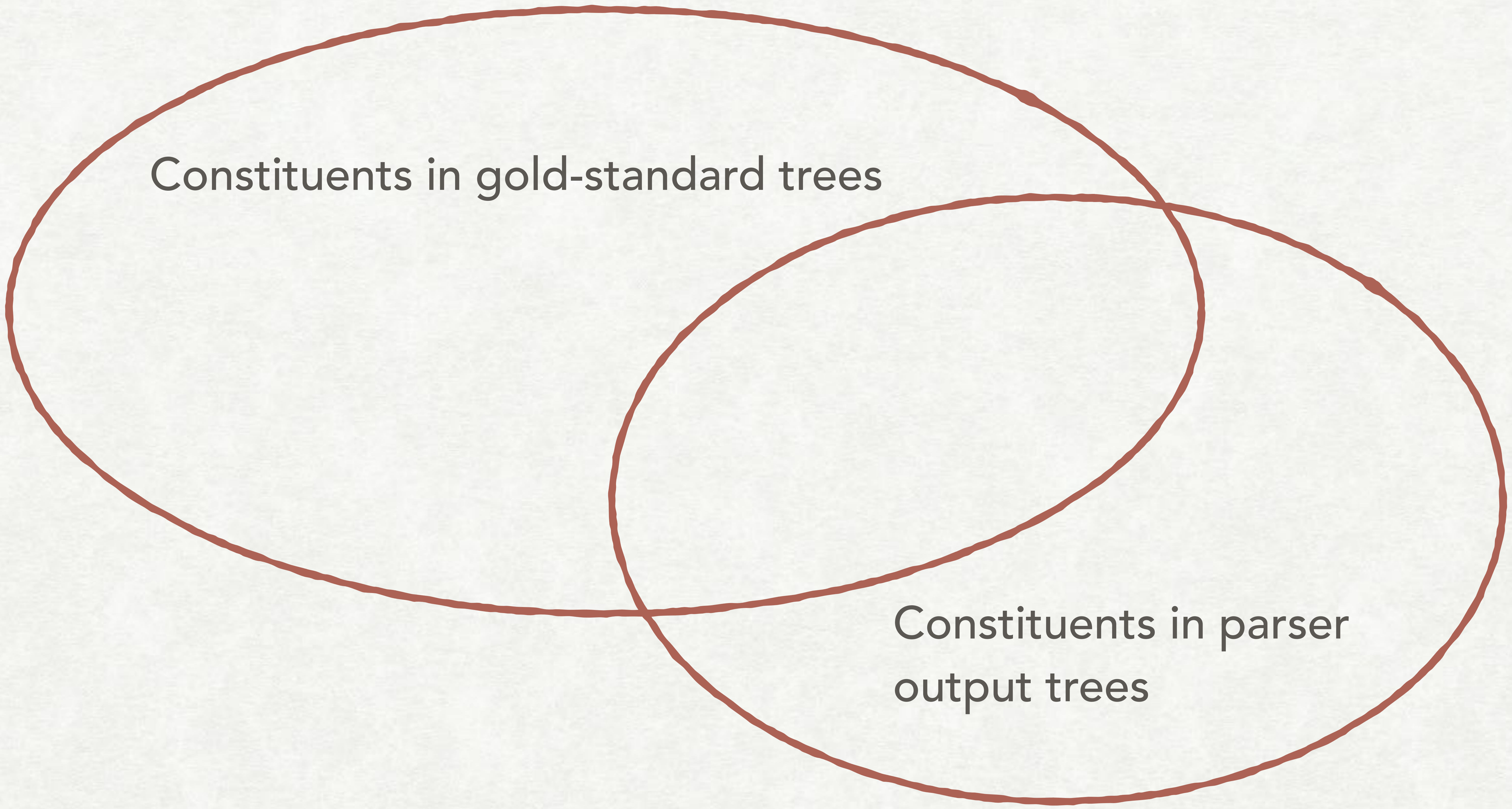
- internally consistent set of universal dependency relations
- used to construct a large body of treebanks in **many languages**
- useful for cross-lingual training (since the PoS and the dependency labels are the same cross-linguistically)

Not immediately applicable to what we talked about, since it's relatively hard to learn constituency information from dependency trees

Very relevant to training dependency parsers

PARSING EVALUATION

PARSEVAL



Constituents in gold-standard trees
Constituents in parser output trees

PARSEVAL

$$\textbf{labeled recall:} = \frac{\text{\# of correct constituents in candidate parse of } s}{\text{\# of correct constituents in treebank parse of } s}$$

$$\textbf{labeled precision:} = \frac{\text{\# of correct constituents in candidate parse of } s}{\text{\# of total constituents in candidate parse of } s}$$

cross-brackets: the number of crossed brackets (e.g. the number of constituents for which the treebank has a bracketing such as ((A B) C) but the candidate parse has a bracketing such as (A (B C))).

THE F-MEASURE

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

$$F_1 = \frac{2PR}{P + R}$$

NEXT CLASS

Neural Models for Dependency Parsing