CZ4045 Natural Language Processing

Statistical parsing (Chapter 14)

Takeaways

- Probabilistic CFG
  - Treebank
  - Probabilistic CKY
  - Problems
    - Attachment ambiguities
    - Structural dependencies between rules
  - Solutions
    - Splitting non-terminals
    - Lexicalized PCFG
- Evaluating Parsing Accuracy
  - Sentence-level accuracy
  - Constituent-level accuracy

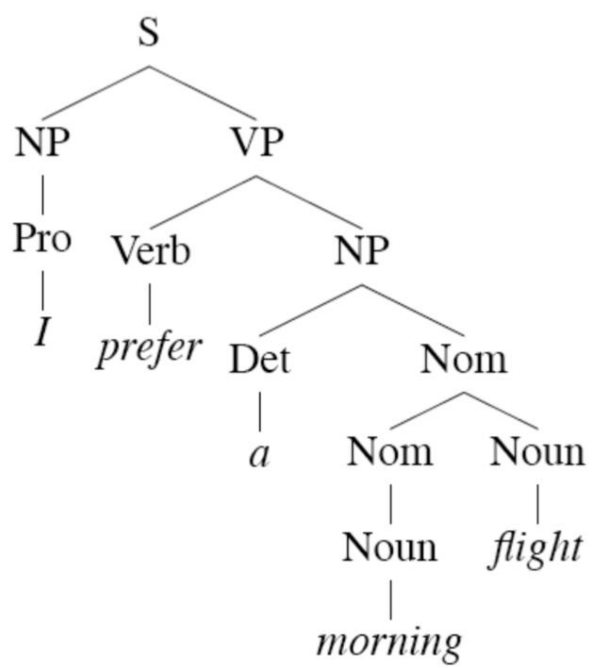
Review

Syntactic Parsing at Hand

- The task of taking a string and a CFG grammar and returning phrase structure(s)

I prefer a morning flight.

$$
\begin{array}{rcl}
S & \rightarrow & NP\ VP \\
\\
NP & \rightarrow & Pronoun \\
 & | & Proper\text{-}Noun \\
 & | & Det\ Nominal \\
Nominal & \rightarrow & Nominal\ Noun \\
 & | & Noun \\
\\
VP & \rightarrow & Verb \\
 & | & Verb\ NP \\
 & | & Verb\ NP\ PP \\
 & | & Verb\ PP \\
\\
PP & \rightarrow & Preposition\ NP
\end{array}
$$



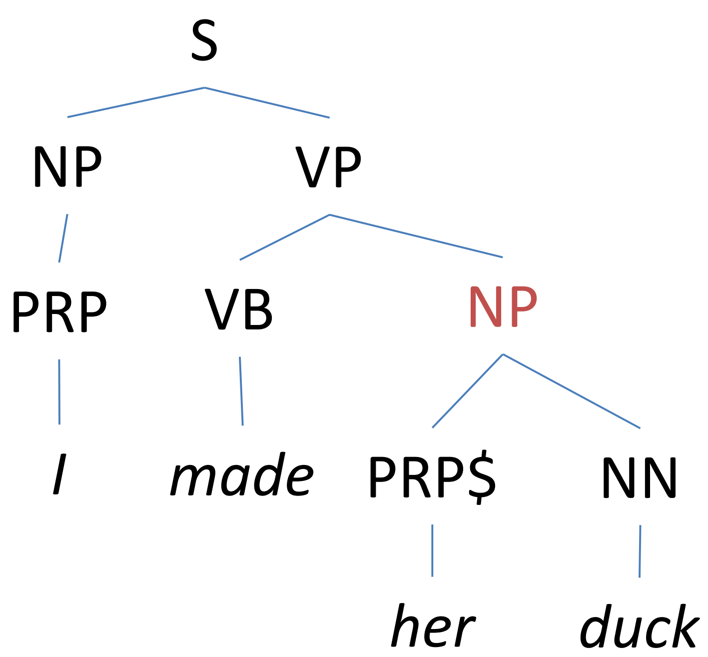
Review

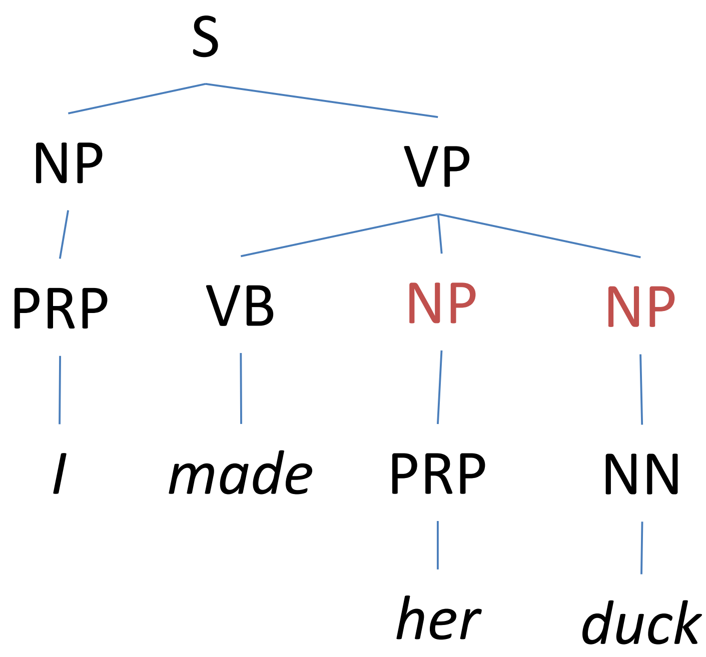
Ambiguity is Pervasive

- Find at least 5 meanings of this sentence: **I made her duck**

- Possible meanings
  - I cooked waterfowl for her
  - I cooked waterfowl belonging to her
  - I created the (plaster?) duck she owns
  - I caused her to quickly lower her head and body
  - I waved my magic wand and turned her into undifferentiated waterfowl

Ambiguity Resolution by Syntactic Structures





I cooked waterfowl belonging to her
I created the duck she owns

I cooked waterfowl for her benefit
I waved my magic wand and turned her into undifferentiated waterfowl

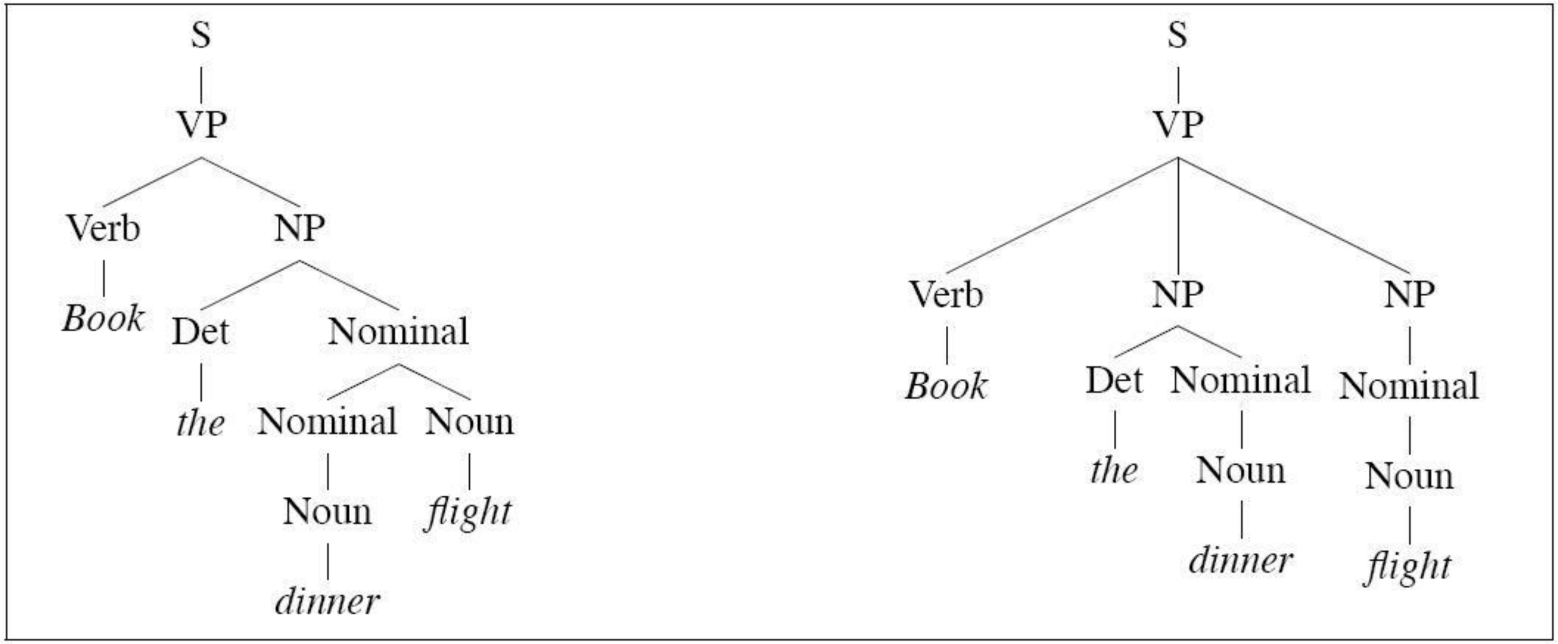
Exercise: Attachment Ambiguity

- How many distinct phrase structures may the following sentence have due to the prepositional phrase attachment ambiguities?

  - **John wrote the book with a pen in the room**

Need for Syntactic Disambiguation

- Consider the two parses of “Book the dinner flight”
- The left parse is sensible, while the right is not
  - How about “Can you book John a flight?”



Need for Common Sense

- People usually provide **only useful information** and take the rest for granted. The rest is common-sense: obvious things people know and usually leave unstated

Open Mind Common Sense

- OMCS is a MIT-based AI project whose goal is to build a common-sense knowledge base from the contributions of many thousands of people across the Web (as common-sense cannot be *easily* mined from text)

- OMCS on Wikipedia

Concepts

wooden, keep your money, a waiting room, a cathedral, people sleep, Asbestos, a good job, nice weather, a curb, fluff

Vote on these statements...

→ Grapefruits are a sort of fruits
→ You can use a rocking chair to relax
→ flirting is for having fun
→ Judaism is a religion
→ Something that might happen when you fish is you catch fish
→ Sometimes applying for a job causes rejection
→ Something you find on a table is computer
→ Kinds of programming languages : c++
→ a number can be used for math
→ mercury is metal

Context-Free Grammar

Review

- G = (T, N, S, R)
  - T: a set of terminals (e.g. ‘flight’)
  - N: a set of non-terminals (e.g. Noun)
  - S: the start symbol, a non-terminal
  - R: rules of the form X →γ
    - X: a non-terminal
    - γ: a sequence of terminals and non-terminals

$$
\boxed{\begin{array}{rcl}
NP & \rightarrow & Det\ Nominal \\
NP & \rightarrow & ProperNoun \\
Nominal & \rightarrow & Noun \mid Nominal\ Noun
\end{array}}
$$

Probabilistic Context-Free Grammar (PCFG)

- G = (T, N, S, R, P)
  - T: a set of terminals (e.g. ‘boy’)
  - N: a set of non-terminals (e.g. Noun)
  - S: the start symbol, a non-terminal
  - R: rules of the form X $\rightarrow \gamma$
  - P(R) gives the probability of each rule

Grammar	
$S \rightarrow NP\ VP$	[.80]
$S \rightarrow Aux\ NP\ VP$	[.15]
$S \rightarrow VP$	[.05]

$$\forall X \in N, \sum_{X \rightarrow \gamma \in R} P(X \rightarrow \gamma) = 1$$

PCFG: Example

Grammar		Lexicon
$S \rightarrow NP\ VP$	[.80]	$Det \rightarrow that$ [.10] | a [.30] | the [.60]
$S \rightarrow Aux\ NP\ VP$	[.15]	$Noun \rightarrow book$ [.10] | $flight$ [.30]
$S \rightarrow VP$	[.05]	| $meal$ [.15] | $money$ [.05]
$NP \rightarrow Pronoun$	[.35]	| $flights$ [.40] | $dinner$ [.10]
$NP \rightarrow Proper\text{-}Noun$	[.30]	$Verb \rightarrow book$ [.30] | $include$ [.30]
$NP \rightarrow Det\ Nominal$	[.20]	| $prefer;$ [.40]
$NP \rightarrow Nominal$	[.15]	$Pronoun \rightarrow I$ [.40] | she [.05]
$Nominal \rightarrow Noun$	[.75]	| me [.15] | you [.40]
$Nominal \rightarrow Nominal\ Noun$	[.20]	$Proper\text{-}Noun \rightarrow Houston$ [.60]
$Nominal \rightarrow Nominal\ PP$	[.05]	| NWA [.40]
$VP \rightarrow Verb$	[.35]	$Aux \rightarrow does$ [.60] | can [40]
$VP \rightarrow Verb\ NP$	[.20]	$Preposition \rightarrow from$ [.30] | to [.30]
$VP \rightarrow Verb\ NP\ PP$	[.10]	| on [.20] | $near$ [.15]
$VP \rightarrow Verb\ PP$	[.15]	| $through$ [.05]
$VP \rightarrow Verb\ NP\ NP$	[.05]	
$VP \rightarrow VP\ PP$	[.15]	
$PP \rightarrow Preposition\ NP$	[1.0]	

PCFG

- How to learn the probability of rules?

$$\forall X \in N, \sum_{X \to \gamma \in R} P(X \to \gamma) = 1$$

- How to estimate the probability of parse trees with a PCFG?

- Once we have probabilities of possible parse trees, we can select the parse tree with the highest probability as the parse result for a given string

Probability of Rules

- Need for treebanks! $P(X \rightarrow \beta \mid X)$

$$P(X \rightarrow \beta \mid X) = \frac{\text{count}(X \rightarrow \beta)}{\sum_{\gamma} \text{count}(X \rightarrow \gamma)} = \frac{\text{count}(X \rightarrow \beta)}{\text{count}(X)}$$

$$\forall X \in N, \sum_{X \rightarrow \gamma \in R} P(X \rightarrow \gamma) = 1$$

Treebank (Tree-Annotated Corpus)

- Penn TreeBank (PTB) is a widely used treebank
  - Most well known is the Wall Street Journal section of the Penn TreeBank
  - 1 M words from the 1987-1989 Wall Street Journal

```
( (S (`` ``)
    (S-TPC-2
      (NP-SBJ-1 (PRP We) )
      (VP (MD would)
        VP (VB have)
          (S
            (NP-SBJ (-NONE- *-1) )
            (VP (TO to)
              (VP (VB wait)
                (SBAR-TMP (IN until)
                  (S
                    (NP-SBJ (PRP we) )
                    (VP (VBP have)
                      (VP (VBN collected)
                        (PP-CLR (IN on)
                          (NP (DT those)(NNS assets)))))))))))))
          ('' '')
      BJ (PRP he) )
    (VP (VBD said)
      (S (-NONE- *T*-2) ))
    (. .) ))
```

Increase count(VP → Verb PP) by 1

Mapping Grammars

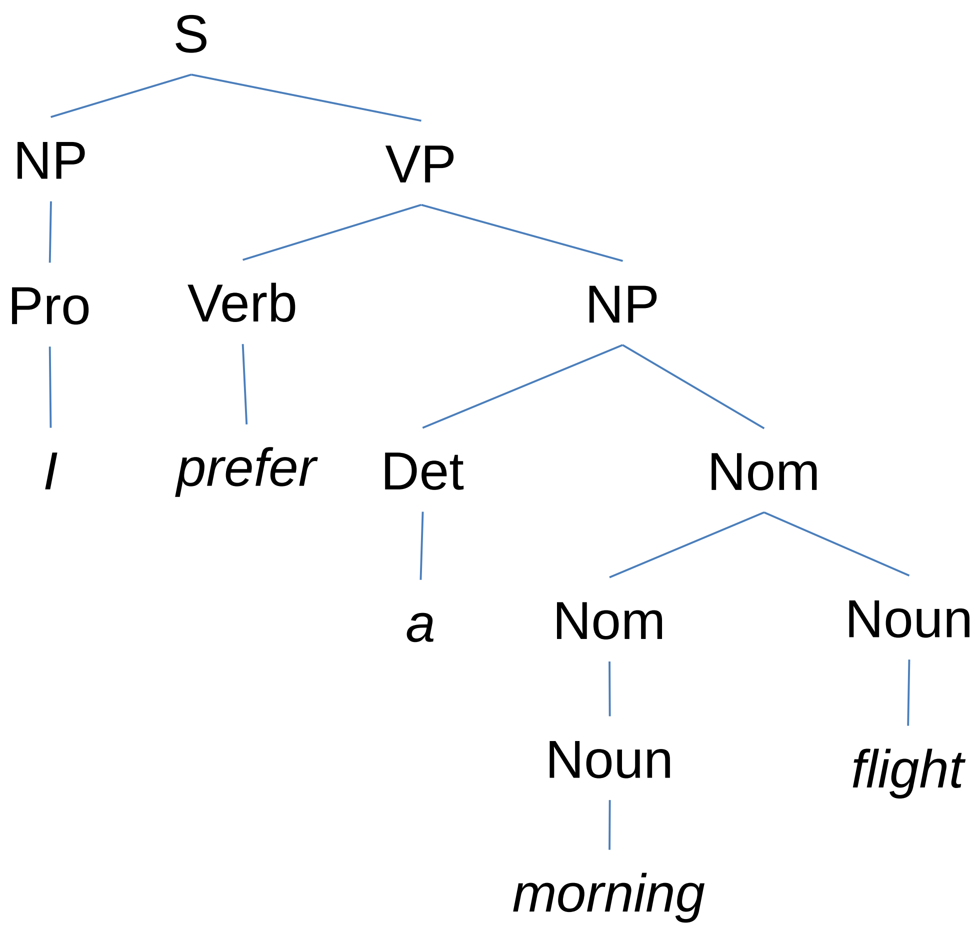
- Problem
  - Your grammar may be different from the grammar of a treebank
    - e.g. Verb vs. VBN
    - e.g. PP vs. PP-CLR

- Solution steps
  - Transform the treebank according to your grammar
  - **Learn the probabilistic model for the grammar based on the transformed treebank**
  - **Parse sentences with the probabilistic model**
  - Transform the parse results back into the treebank grammar for evaluation

Review

Derivation

- A derivation (parse tree) consists of the bag of grammar rules that are in the tree



1. S → NP VP
2. NP → Pro
   Pro → I
3. VP → Verb NP
   Verb → prefer
4. NP → Det Nom
   Det → a
5. Nom → Nom Noun
   Noun → morning
6. Nom → Noun
   Noun → flight

Probability of Parse Trees

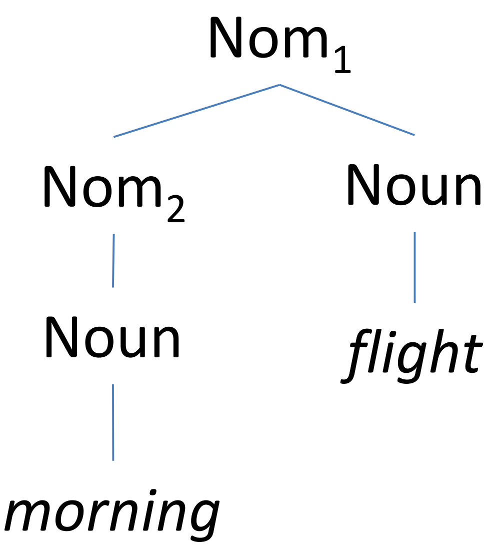
- A derivation (parse tree) consists of the bag of grammar rules that are in the tree
  - The probability of a tree is the product of the probabilities of the rules in the derivation.

1. S → NP VP
2. NP → Pro
   Pro → I
3. VP → Verb NP
   Verb → prefer
4. NP → Det Nom
   Det → a
5. Nom → Nom Noun
   Noun → morning
6. Nom → Noun
   Noun → flight

$$P(T,S) = \prod_{node \in T} P(rule(n))$$

Probability of Parse Tree: Example

- Noun $\rightarrow$ morning [0.10]
- Noun $\rightarrow$ flight [0.40]
- Nom $\rightarrow$ Noun [0.75]
- Nom $\rightarrow$ Nom Noun [0.20]

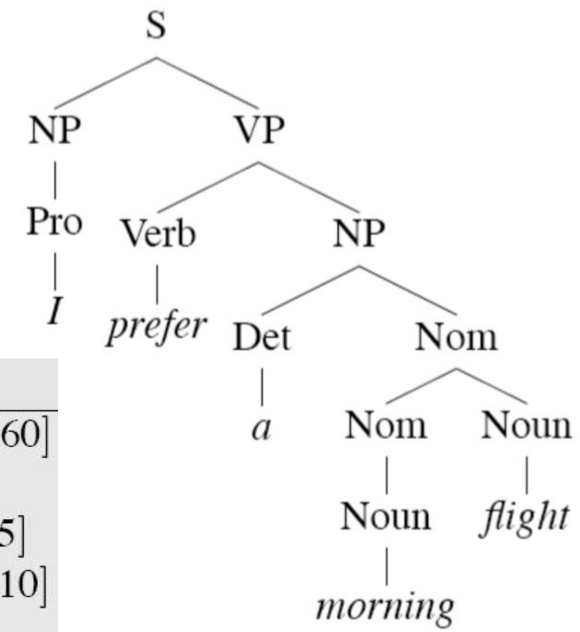


$$P(Nom_2) = 0.75 \times 0.1 = 0.75 \times 10^{-1}$$

$$P(Nom_1) = 0.2 \times 0.75 \times 10^{-1} \times 0.4 = 0.6 \times 10^{-2}$$

Exercise: Probability of Parse Tree

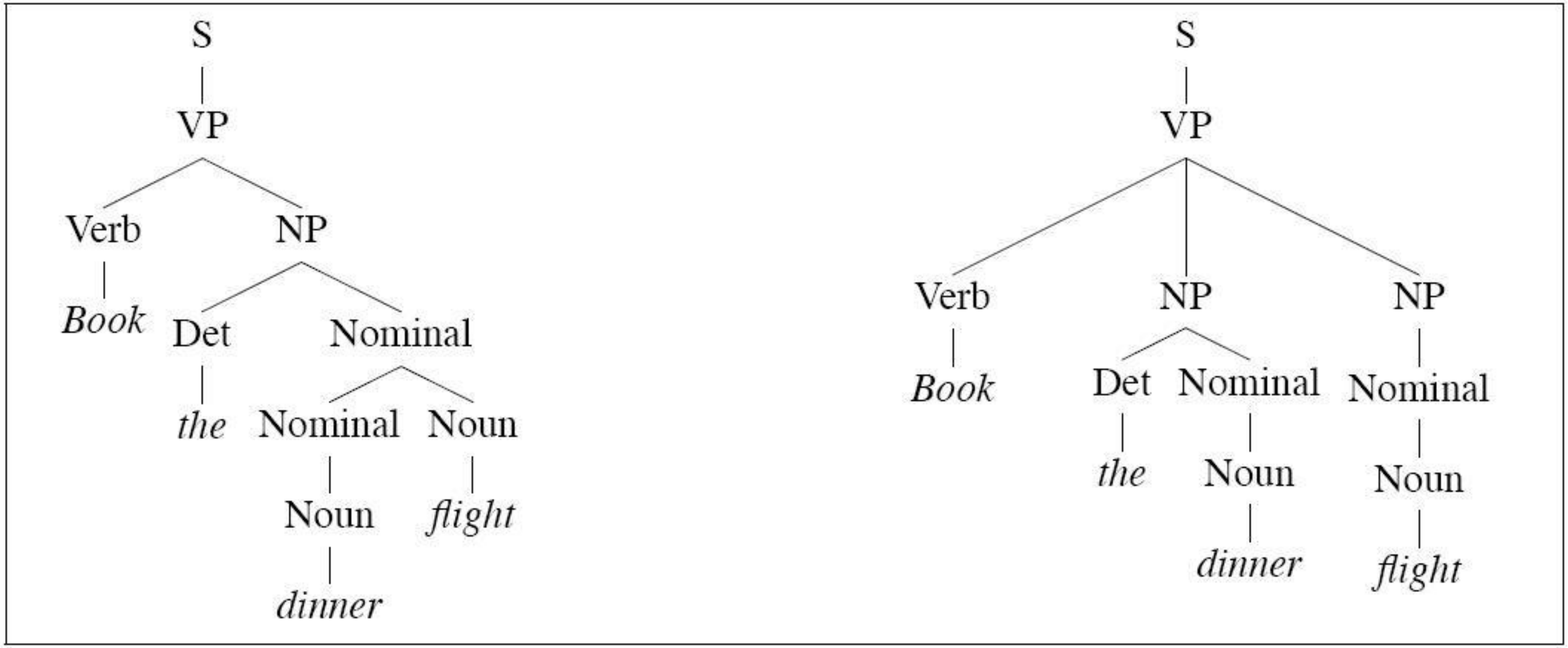
- Calculate the probability of the parse tree below



Grammar		Lexicon
S → NP VP	[.80]	Det → that [.10] $\mid$ a [.30] $\mid$ the [.60]
S → Aux NP VP	[.15]	Noun → book [.10] $\mid$ flight [.30]
S → VP	[.05]	$\mid$ meal [.15] morning [.05]
NP → Pronoun	[.35]	$\mid$ flights [.40] $\mid$ dinner [.10]
NP → Proper-Noun	[.30]	Verb → book [.30] $\mid$ include [.30]
NP → Det Nominal	[.20]	$\mid$ prefer; [.40]
NP → Nominal	[.15]	Pronoun → I [.40] $\mid$ she [.05]
Nominal → Noun	[.75]	$\mid$ me [.15] $\mid$ you [.40]
Nominal → Nominal Noun	[.20]	Proper-Noun → Houston [.60]
Nominal → Nominal PP	[.05]	$\mid$ NWA [.40]
VP → Verb	[.35]	Aux → does [.60] $\mid$ can [40]
VP → Verb NP	[.20]	Preposition → from [.30] $\mid$ to [.30]
VP → Verb NP PP	[.10]	$\mid$ on [.20] $\mid$ near [.15]
VP → Verb PP	[.15]	$\mid$ through [.05]
VP → Verb NP NP	[.05]	
VP → VP PP	[.15]	
PP → Preposition NP	[1.0]	

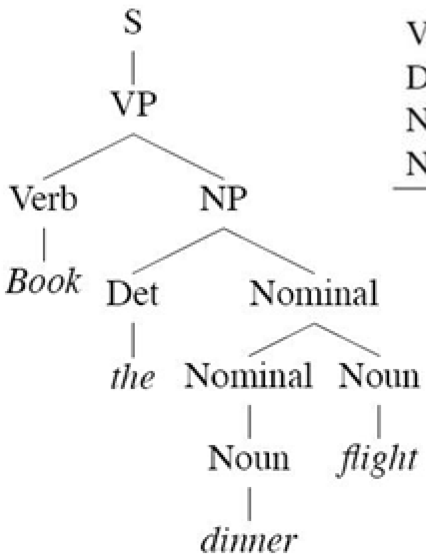
Why Do We Need Probability of Parse Trees?

- Once we have probabilities of possible parse trees, we can select the parse tree with the highest probability as the parse result for a given string

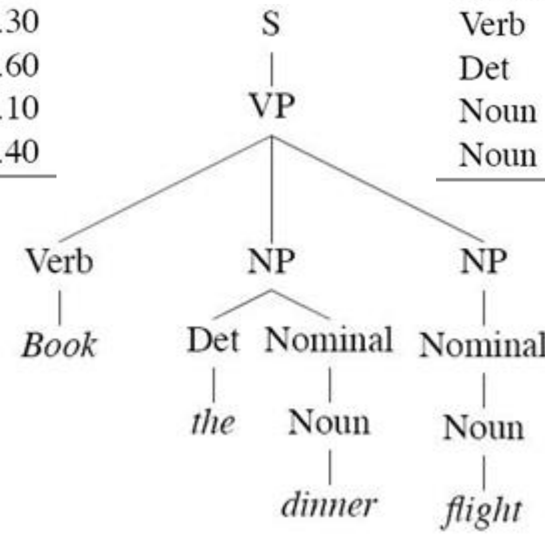


Why Do We Need Probability of Parse Trees?

Rules			P
S	→	VP	.05
VP	→	Verb NP	.20
NP	→	Det Nominal	.20
Nominal	→	Nominal Noun	.20
Nominal	→	Noun	.75
Verb	→	book	.30
Det	→	the	.60
Noun	→	dinner	.10
Noun	→	flights	.40



Rules			P
S	→	VP	.05
VP	→	Verb NP NP	.10
NP	→	Det Nominal	.20
NP	→	Nominal	.15
Nominal	→	Noun	.75
Nominal	→	Noun	.75
Verb	→	book	.30
Det	→	the	.60
Noun	→	dinner	.10
Noun	→	flights	.40



$$P(T_{left}) = .05 * .20 * .20 * .20 * .75 * .30 * .60 * .10 * .40 = \mathbf{2.2 \times 10^{-6}}$$
$$P(T_{right}) = .05 * .10 * .20 * .15 * .75 * .75 * .30 * .60 * .10 * .40 = \mathbf{6.1 \times 10^{-7}}$$

But How Accurate/General Are These Probabilities?

- Probabilities are bound to specific datasets or corpora and, in general, are not domain-independent

Probabilistic CKY

- Probability of a new constituent *A* derived from the rule $A \rightarrow B\ C$:
  - $P(A \rightarrow B\ C \mid A) * P(B) * P(C)$

  - Where P(B) and P(C) are already in the table given the way that CKY operates
  - What we store is the MAX probability over all the A rules for a given cell in the table

Probabilistic CKY: Example

Det: .40 [0,1]	NP: .30 *.40 *.02 = .0024 [0,2]	[0,3]	[0,4]	[0,5]
	N: .02 [1,2]	[1,3]	[1,4]	[1,5]
		V: .05 [2,3]	[2,4]	[3,5]
			[3,4]	[3,5]
				[4,5]
The	flight	includes	a	meal

- $P(NP \rightarrow Det\ N \mid NP) = 0.3$
  - The probability of Cell [0, 2] to be noun phrase with [0,1] being a determiner and [1, 2] being a noun.

Exercise: Probabilistic CKY

- What constituents (with what probability) can you make for substring “runs down”?
  - PP → IN NP 0.002
  - NP → NNS NNS 0.010
  - NP → NNS NP 0.005
  - NP → NNS RB 0.001
  - VP → VB RB 0.045
  - VP → VB NP 0.015

??

NNS 0.0023
VB 0.0010

RB 0.2000
IN 0.0014
NNS 0.0001

runs

down

Exercise: Probabilistic CKY

- Probability of a new constituent *A* derived from the rule $A \rightarrow B\ C$:
  $P(A \rightarrow B\ C \mid A) * P(B) * P(C)$

  - PP → IN NP 0.002
  - NP → NNS NNS 0.010
  - NP → NNS NP 0.005
  - NP → NNS RB 0.001
  - VP → VB RB 0.045
  - VP → VB NP 0.015

  - **NP → NNS NNS = 0.23 x 10^-8**
  - **NP → NNS RB = 0.46 x 10^-6**
  - **VP → VB RB = 0.9 x 10^-5**

??	
NNS 0.0023 VB 0.0010	RB 0.2000 IN 0.0014 NNS 0.0001
runs	down

Probabilistic CKY Algorithm

function PROBABILISTIC-CKY(*words*, *grammar*) **returns** most probable parse and its probability

for $j \leftarrow$ **from** 1 **to** LENGTH(*words*) **do**
  for all $\{ A \mid A \rightarrow words[j] \in grammar \}$
    $table[j-1, j, A] \leftarrow P(A \rightarrow words[j])$
  for $i \leftarrow$ **from** $j-2$ **downto** 0 **do**
    for $k \leftarrow i+1$ **to** $j-1$ **do**
      for all $\{ A \mid A \rightarrow BC \in grammar,$ **and** $table[i,k,B] > 0$ **and** $table[k,j,C] > 0 \}$
        if $(table[i,j,A] < P(A \rightarrow BC) \times table[i,k,B] \times table[k,j,C])$ **then**
          $table[i,j,A] \leftarrow P(A \rightarrow BC) \times table[i,k,B] \times table[k,j,C]$
          $back[i,j,A] \leftarrow \{k, B, C\}$
  return BUILD_TREE(*back*[1, LENGTH(*words*), *S*]), *table*[1, LENGTH(*words*), *S*]

Problems with PCFG

- Doesn't take the actual words (Grammar) into account
  - e.g., verb subcategorization
- Doesn't take into account where in the derivation a rule is used
  - e.g., NPs that are syntactic objects are more likely to be Pronouns.

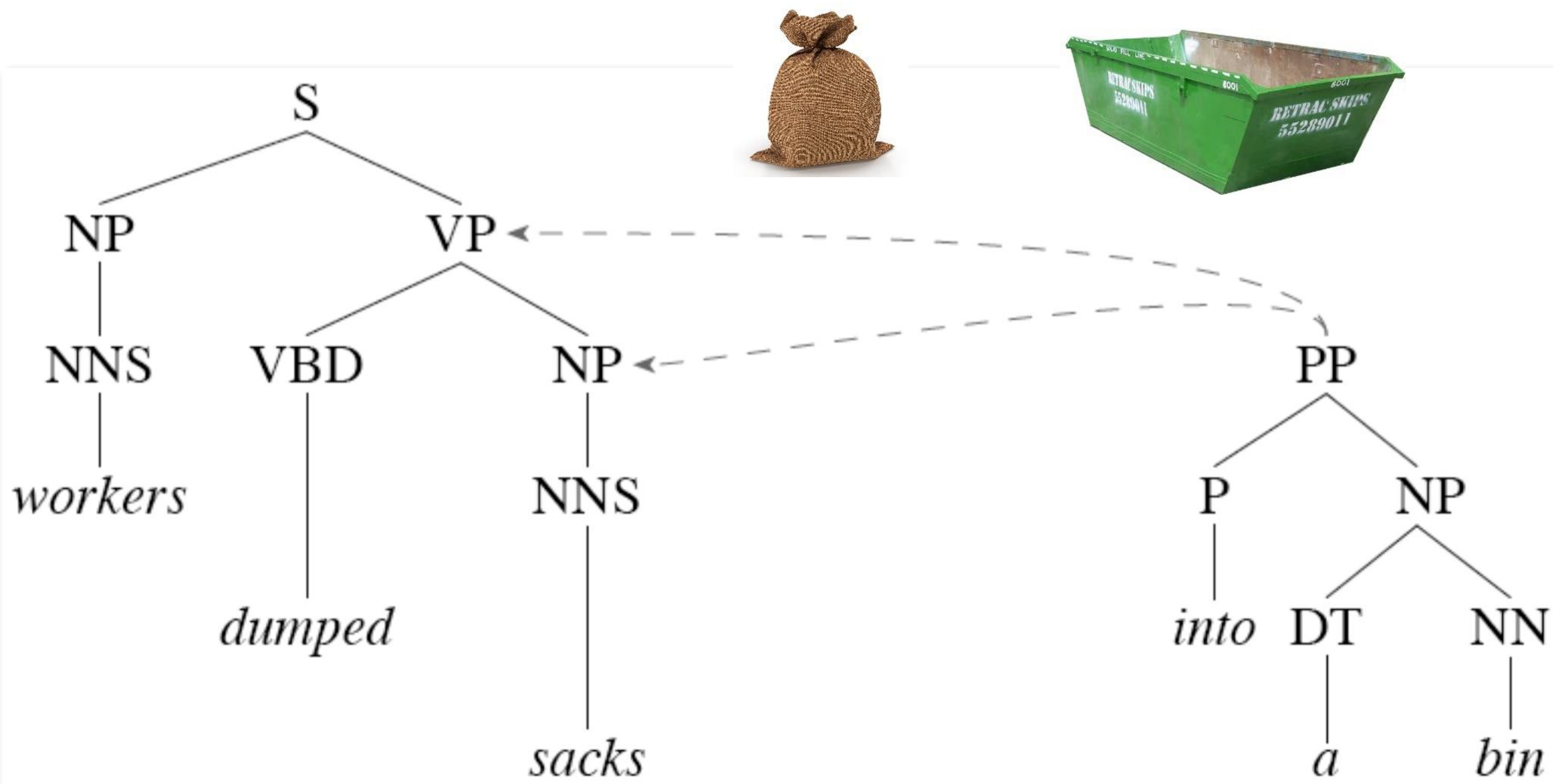
Grammar		Lexicon
S → *NP VP*	[.80]	*Det* → *that* [.10] $\mid$ *a* [.30] $\mid$ *the* [.60]
S → *Aux NP VP*	[.15]	*Noun* → *book* [.10] $\mid$ *flight* [.30]
S → *VP*	[.05]	$\mid$ *meal* [.15] $\mid$ *money* [.05]
NP → *Pronoun*	[.35]	$\mid$ *flights* [.40] $\mid$ *dinner* [.10]
NP → *Proper-Noun*	[.30]	*Verb* → *book* [.30] $\mid$ *include* [.30]
NP → *Det Nominal*	[.20]	$\mid$ *prefer*; [.40]
NP → *Nominal*	[.15]	*Pronoun* → *I* [.40] $\mid$ *she* [.05]
Nominal → *Noun*	[.75]	$\mid$ *me* [.15] $\mid$ *you* [.40]
Nominal → *Nominal Noun*	[.20]	*Proper-Noun* → *Houston* [.60]
Nominal → *Nominal PP*	[.05]	$\mid$ *NWA* [.40]
VP → *Verb*	[.35]	*Aux* → *does* [.60] $\mid$ *can* [40]
VP → *Verb NP*	[.20]	*Preposition* → *from* [.30] $\mid$ *to* [.30]
VP → *Verb NP PP*	[.10]	$\mid$ *on* [.20] $\mid$ *near* [.15]
VP → *Verb PP*	[.15]	$\mid$ *through* [.05]
VP → *Verb NP NP*	[.05]	
VP → *VP PP*	[.15]	
PP → *Preposition NP*	[1.0]	

Specific Problems

- Attachment ambiguities
  - Prepositional phrase (PP) attachment
  - Coordination problem
- Structural dependencies between rules

PP Attachment

- Example sentence: Workers dumped sacks into a bin.



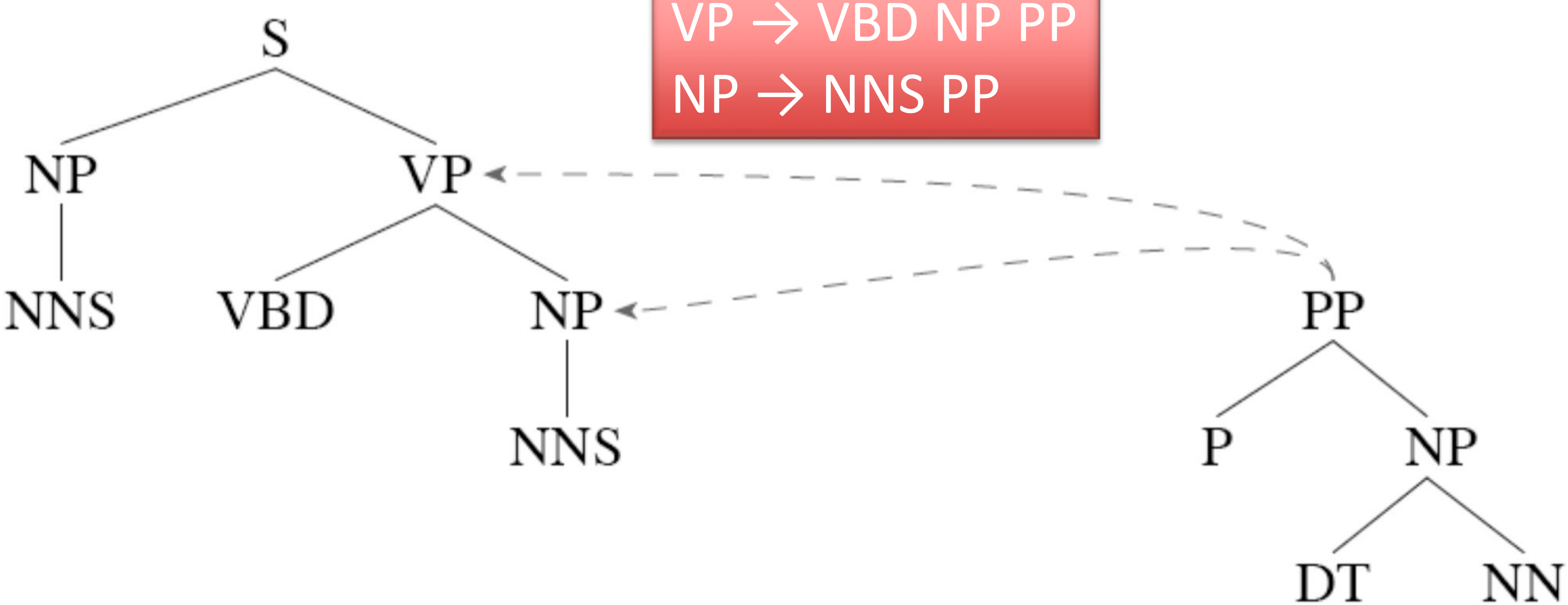
‘Dump’ has a stronger association with ‘into’

PP Attachment

- NNS VBD NNS P DT NN
  - Rule does not consider the actural words in the sentence.
  - So we are not using the actual words here, but only the rules

VP → VBD NP PP

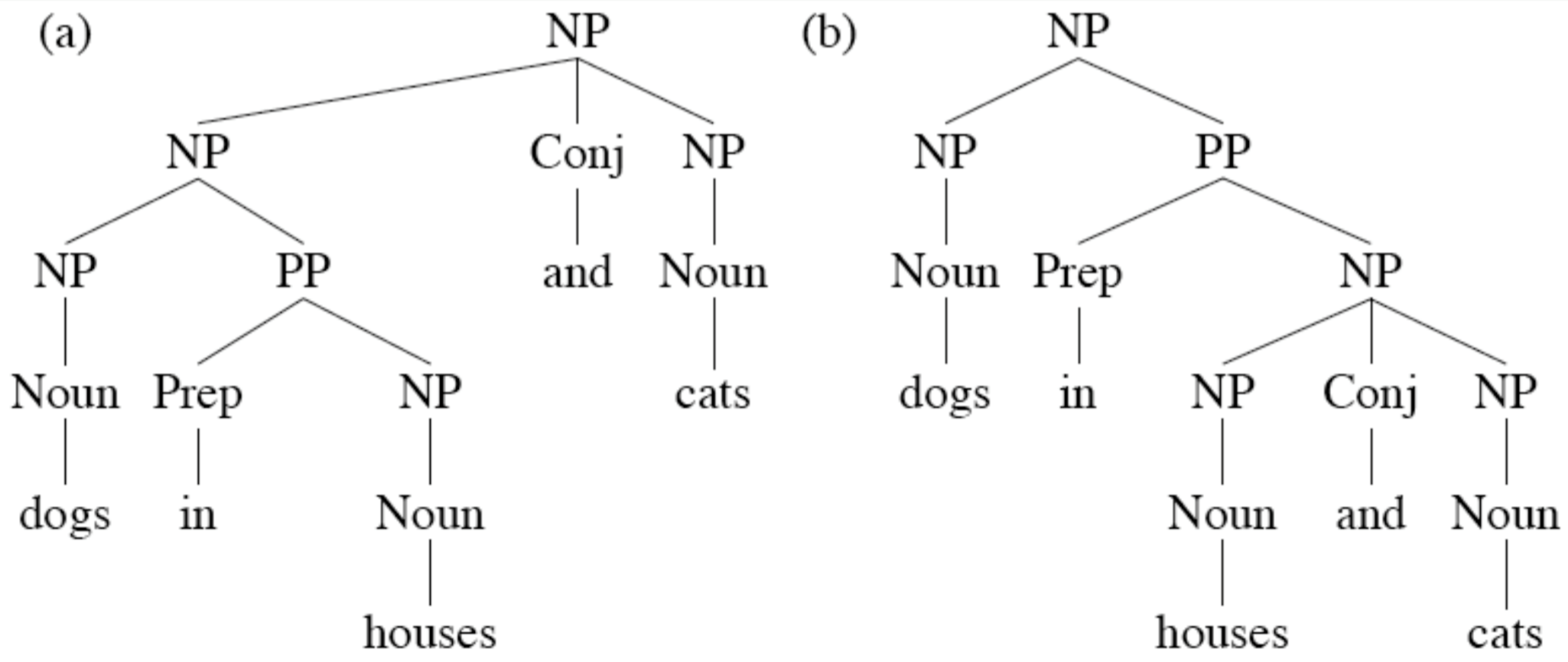
NP → NNS PP



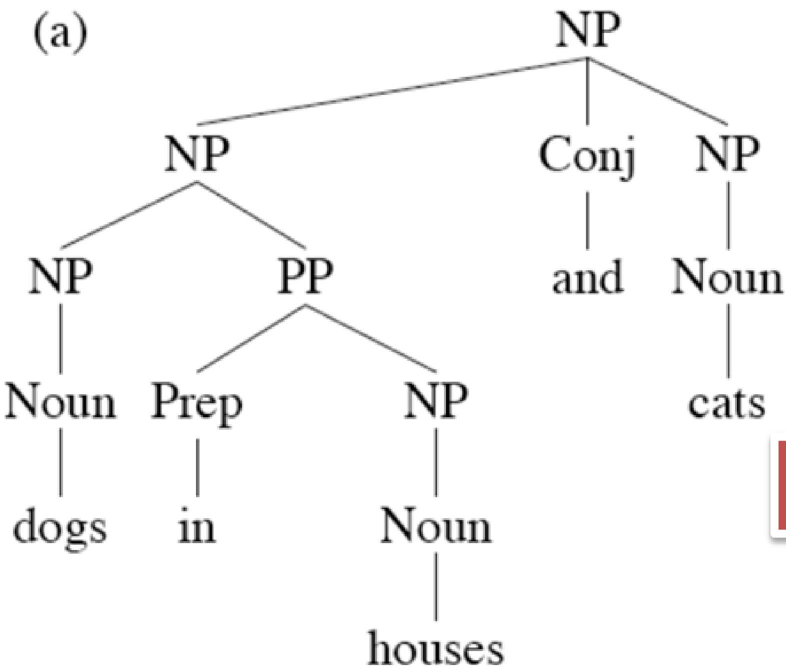
Both rules are valid, and we cannot determine the attachment here.

Coordination Problem

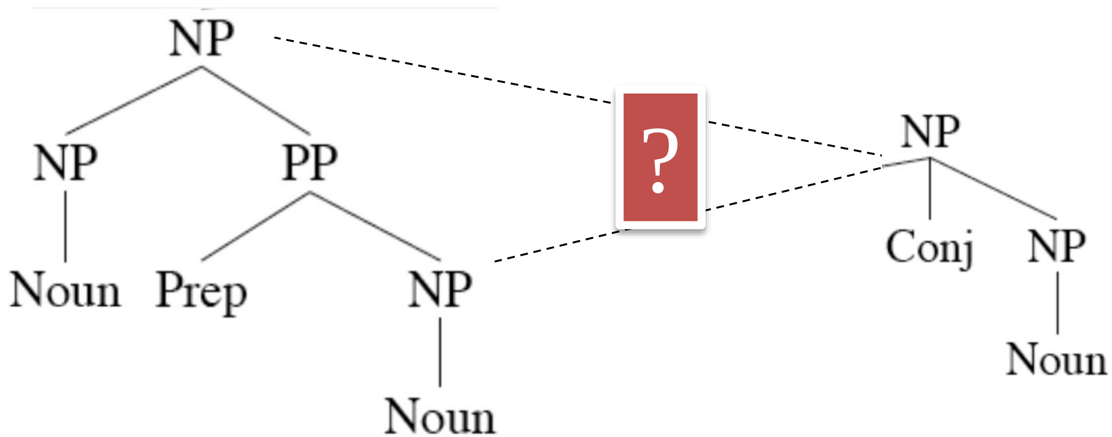
- Most grammars have such (implicit) rules as “X → X and X”
  - This leads to massive ambiguity problems.



Coordination Problem



Again the rules do not consider the words



Structural Dependencies Between Rules

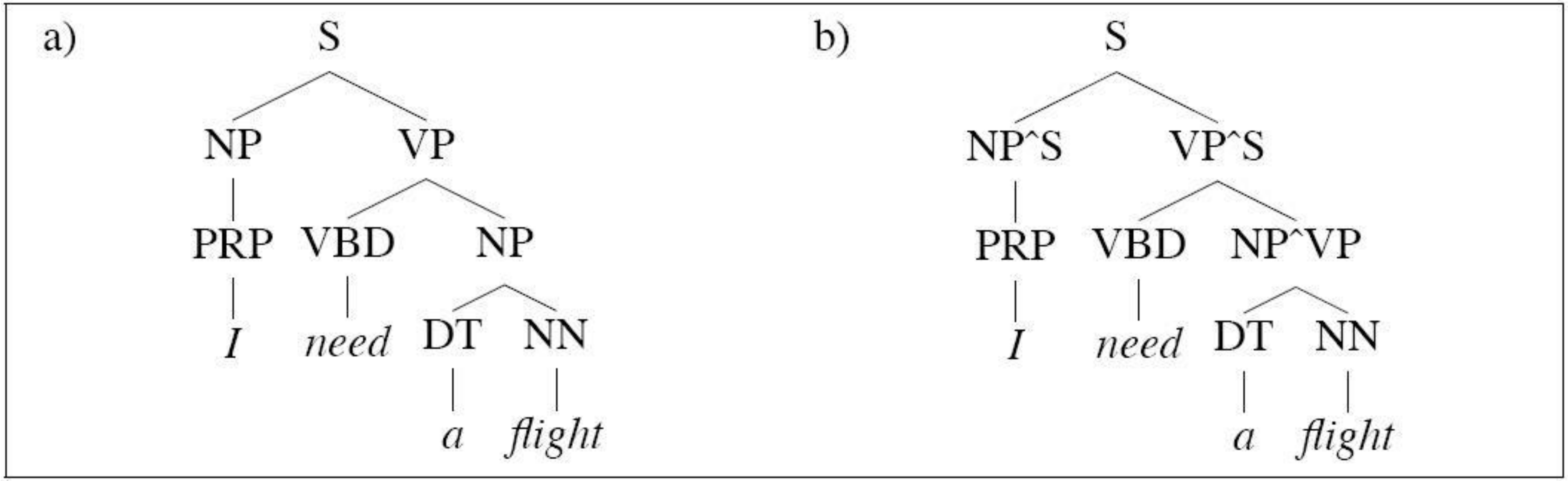
- Example probability for rules of NP
  - NP → DT NN (0.28)
  - NP → PRP (0.25)

- Rules that involve NP
  - S → **NP** VP
  - VP → Verb **NP**

	Pronoun	Non-Pronoun
Subject	91%	9%
Object	34%	66%

Improving PCFG: Splitting Non-Terminals

- Encoding contextual dependencies into PCFG symbols
  - NP is a child of S → NP^S
  - NP is a child of VP → NP^VP



Improving PCFG: Splitting Non-Terminals

Grammar
$S \rightarrow NP\ VP$
$S \rightarrow Aux\ NP\ VP$
$S \rightarrow VP$
$NP \rightarrow Pronoun$
$NP \rightarrow Proper\text{-}Noun$
$NP \rightarrow Det\ Nominal$
$NP \rightarrow Nominal$
$Nominal \rightarrow Noun$
$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow Verb$
$VP \rightarrow Verb\ NP$
$VP \rightarrow Verb\ NP\ PP$
$VP \rightarrow Verb\ PP$
$VP \rightarrow Verb\ NP\ NP$
$VP \rightarrow VP\ PP$
$PP \rightarrow Preposition\ NP$

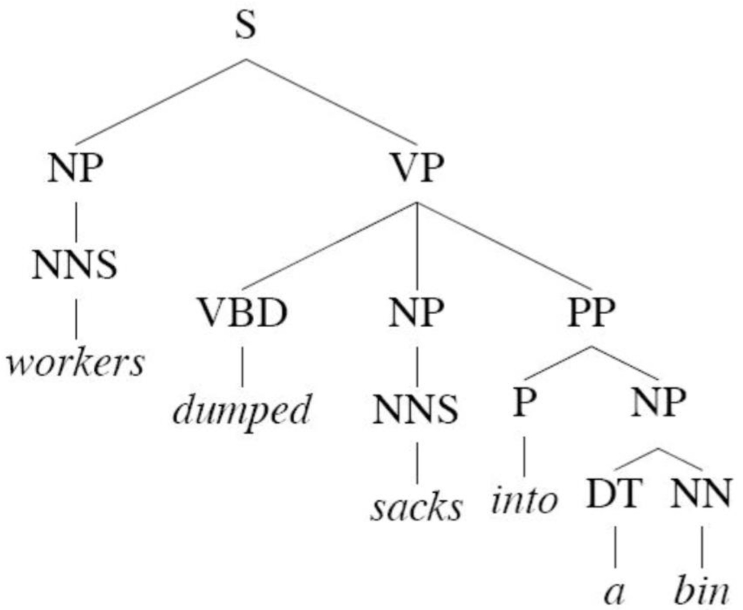
NP^S → Pronoun
NP^VP → Pronoun
NP^PP → Pronoun

NP^S → Det Nominal^NP
NP^VP → Det Nominal^NP
NP^PP → Det Nominal^NP

	Pronoun	Non-Pronoun
Subject	91%	9%
Object	34%	66%

Improving PCFG: Lexicalized PCFG

- How to add lexical information to rules?



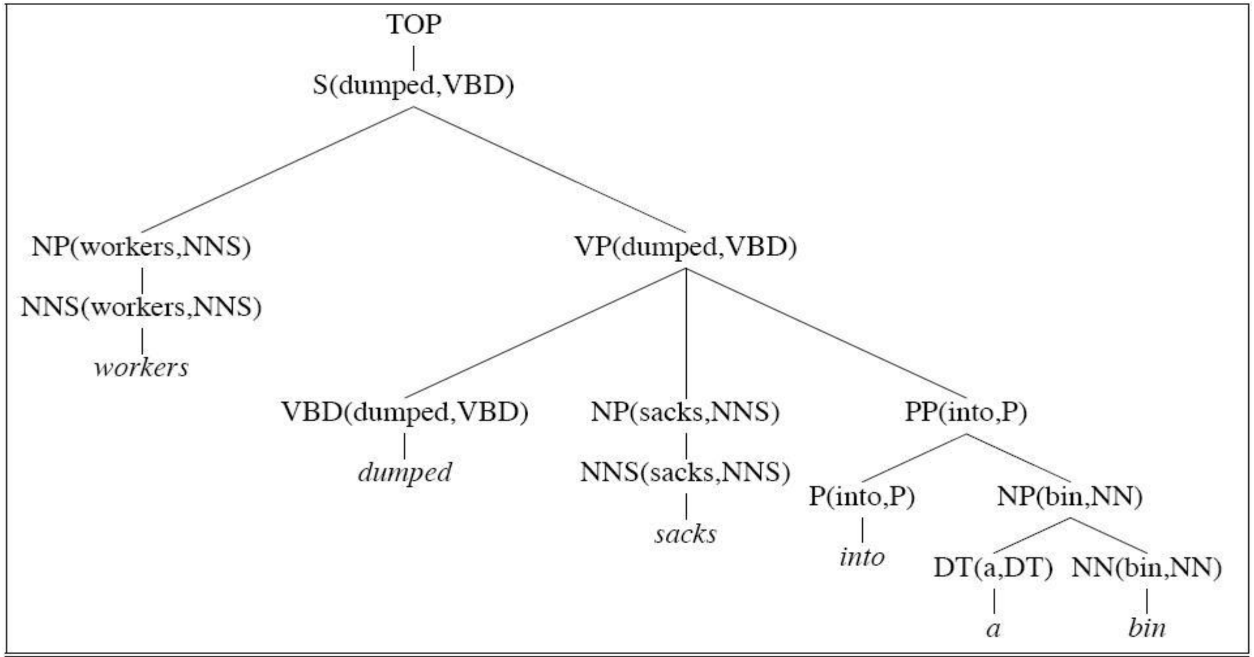
- (Review) Lexical head
  - The word in the phrase that is grammatically the most important
  - E.g. N is the head of NP
  - E.g. V is the head of VP

- Now, we put the **lexicon head** into the rules

- VP → VBD NP PP
  - VP(**dumped**) → VBD(dumped) NP(sacks) PP(into)
  - VP(**dumped,VBD**) → VBD(dumped,VBD) NP(sacks,NNS) PP(into,P)

Improving PCFG: Lexicalized PCFG

- Parse tree annotated with lexical heads for all constituents



Internal Rules					**Lexical Rules**		
TOP	→	S(dumped,VBD)			NNS(workers,NNS)	→	workers
S(dumped,VBD)	→	NP(workers,NNS)	VP(dumped,VBD)		VBD(dumped,VBD)	→	dumped
NP(workers,NNS)	→	NNS(workers,NNS)			NNS(sacks,NNS)	→	sacks
VP(dumped,VBD)	→	VBD(dumped, VBD)	NP(sacks,NNS)	PP(into,P)	P(into,P)	→	into
PP(into,P)	→	P(into,P)	NP(bin,NN)		DT(a,DT)	→	a
NP(bin,NN)	→	DT(a,DT)	NN(bin,NN)		NN(bin,NN)	→	bin

Improving PCFG: Lexicalized PCFG

- Issues
  - Treebank is not big enough to model all lexical rules
  - Most rule probabilities will come out 0
  - (We will not cover more advanced methods here to overcome this problem)

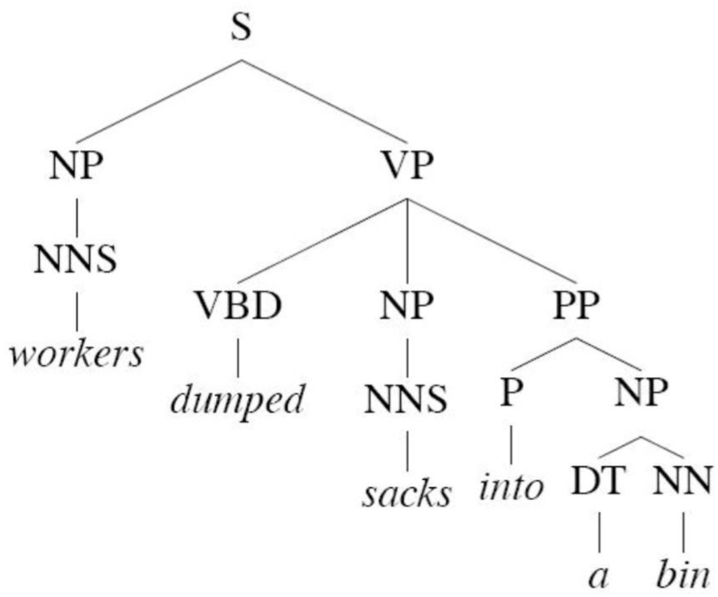
- PCFGs: 73% accuracy

- Lexicalized PCFGs: 88%

Evaluating Parsing Accuracy

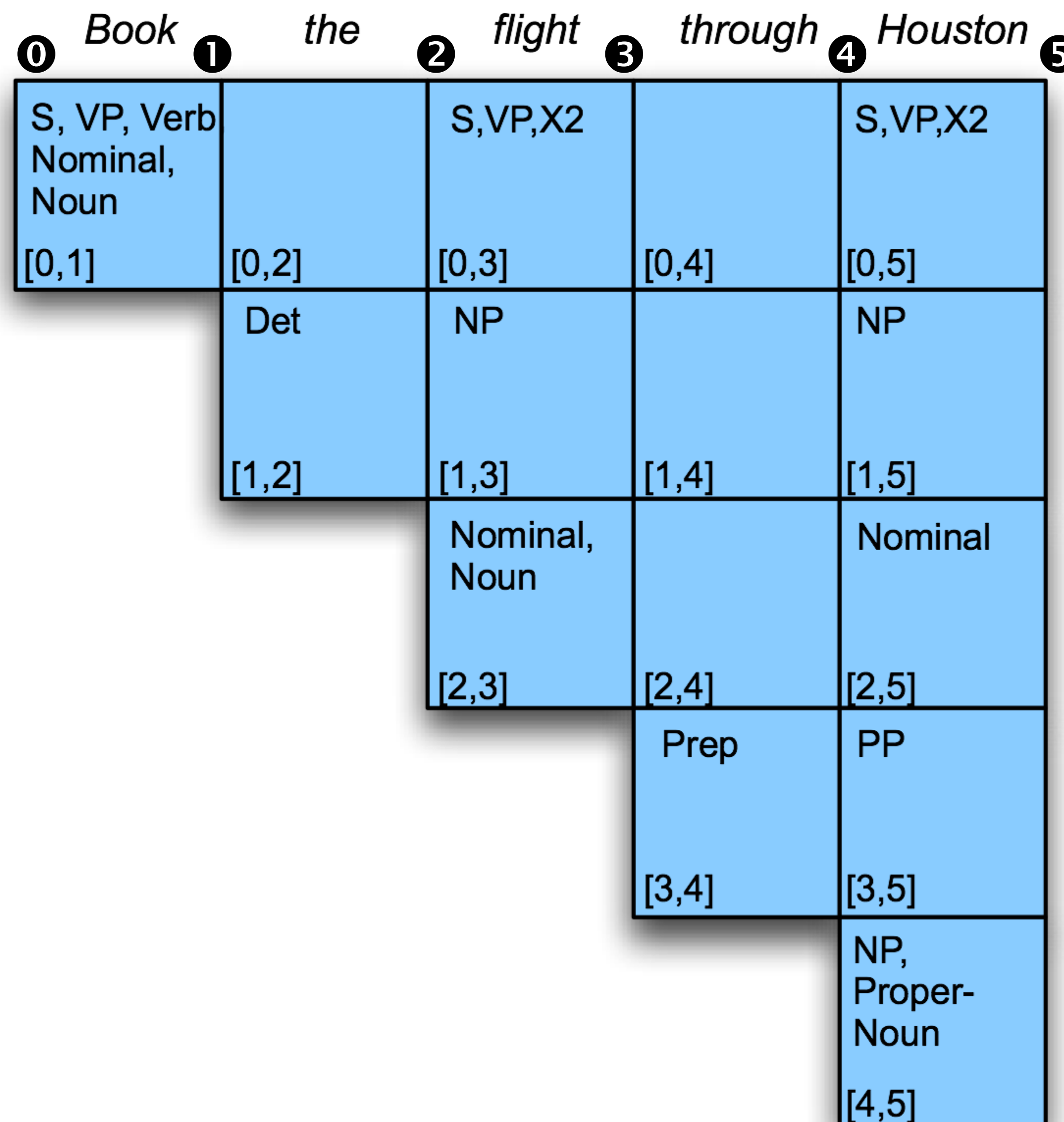
- Sentence-level accuracy
  - But most sentences are not given a completely correct parse by any existing parser

- Constituent-level accuracy
  - Constituent as labeled span: [label, start, finish], e.g. [NP, 0, 1]



Evaluating Parsing Accuracy: Labeled Spans

- [S, 0, 5]
- [VP, 0, 5]
- [Verb, 0, 1]
- [NP, 1, 5]
- [Det, 1, 2]
- [Nominal, 2, 5]
- [Noun, 2, 3]
- [PP, 3, 5]
- [Prep, 3, 4]
- [NP, 4, 5]
- [Noun, 4, 5]



Example

```
(ROOT
  (S
    (INTJ (VB Please))
    (VP (VB repeat)
      (NP (DT that)))
    (. .)))
```

```
(ROOT
  (S
    (ADVP (RB Please))
    (VP (VB repeat)
      (NP (DT that)))
    (. .)))
```

TP = 5,

FP = 2,

FN = 2

(S, 0, 3)
(INTJ, 0, 1) (VB, 0, 1)
(VP, 1, 3) (VB, 1, 2)
(NP, 2, 3) (DT, 2, 3)

(S, 0, 3)
(ADVP, 0, 1) (RB, 0, 1)
(VP, 1, 3) (VB, 1, 2)
(NP, 2, 3) (DT, 2, 3)

Example

```
(S (NP (PRP I))
   (VP (VBP need)
     (S (VP (TO to)
       (VP (VB fly)
         (PP (IN between)
           (NP (NNP Philadelphia)
             (CC and)
             (NNP Atlanta))))))) (. .)))
```

```
(S (NP (PRP I))
   (VP (VBP need)
     (S (VP (TO to)
       (VP (VB fly)
         (PP (IN between)
           (NP
             (NP (NNP Philadelphia))
               (CC and)
             (NP (NNP Atlanta)))))))) (. .)))
```

(S,0,8) (NP,0,1) (PRP,0,1)
(VP,1,8) (VBP,1,2)
(S,2,8) (VP,2,8) (TO,2,3)
(VP,3,8) (VB,3,4)
(PP,4,8) (IN,4,5)
(NP,5,8) (NNP,5,6)
(CC,6,7) (NNP,7,8)

(S,0,8) (NP,0,1) (PRP,0,1)
(VP,1,8) (VBP,1,2)
(S,2,8) (VP,2,8) (TO,2,3)
(VP,3,8) (VB,3,4)
(PP,4,8) (IN,4,5)
(NP,5,8) (NP,5,6) (NNP,5,6)
(CC,6,7) (NP,7,8) (NNP,7,8)

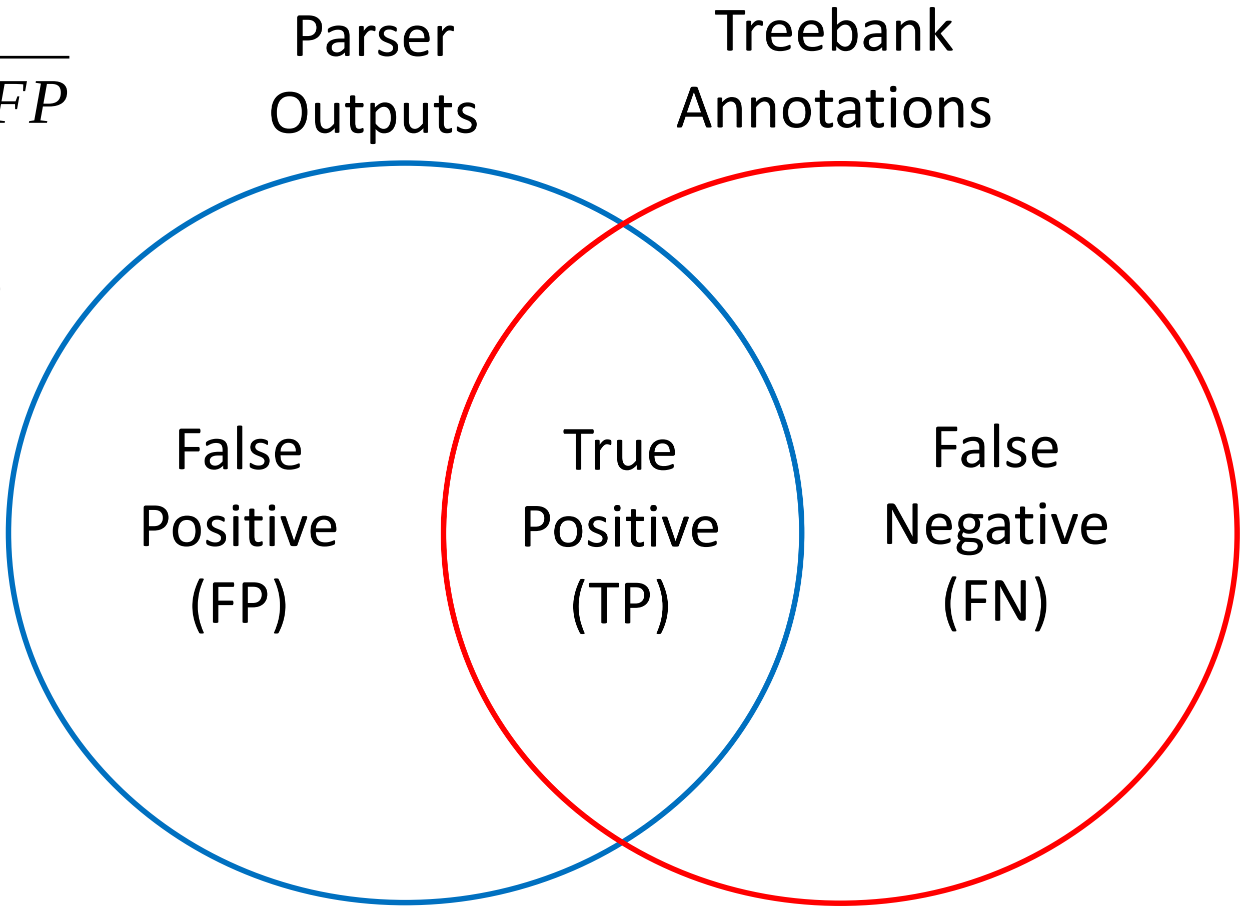
TP = 16,
FP = 0,
FN = 2

Evaluating Parsing Accuracy: Measures

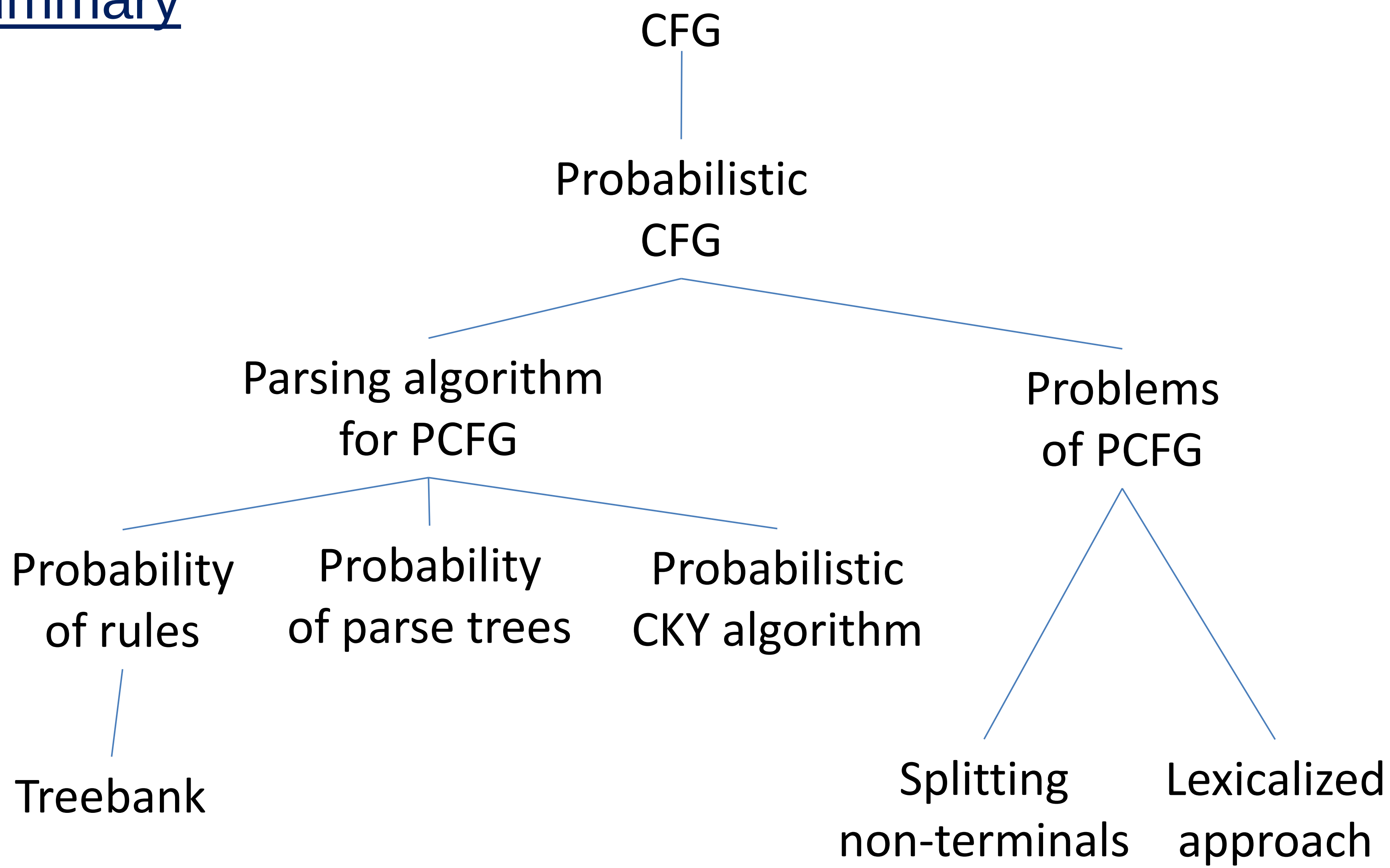
$$\text{Precision}(P) = \frac{TP}{TP+FP}$$

$$\text{Recall}(R) = \frac{TP}{TP+FN}$$

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



Summary


CFG
Probabilistic CFG
Parsing algorithm for PCFG
Problems of PCFG
Probability of rules
Probability of parse trees
Probabilistic CKY algorithm
Treebank
Splitting non-terminals
Lexicalized approach

Recap

- Probabilistic CFG
  - Treebank
  - Probabilistic CKY
  - Problems
    - Attachment ambiguities
    - Structural dependencies between rules
  - Solutions
    - Splitting non-terminals
    - Lexicalized PCFG
- Evaluating Parsing Accuracy
  - Sentence-level accuracy
  - Constituent-level accuracy