THE OHIO STATE UNIVERSITY

CSE 5525: Foundations of Speech and Language Processing

Syntax I

Huan Sun (CSE@OSU)

Many thanks to Prof. Greg Durrett @ UT Austin for sharing his slides.
Some slides adapted from Dan Klein, UC Berkeley

This Lecture

- Constituency formalism
- Context-free grammars and the CKY algorithm
- Refining grammars
- Discriminative parsers

Syntax & Constituency

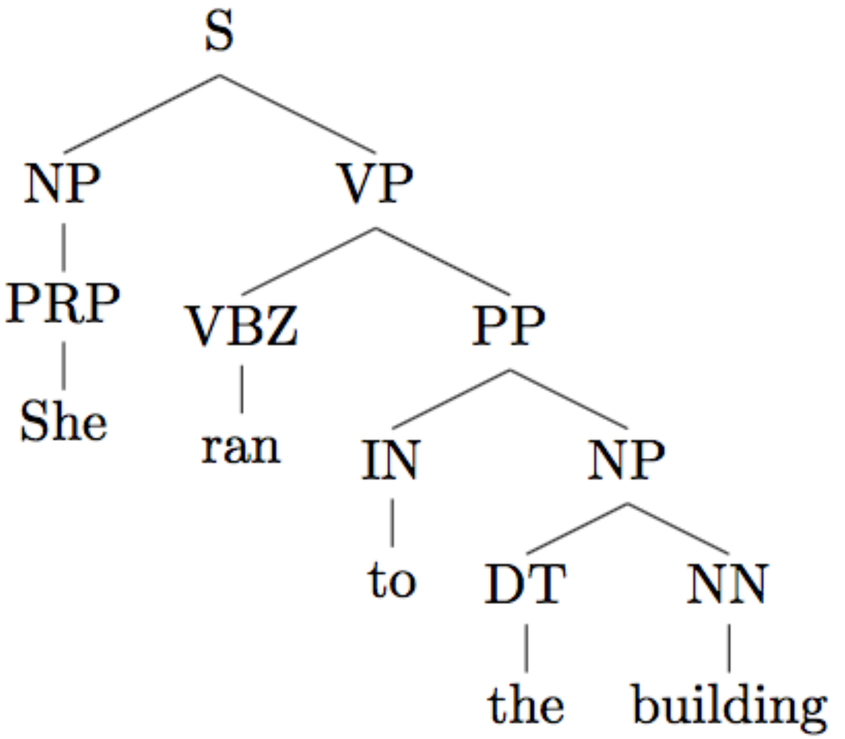
Syntax

- Study of word order and how words form sentences
- Why do we care about syntax? (useful for many tasks like phrase extraction)
  - Multiple interpretations of words (noun or verb?)
  - Recognize verb-argument structures (who is doing what to whom?)
  - Higher level of abstraction beyond words: some languages are SVO, some are VSO, some are SOV, parsing can canonicalize

Constituency Parsing

- Tree-structured syntactic analyses of sentences

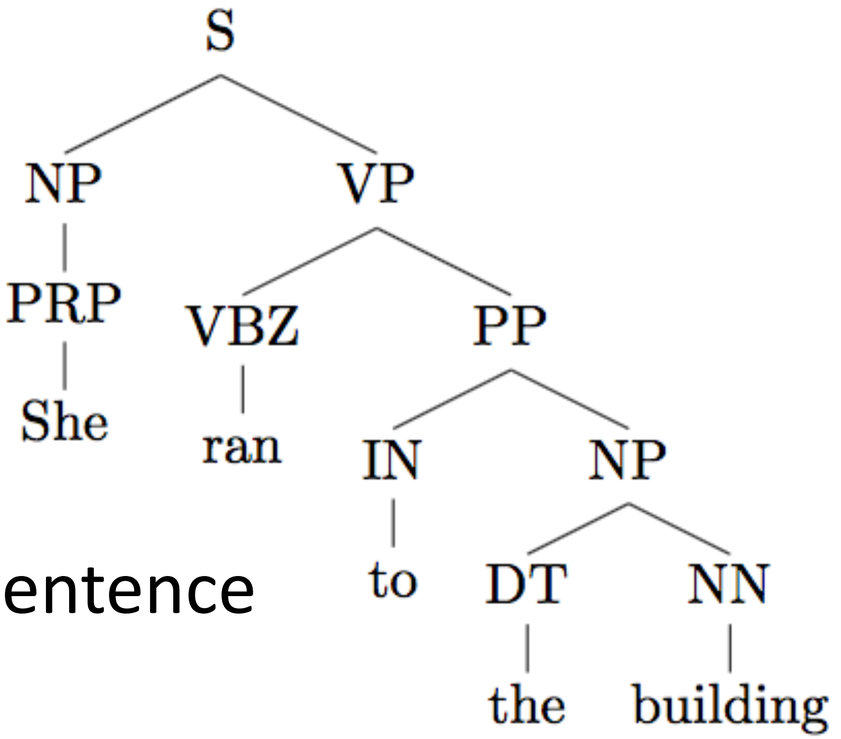
Constituency parsing is the task of breaking a text into sub-phrases, or constituents. Non-terminals in the parse tree are types of phrases, the terminals are the words in the sentence. —AllenNLP demo



Constituency Parsing

- Tree-structured syntactic analyses of sentences

Constituency parsing is the task of breaking a text into sub-phrases, or constituents. Non-terminals in the parse tree are types of phrases, the terminals are the words in the sentence. —AllenNLP demo



- Bottom layers are POS tags & words in the sentence

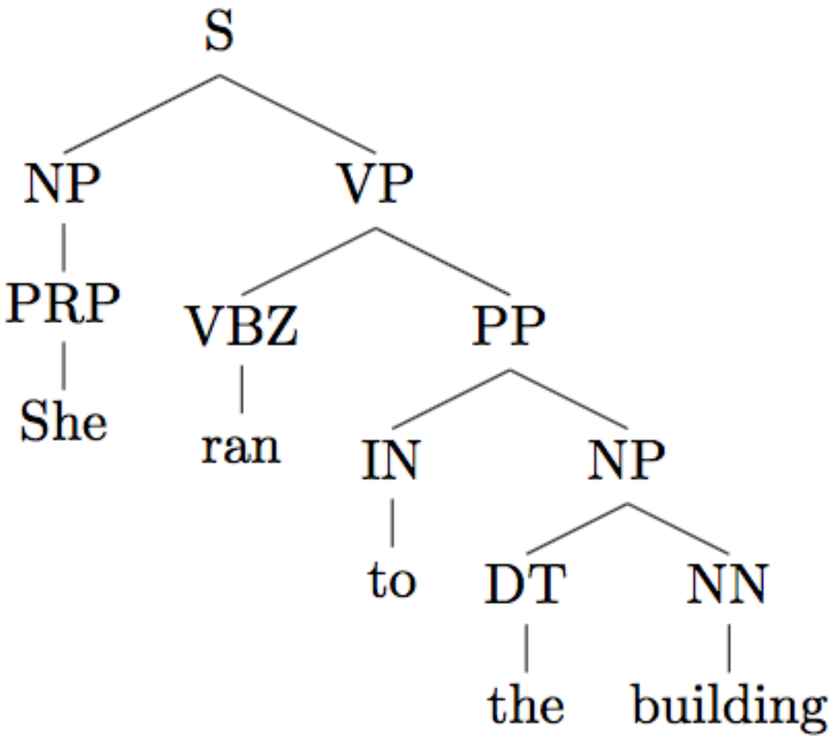
- Common things: noun phrases,
  verb phrases, prepositional phrases

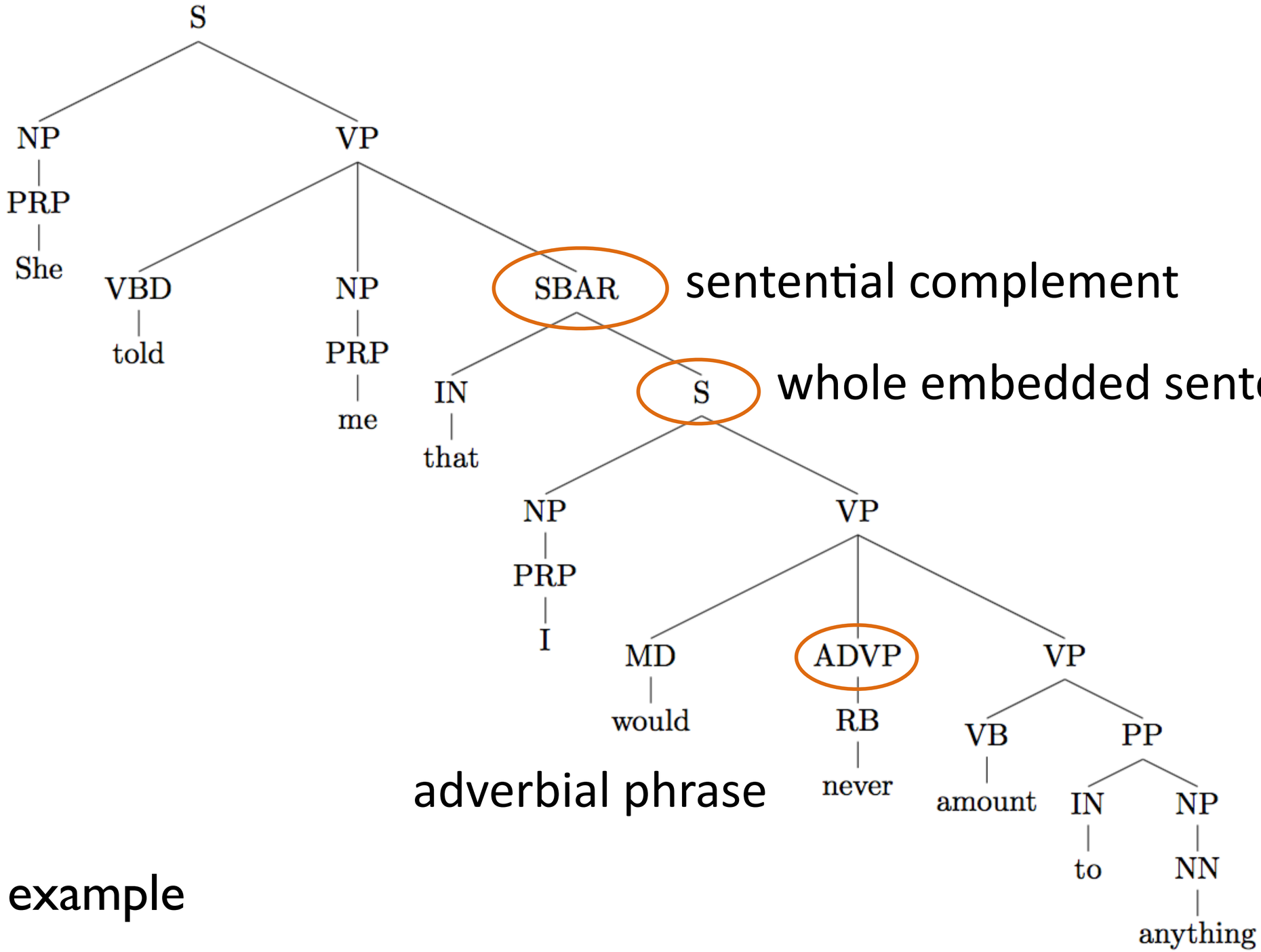
Constituency Parsing

- Tree-structured syntactic analyses of sentences

- Common things: noun phrases, verb phrases, prepositional phrases
- Bottom layers are POS tags & words in the sentence

- We will use English sentences as examples, but note that constituency makes sense for a lot of languages





sentential complement

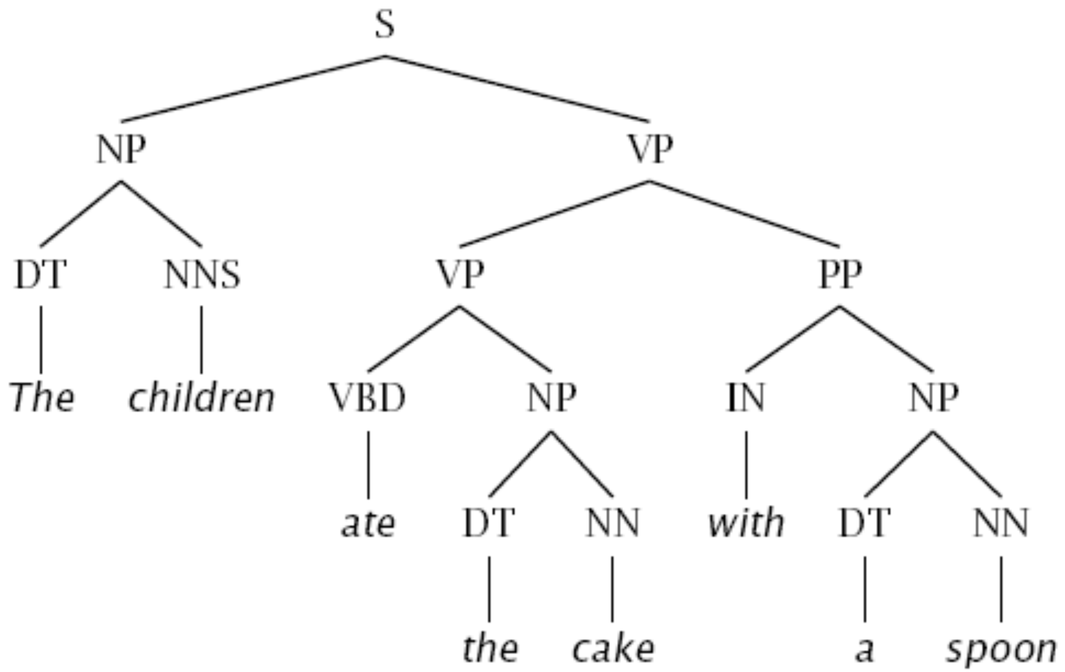
whole embedded sentence

adverbial phrase

Another example

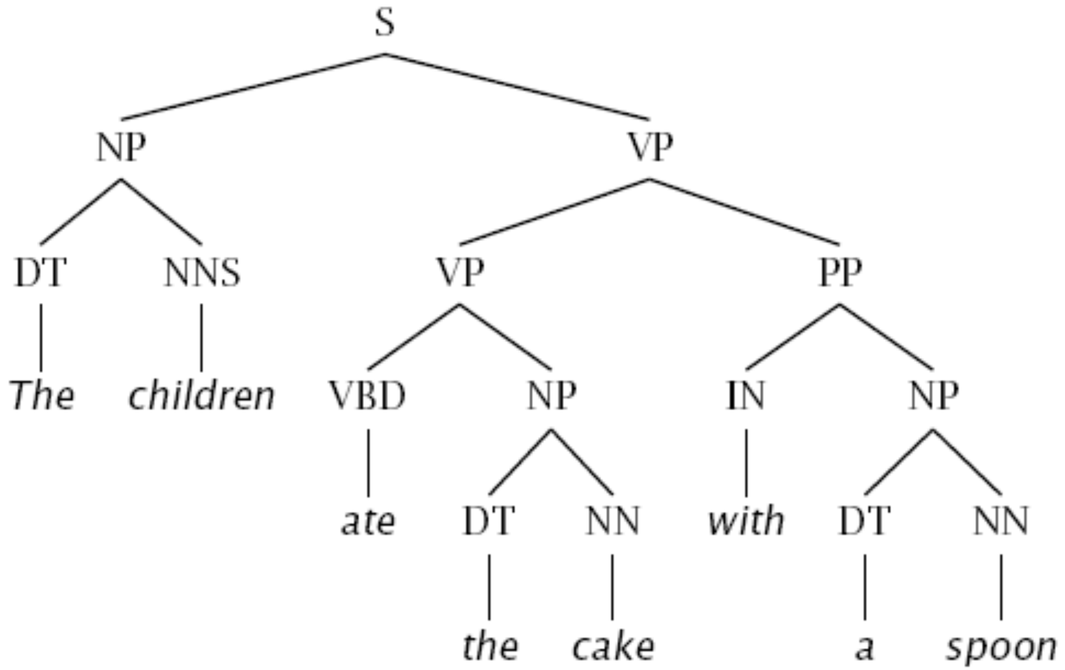
Challenges

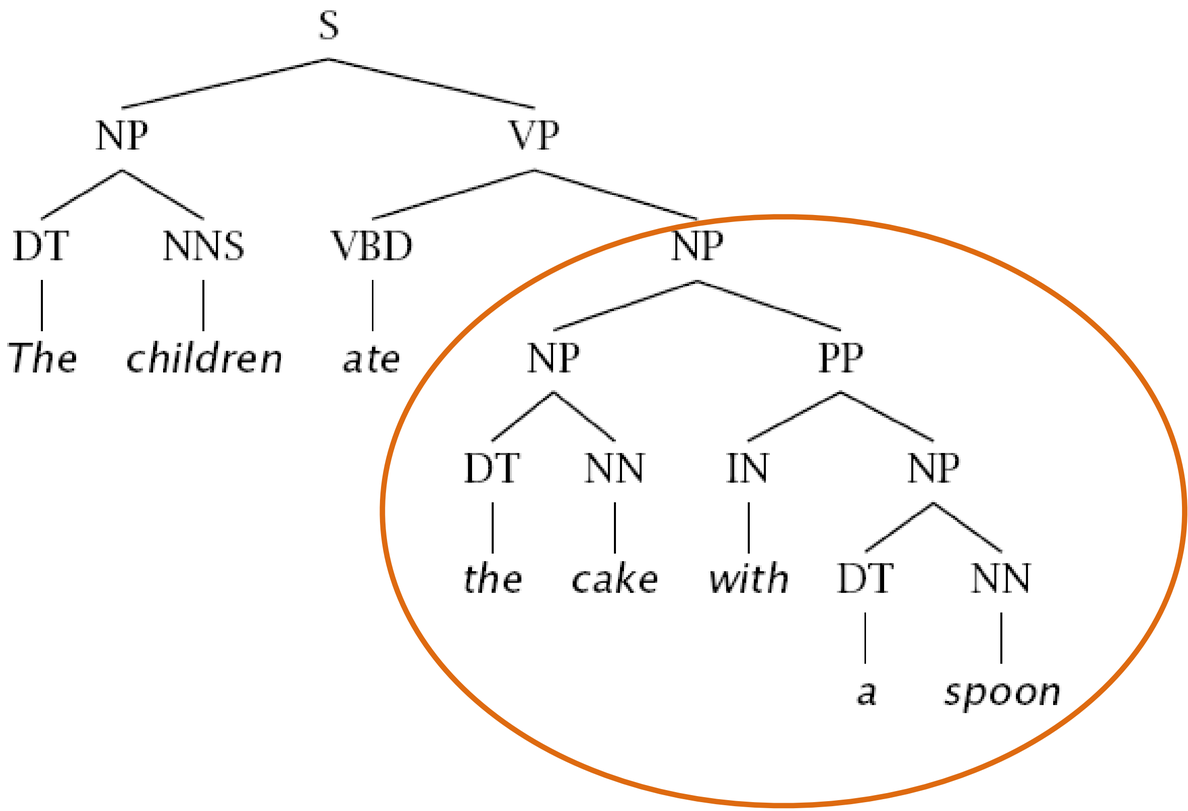
- PP (Prepositional Phrase) attachment ambiguity



Challenges

- PP (Prepositional Phrase) attachment ambiguity



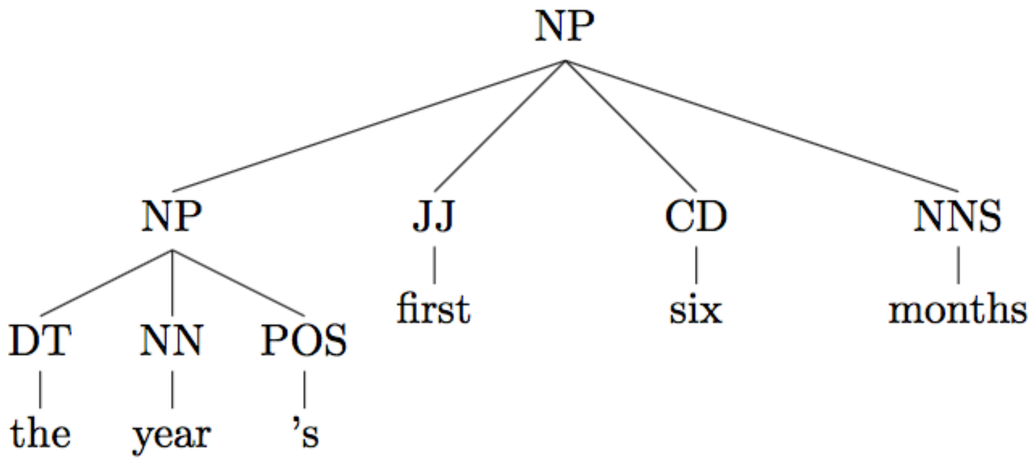


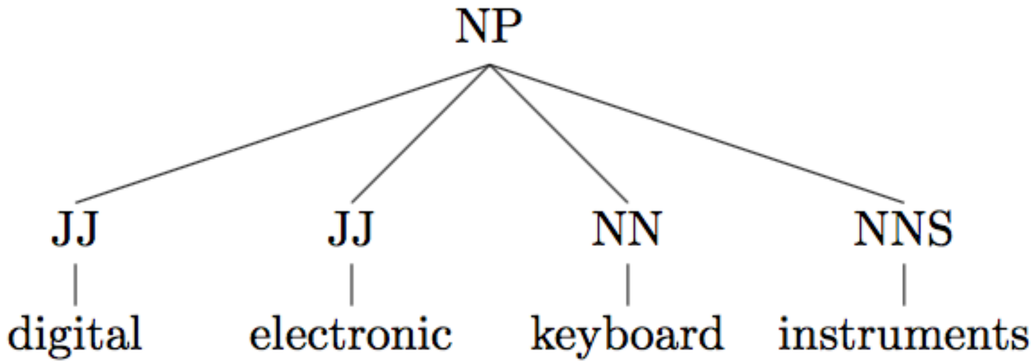
same parse as “the cake with some icing”

Challenges

- NP internal structure: tags + depth of analysis

There are a variety of components an NP can contain





Review tags, if needed: https://cs.nyu.edu/grishman/jet/guide/PennPOS.html

Coordination ambiguity

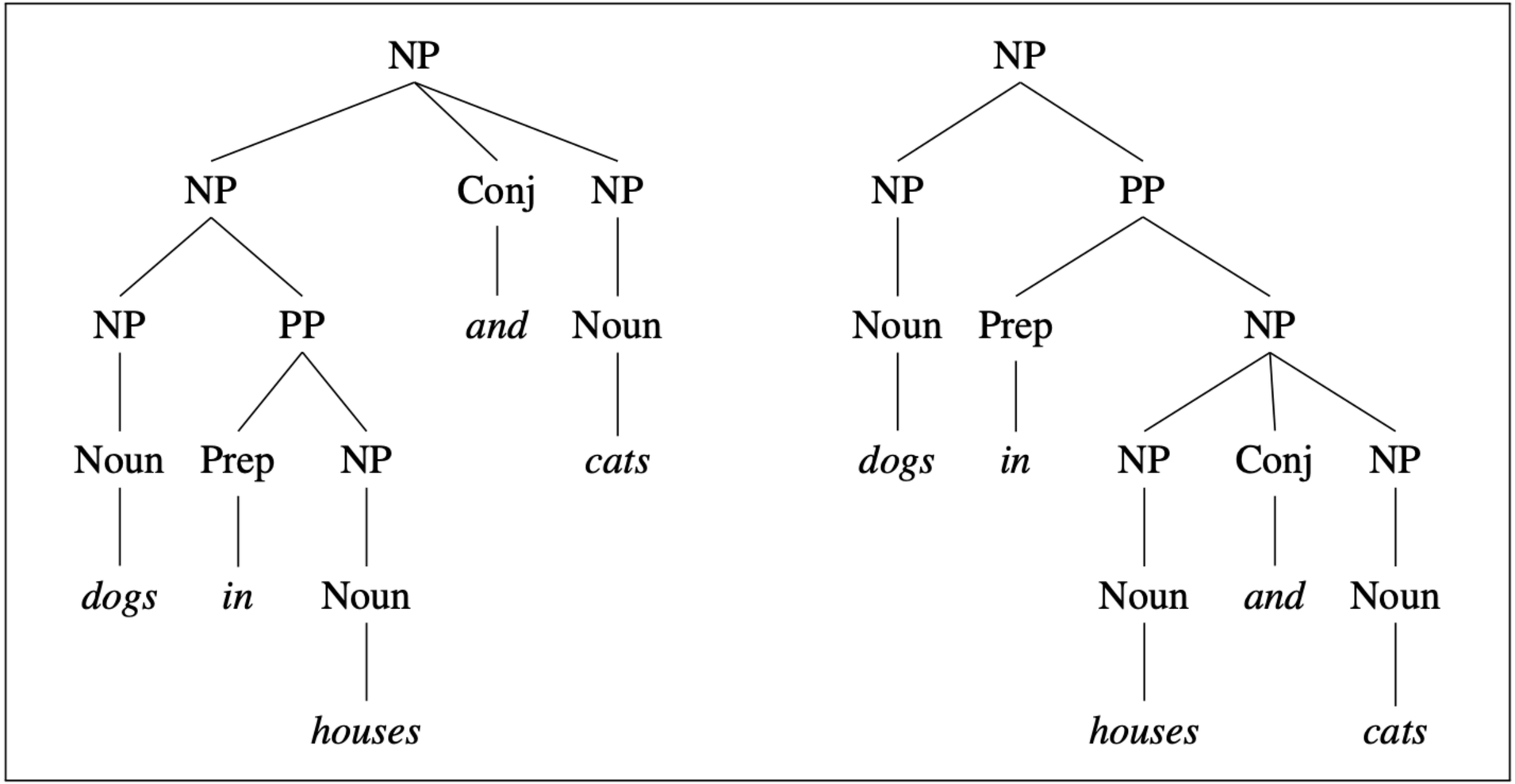
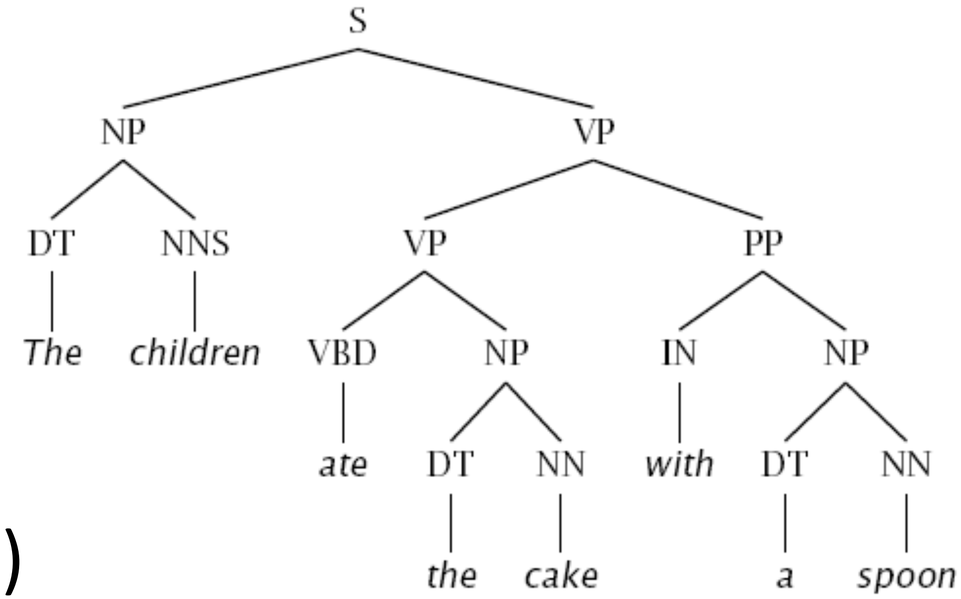


Figure 14.7 An instance of coordination ambiguity. Although the left structure is intuitively the correct one, a PCFG will assign them identical probabilities since both structures use exactly the same set of rules. After Collins (1999).

https://web.stanford.edu/~jurafsky/slp3/14.pdf

Constituency

- How do we humans know what the constituents are?
- "Constituency tests":
  - Substitution by *pro-form* (e.g., pronoun)
  - Clefting (*with a spoon is how the children* …)
  - Answer ellipsis (What did they eat? *the cake*)
    (How? *with a spoon*)



https://people.umass.edu/nconstan/201/Constituency%20Tests.pdf

Context-Free Grammars

The most widely used formal system for modeling constituent structure in English and other natural languages

Also called Phrase-Structure Grammars

Chapter 12.2 in textbook JM: https://web.stanford.edu/~jurafsky/slp3/12.pdf

CFGs

Rules (or, productions)

ROOT → S

S → NP VP

NP → DT NN

NP → NN NNS

NP → NP PP

VP → VBP NP

VP → VBP NP PP

PP → IN NP

Lexicon

NN → interest

NNS → raises

VBP → interest

VBZ → raises

- Rules/productions: symbols which rewrite as one or more symbols
  - each rule expresses the ways that symbols can be grouped and ordered together

- Lexicon consists of “preterminals” (POS tags) rewriting as terminals (words)

CFGs

Rules/productions

ROOT → S

S → NP VP

NP → DT NN

NP → NN NNS

NP → NP PP

VP → VBP NP

VP → VBP NP PP

PP → IN NP

Lexicon

NN → interest

NNS → raises

VBP → interest

VBZ → raises

The following rules express that an **NP** (or **noun phrase**) can be composed of either a *ProperNoun* or a determiner (*Det*) followed by a *Nominal*; a *Nominal* in turn can consist of one or more *Noun*s.

$$\begin{aligned} NP &\rightarrow Det\ Nominal \\ NP &\rightarrow ProperNoun \\ Nominal &\rightarrow Noun \mid Nominal\ Noun \end{aligned}$$

Example

Context-free rules can be hierarchically embedded, so we can combine the previous rules with others, like the following, that express facts about the lexicon:

$$\begin{aligned} Det &\rightarrow a \\ Det &\rightarrow the \\ Noun &\rightarrow flight \end{aligned}$$

CFGs

Rules/productions

ROOT → S

S → NP VP

NP → DT NN

NP → NN NNS

NP → NP PP

VP → VBP NP

VP → VBP NP PP

PP → IN NP

Lexicon

NN → interest

NNS → raises

VBP → interest

VBZ → raises

- Rules/productions: symbols which rewrite as one or more symbols
- Lexicon consists of "preterminals" (POS tags) rewriting as terminals (words)
- CFG is a tuple (N, T, S, R): N = nonterminals, T = terminals, S = start symbol (generally a special ROOT symbol), R = rules
  - Terminals: words in the language. Non-terminals: symbols that express abstractions over these terminals. What is the item to the left of the arrow?

CFGs

Rules/productions

ROOT → S

S → NP VP

NP → DT NN

NP → NN NNS

NP → NP PP

VP → VBP NP

VP → VBP NP PP

PP → IN NP

Lexicon

NN → interest

NNS → raises

VBP → interest

VBZ → raises

- Rules/productions: symbols which rewrite as one or more symbols
- Lexicon consists of “preterminals” (POS tags) rewriting as terminals (words)
- CFG is a tuple (N, T, S, R): N = nonterminals, T = terminals, S = start symbol (generally a special ROOT symbol), R = rules
- Probabilistic CFG (PCFG): conditional probabilities associated with rules

CFGs

Rules/productions

ROOT → S	1.0	NP → NP PP	0.3
S → NP VP	1.0	VP → VBP NP	0.7
NP → DT NN	0.2	VP → VBP NP PP	0.3
NP → NN NNS	0.5	PP → IN NP	1.0

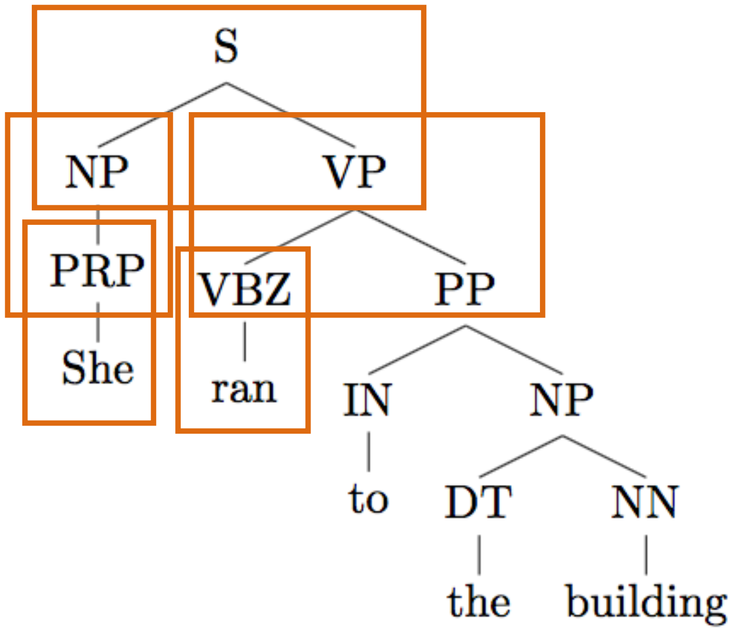
Lexicon

NN → interest	1.0
NNS → raises	1.0
VBP → interest	1.0
VBZ → raises	1.0

- Rules/productions: symbols which rewrite as one or more symbols
- Lexicon consists of “preterminals” (POS tags) rewriting as terminals (words)
- CFG is a tuple (N, T, S, R): N = nonterminals, T = terminals, S = start symbol (generally a special ROOT symbol), R = rules
- PCFG: conditional probabilities associated with rules (num. are made up)

Estimating PCFGs

- Tree *T* is a series of rule applications *r*. (Each rule expands a non-terminal node.)



T includes:

S → NP VP

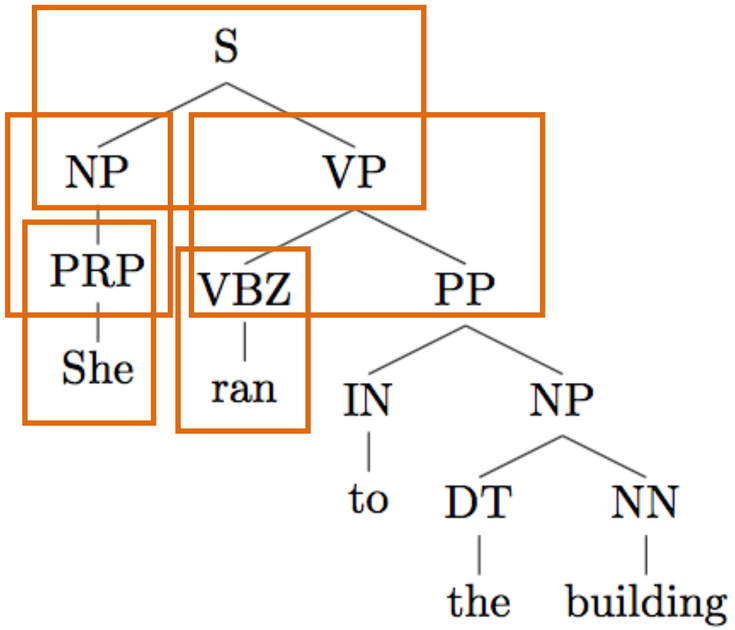
NP → PRP

NP → DT NN

…

Estimating PCFGs

- Tree *T* is a series of rule applications *r*. $P(T) = \prod_{r \in T} P(r|\text{parent}(r))$



T includes:

S → NP VP

NP → PRP

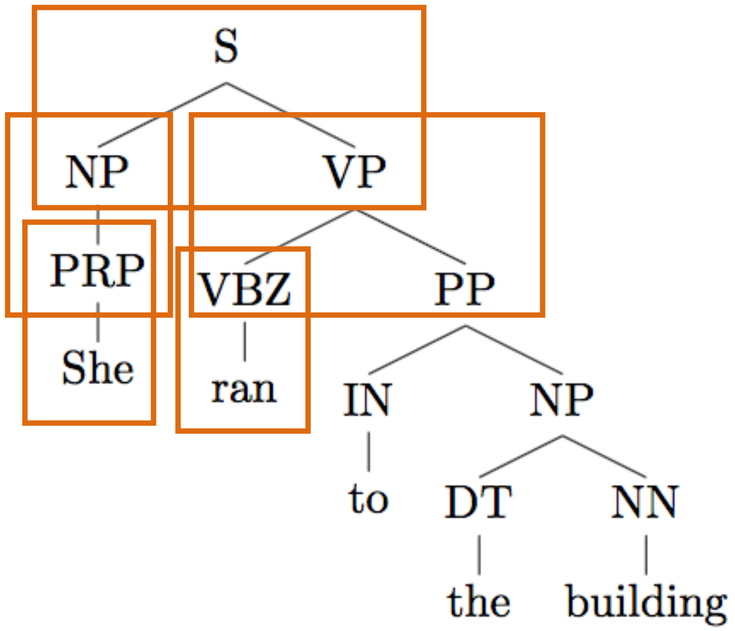
NP → DT NN

…

The probability of a particular parse tree T is defined as the product of the probabilities of all the rules used to expand each of the non-terminal nodes in the parse tree T

Estimating PCFGs

- Tree *T* is a series of rule applications *r*. $P(T) = \prod_{r \in T} P(r|\text{parent}(r))$



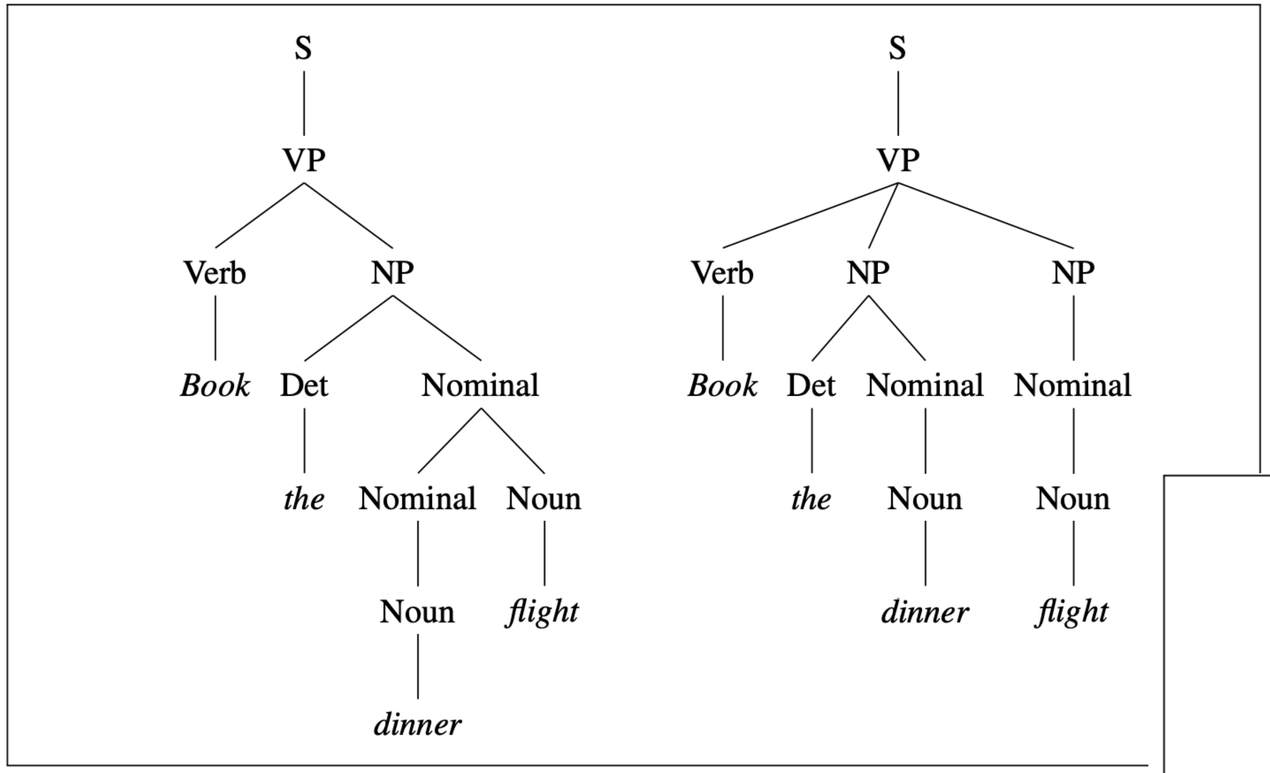
S → NP VP 1.0

NP → PRP 0.5

NP → DT NN 0.5

…

- Maximum likelihood PCFG: count and normalize! Same as HMMs / Naive Bayes

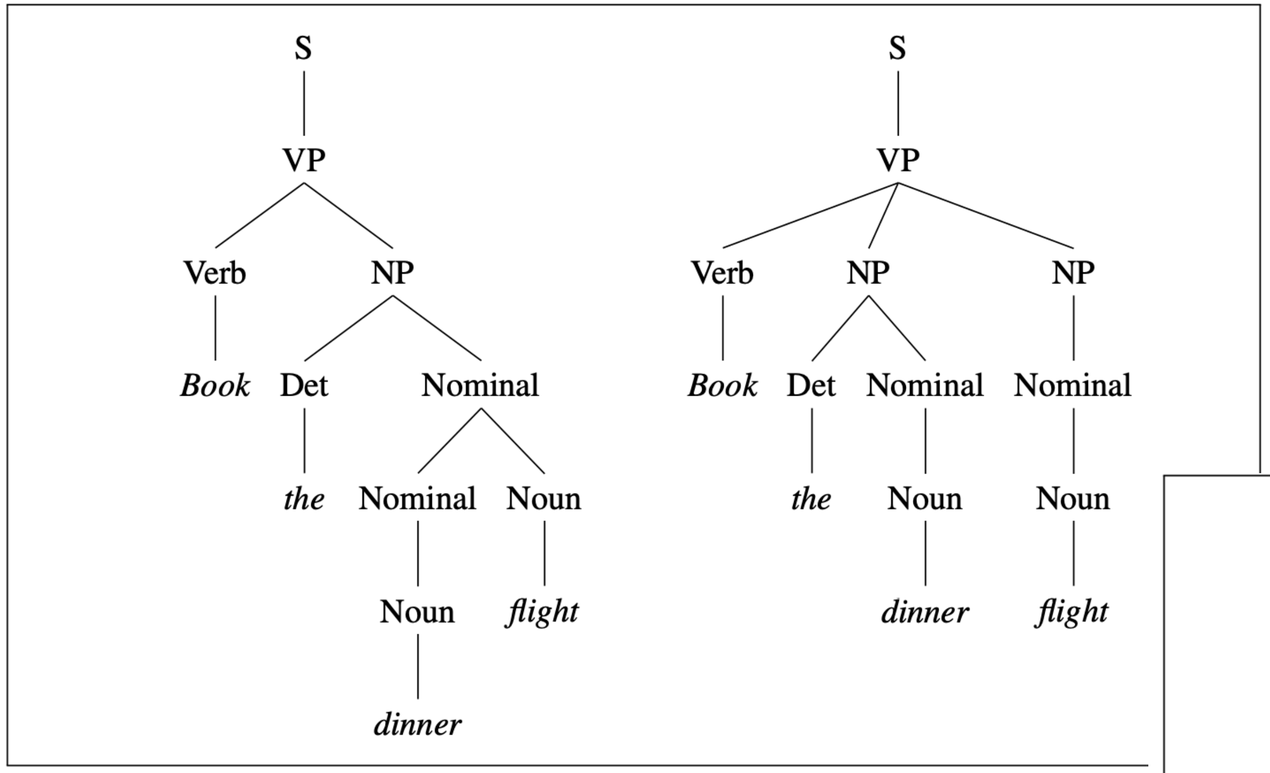


Rules			P	Rules			P
S	→	VP	.05	S	→	VP	.05
VP	→	Verb NP	.20	VP	→	Verb NP NP	.10
NP	→	Det Nominal	.20	NP	→	Det Nominal	.20
Nominal	→	Nominal Noun	.20	NP	→	Nominal	.15
Nominal	→	Noun	.75	Nominal	→	Noun	.75
				Nominal	→	Noun	.75
Verb	→	book	.30	Verb	→	book	.30
Det	→	the	.60	Det	→	the	.60
Noun	→	dinner	.10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

Figure 14.2 Two parse trees for an ambiguous sentence. The parse on the left corresponds to the sensible meaning "Book a flight that serves dinner", while the parse on the right corresponds to the nonsensical meaning "Book a flight on behalf of 'the dinner' ".

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

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



How to produce the most-likely parse T?

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

Figure 14.2 Two parse trees for an ambiguous sentence. The parse on the left corresponds to the sensible meaning "Book a flight that serves dinner", while the parse on the right corresponds to the nonsensical meaning "Book a flight on behalf of 'the dinner' ".

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

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