# GRAMMARS

David Kauchak
CS159 – Fall 2020

*some slides adapted from Ray Mooney*

## Admin

Assignment 3 out today: due next Wednesday

Quiz

## Context free grammar

$S \rightarrow NP\ VP$

left hand side (single symbol)

right hand side (one or more symbols)

## Formally…

G = (NT, T, P, S)

NT: finite set of nonterminal symbols

T: finite set of terminal symbols, NT and T are disjoint

P: finite set of productions of the form
$A \rightarrow \alpha$, $A \in NT$ and $\alpha \in (T \cup NT)^*$

$S \in NT$: start symbol

## CFG: Example

Many possible CFGs for English, here is an example (fragment):

S → NP VP

VP → V NP

NP → DetP N | DetP AdjP N

AdjP → Adj | Adv AdjP

N → boy | girl

V → sees | likes

Adj → big | small

Adv → very

DetP → a | the

## Derivations in a CFG

S → NP VP
VP → V NP
NP → DetP N | DetP AdjP N
AdjP → Adj | Adv AdjP
N → boy | girl
V → sees | likes
Adj → big | small
Adv → very
DetP → a | the

**S**

What can we do?

## Derivations in a CFG

**S → NP VP**
VP → V NP
NP → DetP N | DetP AdjP N
AdjP → Adj | Adv AdjP
N → boy | girl
V → sees | likes
Adj → big | small
Adv → very
DetP → a | the

S

## Derivations in a CFG

S → NP VP
VP → V NP
NP → DetP N | DetP AdjP N
AdjP → Adj | Adv AdjP
N → boy | girl
V → sees | likes
Adj → big | small
Adv → very
DetP → a | the

NP VP

What can we do?

## Derivations in a CFG

S → NP VP
**VP → V NP**
**NP → DetP N | DetP AdjP N**
AdjP → Adj | Adv AdjP
N → boy | girl
V → sees | likes
Adj → big | small
Adv → very
DetP → a | the

NP VP

## Derivations in a CFG

S → NP VP
VP → V NP
**NP → DetP N** | DetP AdjP N
AdjP → Adj | Adv AdjP
N → boy | girl
V → sees | likes
Adj → big | small
Adv → very
DetP → a | the

DetP N VP

## Derivations in a CFG

S → NP VP
VP → V NP
NP → DetP N | DetP AdjP N
AdjP → Adj | Adv AdjP
**N → boy** | girl
V → sees | likes
Adj → big | small
Adv → very
**DetP →** a | **the**

DetP N VP

## Derivations in a CFG

S → NP VP
**VP → V NP**
NP → DetP N | DetP AdjP N
AdjP → Adj | Adv AdjP
N → boy | girl
**V →** sees | **likes**
Adj → big | small
Adv → very
DetP → a | the

the boy VP

## Derivations in a CFG

S → NP VP
VP → V NP
**NP → DetP N** | DetP AdjP N
AdjP → Adj | Adv AdjP
**N →** boy | **girl**
V → sees | likes
Adj → big | small
Adv → very
**DetP → a** | the

the boy likes NP

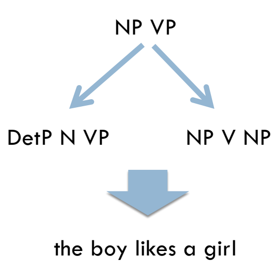
## Derivations in a CFG

S → NP VP
VP → V NP
NP → DetP N | DetP AdjP N
AdjP → Adj | Adv AdjP
N → boy | girl
V → sees | likes
Adj → big | small
Adv → very
DetP → a | the

the boy likes a girl

## Derivations in a CFG; Order of Derivation Irrelevant

S → NP VP
**VP → V NP**
**NP → DetP N** | DetP AdjP N
AdjP → Adj | Adv AdjP
N → boy | girl
V → sees | likes
Adj → big | small
Adv → very
DetP → a | the

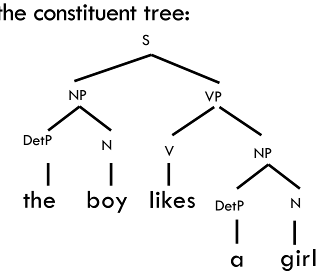



## Derivations of CFGs

String rewriting system: we derive a string

Derivation history shows the constituent tree:

the boy likes a girl



## Parsing

Parsing is the field of NLP interested in automatically determining the syntactic structure of a sentence

Parsing can be thought of as determining what sentences are "valid" English sentences

As a byproduct, we often can get the structure

## Parsing

Given a CFG and a sentence, determine the possible parse tree(s)

S -> NP VP
NP -> N
NP -> PRP
NP -> N PP
VP -> V NP
VP -> V NP PP
PP -> IN N
PRP -> I
V -> eat
N -> sushi
N -> tuna
IN -> with

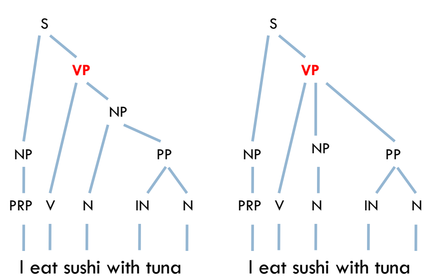
I eat sushi with tuna

What parse trees are possible for this sentence?

How did you do it?

What if the grammar is much larger?

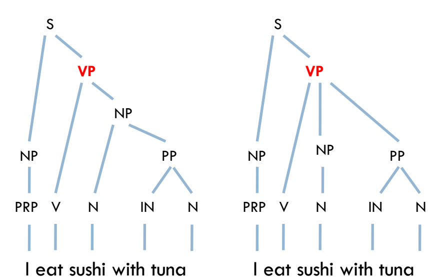
## Parsing




S -> NP VP
NP -> PRP
NP -> N PP
NP -> N
VP -> V NP
VP -> V NP PP
PP -> IN N
PRP -> I
V -> eat
N -> sushi
N -> tuna
IN -> with

What is the difference between these parses?

## Parsing ambiguity



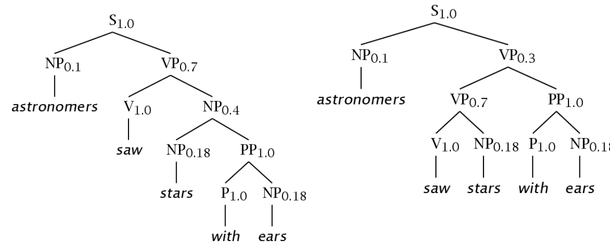

S -> NP VP
NP -> PRP
NP -> N PP
NP -> N
VP -> V NP
VP -> V NP PP
PP -> IN N
PRP -> I
V -> eat
N -> sushi
N -> tuna
IN -> with

How can we decide between these?

## A Simple PCFG

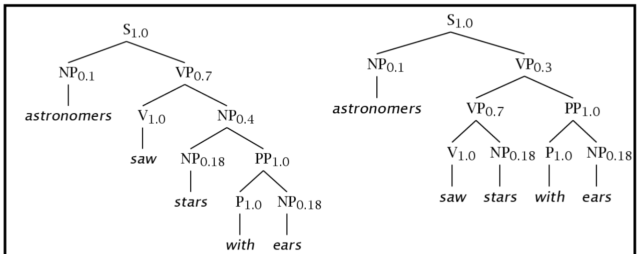
Probabilities!

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| S | → | NP VP | 1.0 | NP | → | NP PP | 0.4 |
| VP | → | V NP | 0.7 | NP | → | *astronomers* | 0.1 |
| VP | → | VP PP | 0.3 | NP | → | *ears* | 0.18 |
| PP | → | P NP | 1.0 | NP | → | *saw* | 0.04 |
| P | → | *with* | 1.0 | NP | → | *stars* | 0.18 |
| V | → | *saw* | 1.0 | NP | → | *telescope* | 0.1 |



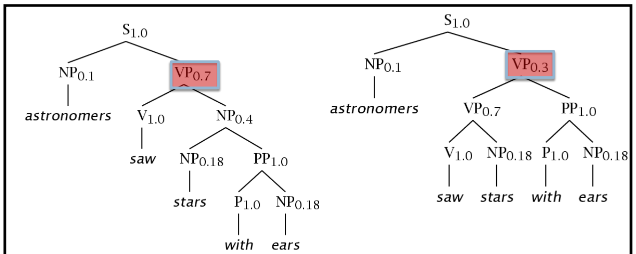

Just like *n*-gram language modeling, PCFGs break the sentence generation process into smaller steps/probabilities

The probability of a parse is the product of the PCFG rules




What are the different interpretations here?

Which do you think is more likely?




= 1.0 * 0.1 * 0.7 * 1.0 * 0.4 * 0.18
    * 1.0 * 1.0 * 0.18
= 0.0009072

= 1.0 * 0.1 * 0.3 * 0.7 * 1.0 * 0.18
    * 1.0 * 1.0 * 0.18
= 0.0006804

## Parsing problems

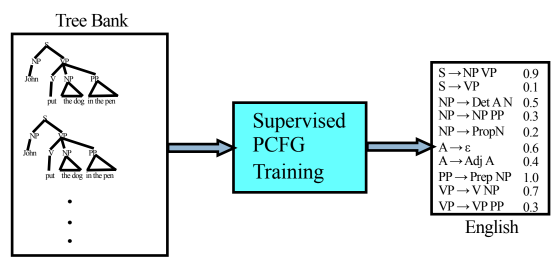
Pick a model
- e.g. CFG, PCFG, …

Train (or learn) a model
- What CFG/PCFG rules should I use?
- Parameters (e.g. PCFG probabilities)?
- What kind of data do we have?

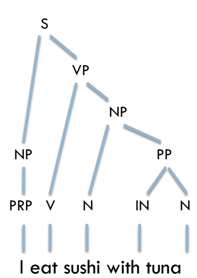
Parsing
- Determine the parse tree(s) given a sentence

## PCFG: Training

If we have example parsed sentences, how can we learn a set of PCFGs?




## Extracting the rules




S → NP VP
NP → PRP
PRP → I
VP → V NP
V → eat
NP → N PP
N → sushi
PP → IN N
IN → with
N → tuna

What CFG rules occur in this tree?

## Estimating PCFG Probabilities

We can extract the rules from the trees

S → NP VP
NP → PRP
PRP → I
VP → V NP
V → eat
NP → N PP
N → sushi
…

| | | | |
|---|---|---|---|
| S | → | NP VP | 1.0 |
| VP | → | V NP | 0.7 |
| VP | → | VP PP | 0.3 |
| PP | → | P NP | 1.0 |
| P | → | *with* | 1.0 |
| V | → | *saw* | 1.0 |

How do we go from the extracted CFG rules to PCFG rules?

## Estimating PCFG Probabilities

Extract the rules from the trees

Calculate the probabilities using MLE

$\alpha \rightarrow \beta$ ⇒ $p(\alpha \rightarrow \beta \mid \alpha)$

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

## Estimating PCFG Probabilities

| Rule | Occurrences |
|---|---|
| S → NP VP | 10 |
| S → V NP | 3 |
| S → VP PP | 2 |
| NP → N | 7 |
| NP → N PP | 3 |
| NP → DT N | 6 |

P( S → V NP) = ?

$$P(S \rightarrow V\ NP) = P(S \rightarrow V\ NP \mid S) = \frac{\text{count}(S \rightarrow V\ NP)}{\text{count}(S)} = \frac{3}{15}$$

## Grammar Equivalence

What does it mean for two grammars to be equal?

## Grammar Equivalence

Weak equivalence: grammars generate the same set of strings

- Grammar 1: NP → DetP N and DetP → a | the
- Grammar 2: NP → a N | the N

Strong equivalence: grammars have the same set of derivation trees

- With CFGs, possible only with useless rules
- Grammar 2: NP → a N | the N
- Grammar 3: NP → a N | the N, DetP → many

## Normal Forms

There are weakly equivalent normal forms (Chomsky Normal Form, Greibach Normal Form)

A CFG is in Chomsky Normal Form (CNF) if all productions are of one of two forms:

- A → B C with A, B, C nonterminals
- A → a, with A a nonterminal and a a terminal

*Every CFG has a weakly equivalent CFG in CNF*

## CNF Grammar

S -> VP
VP -> VB NP
VP -> VB NP PP
NP -> DT NN
NP -> NN
NP -> NP PP
PP -> IN NP
DT -> the
IN -> with
VB -> film
VB -> trust
NN -> man
NN -> film
NN -> trust

S -> VP
VP -> VB NP
VP -> VP2 PP
VP2 -> VB NP
NP -> DT NN
NP -> NN
NP -> NP PP
PP -> IN NP
DT -> the
IN -> with
VB -> film
VB -> trust
NN -> man
NN -> film
NN -> trust

## Probabilistic Grammar Conversion

<table>
<tr><th>Original Grammar</th><th></th><th>Chomsky Normal Form</th><th></th></tr>
<tr><td>S → NP VP</td><td>0.8</td><td>S → NP VP</td><td>0.8</td></tr>
<tr><td>S → Aux NP VP</td><td>0.1</td><td>S → X1 VP<br>X1 → Aux NP</td><td>0.1<br>1.0</td></tr>
<tr><td>S → VP</td><td>0.1</td><td>S → book | include | prefer<br>0.01 0.004 0.006<br>S → Verb NP<br>S → VP PP</td><td><br><br>0.05<br>0.03</td></tr>
<tr><td>NP → Pronoun</td><td>0.2</td><td>NP → I | he | she | me<br>0.1 0.02 0.02 0.06</td><td></td></tr>
<tr><td>NP → Proper-Noun</td><td>0.2</td><td>NP → Houston | NWA<br>0.16 .04</td><td></td></tr>
<tr><td>NP → Det Nominal</td><td>0.6</td><td>NP → Det Nominal</td><td>0.6</td></tr>
<tr><td>Nominal → Noun</td><td>0.3</td><td>Nominal → book | flight | meal | money<br>0.03 0.15 0.06 0.06</td><td></td></tr>
<tr><td>Nominal → Nominal Noun</td><td>0.2</td><td>Nominal → Nominal Noun</td><td>0.2</td></tr>
<tr><td>Nominal → Nominal PP</td><td>0.5</td><td>Nominal → Nominal PP</td><td>0.5</td></tr>
<tr><td>VP → Verb</td><td>0.2</td><td>VP → book | include | prefer<br>0.1 0.04 0.06</td><td></td></tr>
<tr><td>VP → Verb NP</td><td>0.5</td><td>VP → Verb NP</td><td>0.5</td></tr>
<tr><td>VP → VP PP</td><td>0.3</td><td>VP → VP PP</td><td>0.3</td></tr>
<tr><td>PP → Prep NP</td><td>1.0</td><td>PP → Prep NP</td><td>1.0</td></tr>
</table>

## Grammar questions

Can we determine if a sentence is grammatical?

Given a sentence, can we determine the syntactic structure?

Can we determine how likely a sentence is to be grammatical? to be an English sentence?

Next time: parsing

Can we generate candidate, grammatical sentences?