GRAMMARS

David Kauchak
CS159 – Spring 2023
some slides adapted from Ray Mooney

1

Admin

Assignment 3 out today: due next Monday

Quiz

2

Context free grammar

$S \rightarrow NP\ VP$

left hand side (single symbol)

right hand side (one or more symbols)

3

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

4

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 → kid | dog

V → sees | likes

Adj → big | small

Adv → very

DetP → a | the

5

Derivations in a CFG

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

S

What can we do?

6

Derivations in a CFG

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

S

7

Derivations in a CFG

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

NP VP

What can we do?

8

Derivations in a CFG

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

NP VP

9

Derivations in a CFG

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

DetP N VP

10

Derivations in a CFG

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

DetP N VP

11

Derivations in a CFG

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

the kid VP

12

Derivations in a CFG

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

the kid likes NP

13

Derivations in a CFG

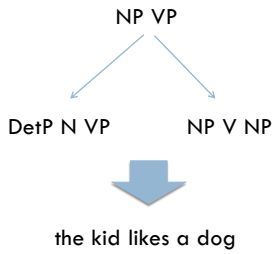
S → NP VP
VP → V NP
NP → DetP N | DetP AdjP N
AdjP → Adj | Adv AdjP
N → kid | dog
V → sees | likes
Adj → big | small
Adv → very
DetP → a | the

the kid likes a dog

14

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 → kid | dog
V → sees | likes
Adj → big | small
Adv → very
DetP → a | the



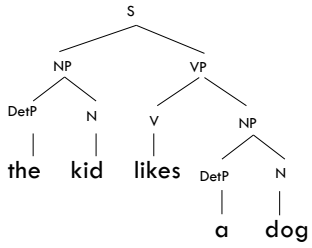
15

Derivations of CFGs

String rewriting system: we derive a string

Derivation history shows the constituent tree:

the kid likes a dog



16

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

17

Parsing

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

I eat sushi with tuna

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

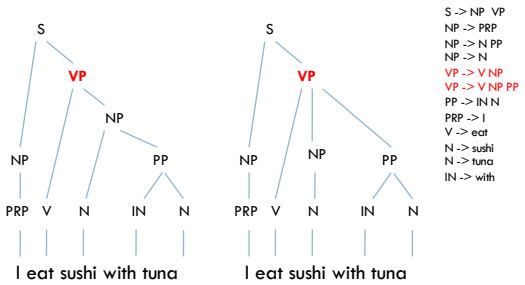
What parse trees are possible for this sentence?

How did you do it?

What if the grammar is much larger?

18

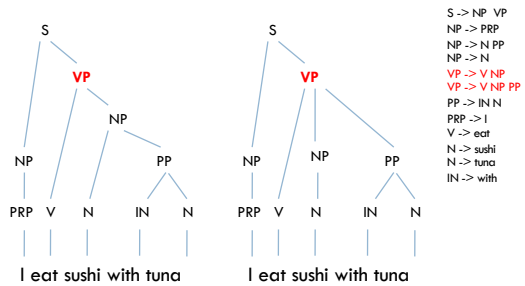
Parsing



What is the difference between these parses?

19

Parsing ambiguity



How can we decide between these?

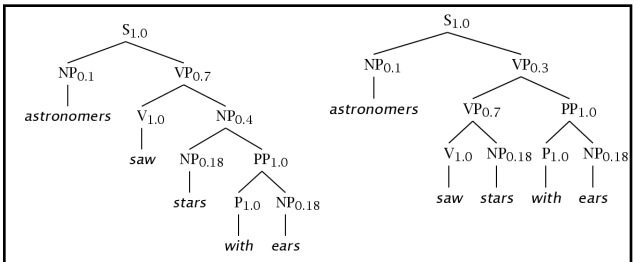
20

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

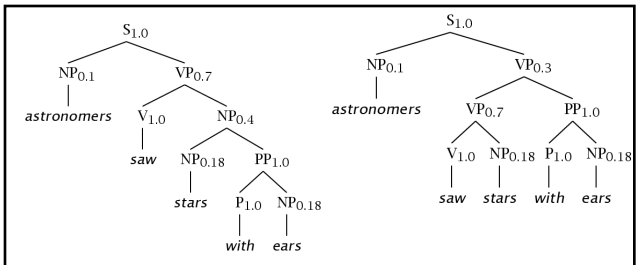
21



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

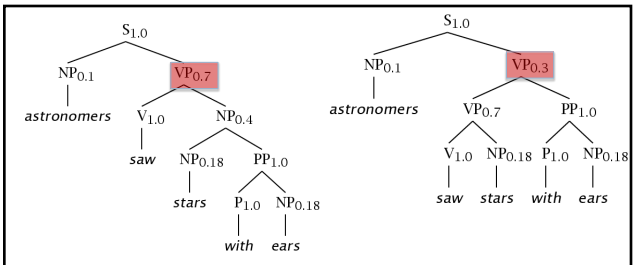
22



What are the different interpretations here?

Which do you think is more likely?

23



= 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

24

Parsing problems

Pick a model
- e.g. CFG, PCFG, …

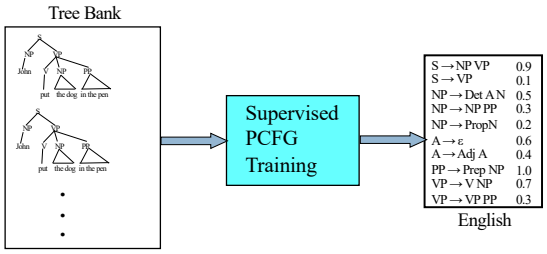
Train (or learn) a model
- What CFG/PCFG rules should I use?
- Parameters (e.g. PCFG probabilities)?
- What kind of data do we have?

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

25

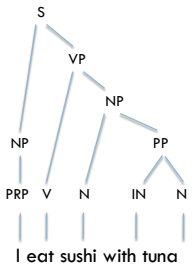
PCFG: Training

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



26

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?

27

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?

28

Estimating PCFG Probabilities

Extract the rules from the trees

Calculate the probabilities using MLE

$$\alpha \rightarrow \beta \quad \Rightarrow \quad p(\alpha \rightarrow \beta \mid \alpha)$$

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

29

Estimating PCFG Probabilities

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

30

Grammar Equivalence

What does it mean for two grammars to be equal?

31

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

32

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 → α, with A a nonterminal and α a terminal

Every CFG has a weakly equivalent CFG in CNF

33

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

34

Probabilistic Grammar Conversion

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

35

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?

37