

PCFGs: Parameter Estimation, Lexicalization and
Parsing

Informatics 2A: Lecture 19

Bonnie Webber and Frank Keller

School of Informatics
University of Edinburgh
bonnie@inf.ed.ac.uk

6 November 2009

Parameter Estimation

Where do we get the parameters (i.e., rule probabilities) for a PCFG?

The easiest way is from a large **parsed corpus** such as the Penn Treebank.

Given a large parsed corpus, we can obtain:

- the **grammar rules** by reading them off the trees in the corpus;
- the **rule probabilities** by comparing the frequency with which a given rule occurs in the corpus, compared with the frequency of its LHS. That is,

$$P(\alpha \rightarrow \beta | \alpha) = \frac{\text{Count}(\alpha \rightarrow \beta)}{\sum_{\gamma} \text{Count}(\alpha \rightarrow \gamma)} = \frac{\text{Count}(\alpha \rightarrow \beta)}{\text{Count}(\alpha)}$$

1 Standard PCFGs

- Parameter Estimation
- Problem 1: Ignoring Lexical Information
- Problem 2: Assuming Independence

2 Lexicalized PCFGs

- Lexicalization
- Head Lexicalization
- Parameter Estimation

3 Parsing PCFGs

Reading:

J&M 2nd edition, ch. 14.2–14.6.1 (same as Lecture 18)
NLTK Book, Chapter 8, final section on Weighted Grammar

Parameter Estimation

Here's a parsed corpus of sentences and parse trees.

S1: [S [NP grass] [VP grows]]
S2: [S [NP grass] [VP grows] [AP slowly]]
S3: [S [NP grass] [VP grows] [AP fast]]
S4: [S [NP bananas] [VP grow]]

We can compute PCFG probabilities as follows:

r	Rule	α	$P(r \alpha)$
$r1$	$S \rightarrow NP VP$	S	
$r2$	$S \rightarrow NP VP AP$	S	

Parameter Estimation

Here's a parsed corpus of sentences and parse trees.

S1: [S [NP grass] [VP grows]]
 S2: [S [NP grass] [VP grows] [AP slowly]]
 S3: [S [NP grass] [VP grows] [AP fast]]
 S4: [S [NP bananas] [VP grow]]

We can continue computing PCFG probabilities:

r	Rule	α	$P(r \alpha)$
r_1	$S \rightarrow NP VP$	S	2/4
r_2	$S \rightarrow NP VP AP$	S	2/4
r_3	$NP \rightarrow grass$	NP	
r_4	$NP \rightarrow bananas$	NP	

Parameter Estimation

Here's a parsed corpus of sentences and parse trees.

S1: [S [NP grass] [VP grows]]
 S2: [S [NP grass] [VP grows] [AP slowly]]
 S3: [S [NP grass] [VP grows] [AP fast]]
 S4: [S [NP bananas] [VP grow]]

We can continue computing PCFG probabilities:

r	Rule	α	$P(r \alpha)$
r_1	$S \rightarrow NP VP$	S	2/4
r_2	$S \rightarrow NP VP AP$	S	2/4
r_3	$NP \rightarrow grass$	NP	3/4
r_4	$NP \rightarrow bananas$	NP	1/4
r_5	$VP \rightarrow grows$	VP	
r_6	$VP \rightarrow grow$	VP	

Parameter Estimation

Here's a parsed corpus of sentences and parse trees.

S1: [S [NP grass] [VP grows]]
 S2: [S [NP grass] [VP grows] [AP slowly]]
 S3: [S [NP grass] [VP grows] [AP fast]]
 S4: [S [NP bananas] [VP grow]]

We can continue computing PCFG probabilities:

r	Rule	α	$P(r \alpha)$
r_1	$S \rightarrow NP VP$	S	2/4
r_2	$S \rightarrow NP VP AP$	S	2/4
r_3	$NP \rightarrow grass$	NP	3/4
r_4	$NP \rightarrow bananas$	NP	1/4
r_5	$VP \rightarrow grows$	VP	3/4
r_6	$VP \rightarrow grow$	VP	1/4
r_7	$AP \rightarrow fast$	AP	
r_8	$AP \rightarrow slowly$	AP	

Parameter Estimation

Here's a parsed corpus of sentences and parse trees.

S1: [S [NP grass] [VP grows]]
 S2: [S [NP grass] [VP grows] [AP slowly]]
 S3: [S [NP grass] [VP grows] [AP fast]]
 S4: [S [NP bananas] [VP grow]]

We now have all possible PCFG probabilities:

r	Rule	α	$P(r \alpha)$
r_1	$S \rightarrow NP VP$	S	2/4
r_2	$S \rightarrow NP VP AP$	S	2/4
r_3	$NP \rightarrow grass$	NP	3/4
r_4	$NP \rightarrow bananas$	NP	1/4
r_5	$VP \rightarrow grows$	VP	3/4
r_6	$VP \rightarrow grow$	VP	1/4
r_7	$AP \rightarrow fast$	AP	1/2
r_8	$AP \rightarrow slowly$	AP	1/2

Parameter Estimation

With these parameters (rule probabilities), we can now compute the probabilities of the four sentences S1–S4:

$$\begin{aligned} P(S1) &= P(r1|S)P(r3|NP)P(r5|VP) \\ &= 2/4 \cdot 3/4 \cdot 3/4 = 0.28125 \end{aligned}$$

$$\begin{aligned} P(S2) &= P(r2|S)P(r3|NP)P(r5|VP)P(r7|AP) \\ &= 2/4 \cdot 3/4 \cdot 3/4 \cdot 1/2 = 0.140625 \end{aligned}$$

$$\begin{aligned} P(S3) &= P(r2|S)P(r3|NP)P(r5|VP)P(r7|AP) \\ &= 2/4 \cdot 3/4 \cdot 3/4 \cdot 1/2 = 0.140625 \end{aligned}$$

$$\begin{aligned} P(S4) &= P(r1|S)P(r4|NP)P(r6|VP) \\ &= 2/4 \cdot 1/4 \cdot 1/4 = 0.03125 \end{aligned}$$

Problem 1: Ignoring Lexical Information

Consider the sentences:

- (1) a. They admired the painting of the queen.
b. They put the painting on the wall.

Because rules for rewriting non-terminals ignore word tokens until the very end, let's consider these simply as strings of POS tags:

- (2) a. PRO TV DET N PREP DET N
b. PRO TV DET N PREP DET N

using the lexical rules:

$$\begin{aligned} N &\rightarrow \textit{painting} \mid \textit{queen} \mid \textit{wall} & \text{PRO} &\rightarrow \textit{they} \\ \text{TV} &\rightarrow \textit{admired} \mid \textit{put} & \text{DET} &\rightarrow \textit{the} \\ \text{PREP} &\rightarrow \textit{of} \mid \textit{on} \end{aligned}$$

Problems with Standard PCFGs

While standard PCFGs are useful for a number of applications, they can produce a wrong result when used to choose the correct parse for an ambiguous sentence.

How can that be?

- ① They ignore lexical information until the very end of the analysis, when word classes are rewritten to word tokens.
- ② They make unwarranted independence assumptions.

How can this lead to the wrong choice among possible parses?

Problem 1: Ignoring Lexical Information

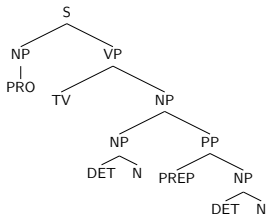
The grammar

$$\begin{aligned} S &\rightarrow NP VP \\ NP &\rightarrow PRO \mid DET N \mid NP PP \\ VP &\rightarrow TV NP \mid TV NP PP \\ PP &\rightarrow PREP NP \end{aligned}$$

provides two possible analyses for the string of POS tags

$$PRO \ TV \ DET \ N \ PREP \ DET \ N$$

Problem 1: Ignoring Lexical Information



Problem 2: Assuming Independence

By definition, a CFG assumes that the expansion of non-terminals is completely **independent**: It doesn't matter

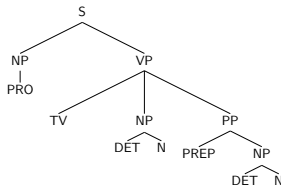
- where a non-terminal is in the analysis;
- what else is (or isn't) in the analysis.

The same assumption holds for standard PCFGs: The probability of a rule is the same, no matter

- where it is applied in the analysis;
- what else is (or isn't) in the analysis.

But this assumption is too simple.

Problem 1: Ignoring Lexical Information



Which do we want for "They admired **the painting of the queen**"?
Which for "They put **the painting on the wall**"?

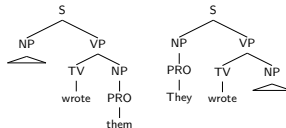
The most appropriate analysis depends, in part, on the **actual words** in the sentence, and not just their POS tags.

Independence Assumption

Consider the rules:

$$\begin{array}{ll} S \rightarrow NP VP & (p_1) \\ VP \rightarrow TV NP & (p_2) \end{array} \quad \begin{array}{ll} NP \rightarrow PRO & (p_3) \\ NP \rightarrow DET NOM & (p_4) \end{array}$$

They assign the same probability to both these trees, because they use the same 5 re-write rules, and probability calculations don't depend on where rules are used.



Independence Assumption

But in one large speech corpus, 91% of 31021 subject NPs are pronouns:

- (3) a. **She's** able to take her baby to work with her.
b. My wife worked until **we** had a family.

while only 34% of 7489 object NPs are pronouns:

- (4) a. Some laws absolutely prohibit **it**.
b. It wasn't clear how NL and Mr. Simmons would respond if Georgia Gulf spurns **them** again.

So the probability of NP → PRO should depend on **where** in the analysis it applies.

Lexicalized PCFGs

In Lecture 13, we saw that each non-terminal in a natural language has a **head** which determines the syntactic properties of the phrase (e.g., which other phrases it can combine with).

Example

Noun Phrase (NP): Noun
Adjective Phrase (AP): Adjective
Verb Phrase (VP): Verb
Prepositional Phrase (PP): Preposition

Key idea: Have each grammar rule specify its lexical head and use it to avoid unwarranted independence assumptions.

How?

Lexicalization

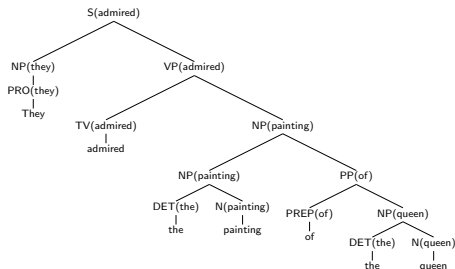
We can lexicalize a PCFG by annotating each non-terminal with its **head word**, starting with the terminals – replacing

VP → V NP PP (p_1)
VP → V NP (p_2)
NP → DET NOM (p_3)
NOM → N PP (p_4)

with rules of the form

VP(put) → V(put) NP(painting) PP(on) (p_1)
VP(admired) → V(admired) NP(painting) PP(on) (p_2)
VP(find) → V(find) NP(painting) PP(on) (p_3)
VP(put) → V(put) NP(painting) (p_4)
VP(admired) → V(admired) NP(painting) (p_5)
NP(painting) → DET(the) NOM(painting) (p_6)

Example



Head Lexicalization

But this would mean an enormous expansion in grammar rules, with no parsed corpus big enough to estimate their probabilities accurately.

Instead we just lexicalize the **head** of phrase:

VP(put)	→	V(put) NP PP	(p1)
VP(admired)	→	V(admired) NP PP	(p2)
VP(find)	→	V(find) NP PP	(p3)
VP(put)	→	V(put) NP	(p4)
VP(admired)	→	V(admired) NP	(p5)
VP(find)	→	V(find) NP	(p6)
NP(painting)	→	DET NOM(painting)	(p7)
NOM(painting)	→	N(painting) PP	(p8)

Such grammars are called **lexicalized PCFGs** or, alternatively, **probabilistic lexicalized CFGs**.

Parameter Estimation

The probabilities of lexicalized PCFGs can be estimated from a syntactically annotated corpus like the Penn TreeBank.

We want to estimate both:

- **rule probabilities** $P(\alpha \rightarrow \beta | \alpha, h(\alpha))$ that condition a rule on its LHS and head, $h(\alpha)$;
- **head probabilities** $P(h(\alpha) | \alpha, h(m(\alpha)))$ that condition the head of a rule on the head of the rule's mother, $h(m(\alpha))$.

These probabilities can be estimated as follows:

$$P(\alpha \rightarrow \beta | \alpha, h(\alpha)) = \frac{\text{Count}(\alpha(h(\alpha)) \rightarrow \beta)}{\text{Count}(\alpha(h(\alpha)))}$$

$$P(h(\alpha) | \alpha, h(m(\alpha))) = \frac{\text{Count}(X(h(m(\alpha))) \rightarrow \dots \alpha(h(\alpha)) \dots)}{\text{Count}(X(h(m(\alpha)))) \rightarrow \dots \alpha \dots)}$$

Parameter Estimation

We can estimate **rule probabilities** from the parsed corpus:

S1: [S [NP grass] [VP grows]]
 S2: [S [NP grass] [VP grows] [AP slowly]]
 S3: [S [NP grass] [VP grows] [AP fast]]
 S4: [S [NP bananas] [VP grow]]

r	Rule	α	$h(\alpha)$	$P(r \alpha, h(\alpha))$
r_1	$S \rightarrow NP VP$	S	grows	
r_2	$S \rightarrow NP VP AP$	S	grows	
r_3	$S \rightarrow NP VP$	S	grow	

where $P(\alpha \rightarrow \beta | \alpha, h(\alpha)) = \frac{\text{Count}(\alpha(h(\alpha)) \rightarrow \beta)}{\text{Count}(\alpha(h(\alpha)))}$

Parameter Estimation

We can estimate **rule probabilities** from the parsed corpus:

S1: [S [NP grass] [VP grows]]
 S2: [S [NP grass] [VP grows] [AP slowly]]
 S3: [S [NP grass] [VP grows] [AP fast]]
 S4: [S [NP bananas] [VP grow]]

r	Rule	α	$h(\alpha)$	$P(r \alpha, h(\alpha))$
r_1	$S \rightarrow NP VP$	S	grows	1/3
r_2	$S \rightarrow NP VP AP$	S	grows	2/3
r_3	$S \rightarrow NP VP$	S	grow	1/1
r_4	$NP \rightarrow grass$	NP	grass	
r_5	$NP \rightarrow bananas$	NP	bananas	

where $P(\alpha \rightarrow \beta | \alpha, h(\alpha)) = \frac{\text{Count}(\alpha(h(\alpha)) \rightarrow \beta)}{\text{Count}(\alpha(h(\alpha)))}$

Parameter Estimation

We can estimate **rule probabilities** from the parsed corpus:

S1: [S [NP grass] [VP grows]]

S2: [S [NP grass] [VP grows] [AP slowly]]

S3: [S [NP grass] [VP grows] [AP fast]]

S4: [S [NP bananas] [VP grow]]

r	Rule	α	$h(\alpha)$	$P(r \alpha, h(\alpha))$
r_1	$S \rightarrow NP VP$	S	grows	1/3
r_2	$S \rightarrow NP VP AP$	S	grows	2/3
r_3	$S \rightarrow NP VP$	S	grow	1/1
r_4	$NP \rightarrow grass$	NP	grass	3/3
r_5	$NP \rightarrow bananas$	NP	bananas	1/1
r_6	$VP \rightarrow grows$	VP	grows	3/3
r_7	$VP \rightarrow grow$	VP	grow	1/1
r_8	$AP \rightarrow fast$	AP	fast	1/1
r_9	$AP \rightarrow slowly$	AP	slowly	1/1

Parameter Estimation

And the head probabilities from this corpus as well:

r	Rule	α	$h(\alpha)$	$h(m(\alpha))$	$P(h(\alpha) h(m(\alpha)))$
r_1	$S \rightarrow NP VP$	S	grows	-	1
r_2	$S \rightarrow NP VP AP$	S	grows	-	1
r_3	$S \rightarrow NP VP$	S	grow	-	1
r_4	$NP \rightarrow grass$	NP	grass	grows	3/3
r_5	$NP \rightarrow bananas$	NP	bananas	grow	1/1
r_6	$VP \rightarrow grows$	VP	grows	grows	3/3
r_7	$VP \rightarrow grow$	VP	grow	grow	1/1
r_8	$AP \rightarrow fast$	AP	fast	grows	1/2
r_9	$AP \rightarrow slowly$	AP	slowly	grows	1/2

$$\text{where } P(h(\alpha)|\alpha, h(m(\alpha))) = \frac{\text{Count}(X(h(m(\alpha))) \rightarrow \dots \alpha(h(\alpha)) \dots)}{\text{Count}(X(h(m(\alpha))) \rightarrow \dots \alpha \dots)}$$

Parameter Estimation

The lexicalized PCFG probability of sentence S with parse tree T is:

$$P(T, S) = \prod_{\alpha \in T} P(\alpha \rightarrow \beta | \alpha, h(\alpha)) P(h(\alpha) | \alpha, h(m(\alpha)))$$

⇒ the product of the **rule probability** and **head probability** at each non-terminal node.

For example, the probability of S2 is:

$$\begin{aligned} P(S_2) &= P(S \rightarrow NP VP AP | S, grows) P(grows | S, -) \\ &\quad P(NP \rightarrow grass | NP, grass) P(grass | NP, grows) \\ &\quad P(VP \rightarrow grows | VP, grows) P(grows | VP, grows) \\ &\quad P(AP \rightarrow slowly | AP, slowly) P(slowly | AP, grows) \\ &= 1/3 \cdot 1 \cdot 3/3 \cdot 3/3 \cdot 3/3 \cdot 3/3 \cdot 1/1 \cdot 1/2 \\ &= 0.1667 \end{aligned}$$

Parsing PCFGs

Different chart parsing algorithms can be used with PCFGs. They can use the probabilities

- to add only the **most likely** analysis to the chart
- to decide in what order to add edges to the chart (ie, in order of **likelihood**)
- to limit which edges get added to the chart (ie, only the **more likely** ones)

A* Parser for PCFGs

In Lecture 18, we saw two analyses of

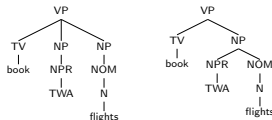
₀ Can ₁ you ₂ book ₃ TWA ₄ flights? ₅

from the PCFG

R1	$S \rightarrow NP VP$	(0.85)	R9	$VP \rightarrow TV NP NP$	(0.05)
R2	$S \rightarrow Aux NP VP$	(0.15)	R10	$VP \rightarrow TV NP$	(0.4)
R3	$NP \rightarrow PRO$	(0.4)	R11	$VP \rightarrow IV$	(0.55)
R4	$NP \rightarrow NOM$	(0.05)	R12	$Aux \rightarrow can$	(0.4)
R5	$NP \rightarrow NPR$	(0.35)	R13	$N \rightarrow flights$	(0.5)
R6	$NP \rightarrow NPR NOM$	(0.2)	R14	$PRO \rightarrow you$	(0.4)
R7	$NOM \rightarrow N$	(0.75)	R15	$TV \rightarrow book$	(0.3)
R8	$NOM \rightarrow N PP$	(0.25)	R16	$NPR \rightarrow TWA$	(0.4)

A* Parser for PCFG

This grammar allows **two** ways of adding **VP** to the chart from 2 to 5 (ie, ₂ book ₃ TWA ₄ flights? ₅).



$$VP: P(R9) P(R5) P(R4) = 0.05 * 0.35 * 0.05 = 0.000875$$

$$VP: P(R10) P(R6) = 0.4 * 0.2 = 0.08$$

So only an edge for the latter would be added to the chart.

Summary

- The rule probabilities of a PCFG can be estimated by counting how often the rules occur in a corpus.
- The usefulness of PCFGs is limited by the lack of lexical information and by strong independence assumptions.
- These limitations can be overcome by lexicalizing the grammars, i.e., by conditioning the rule probabilities on the head word of the rule.
- Several parameter estimation methods are available for lexicalized PCFGs.
- A chart parser can be adapted to use the probabilities in a PCFG.