

Parsing

Probabilistic CFG (PCFG)

Laura Kallmeyer

Heinrich-Heine-Universität Düsseldorf

Summer 2019



Table of contents

- 1 Introduction
- 2 PCFG
- 3 Inside and outside probability
- 4 Parsing

Jurafsky and Martin (2009)

(Some of the slides are due to Wolfgang Maier.)

Data-Driven Parsing

- Linguistic grammars can not only be created manually. Another way to obtain grammars is to interpret the syntactic structures in a treebank as the derivations of a latent grammar and to use an appropriate algorithm for grammar extraction.
- One can also estimate occurrence probabilities for the rules of a grammar. These can be used to determine the best parse, resp. parses of a sentence.
- Furthermore, rule probabilities can serve to speed up parsing.
- Parsing with a probabilistic grammar obtained from a treebank is called **data-driven parsing**.

PCFG (1)

In most cases, probabilistic CFGs are used for data-driven parsing.

PCFG

A **Probabilistic Context-Free Grammar** (PCFG) is a tuple $G_p = (N, T, P, S, p)$ where (N, T, P, S) is a CFG and $p : P \rightarrow [0, 1]^a$ is a function such that for all $A \in N$,

$$\sum_{A \rightarrow \alpha \in P} p(A \rightarrow \alpha) = 1$$

^a $[0, 1]$ denotes $\{i \in \mathbb{R} \mid 0 \leq i \leq 1\}$.

$p(A \rightarrow \alpha)$ is the conditional probability $p(A \rightarrow \alpha \mid A)$

PCFG (2)

PCFG

Start symbol VP

				1	Det \rightarrow the
0.8	VP \rightarrow V NP	1	PP \rightarrow P NP	1	P \rightarrow with
0.2	VP \rightarrow VP PP	0.1	N \rightarrow N PP	0.6	N \rightarrow man
1	NP \rightarrow Det N	1	V \rightarrow sees	0.3	N \rightarrow telescope

PCFG (2)

PCFG

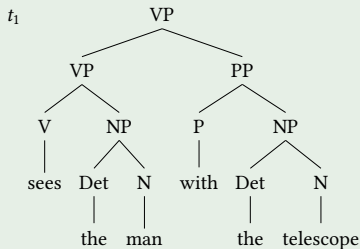
Start symbol VP

				1	Det \rightarrow the
0.8	VP \rightarrow V NP	1	PP \rightarrow P NP	1	P \rightarrow with
0.2	VP \rightarrow VP PP	0.1	N \rightarrow N PP	0.6	N \rightarrow man
1	NP \rightarrow Det N	1	V \rightarrow sees	0.3	N \rightarrow telescope

- Probability of a parse tree: product of the probabilities of the rules used to generate the parse tree.
- Probability of a category A spanning a string w : sum of the probabilities of all parse trees with root label A and yield w .

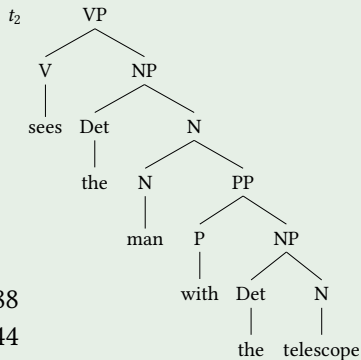
Parse tree probability

0.8 VP → V NP 1 NP → Det N 0.1 N → N PP 1 Det → the 0.6 N → man
 0.2 VP → VP PP 1 PP → P NP 1 V → sees 1 P → with 0.3 N → telescope



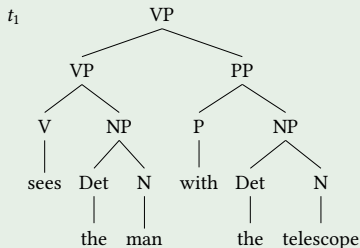
$$P(t_1) = 0.6 \cdot 0.8 \cdot 0.2 \cdot 0.3 = 0.0288$$

$$P(t_2) = 0.6 \cdot 0.8 \cdot 0.1 \cdot 0.3 = 0.0144$$



Parse tree probability

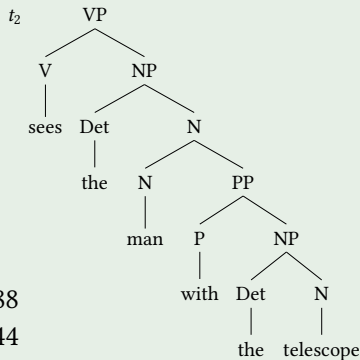
0.8 VP \rightarrow V NP 1 NP \rightarrow Det N 0.1 N \rightarrow N PP 1 Det \rightarrow the 0.6 N \rightarrow man
 0.2 VP \rightarrow VP PP 1 PP \rightarrow P NP 1 V \rightarrow sees 1 P \rightarrow with 0.3 N \rightarrow telescope



$$P(t_1) = 0.6 \cdot 0.8 \cdot 0.2 \cdot 0.3 = 0.0288$$

$$P(t_2) = 0.6 \cdot 0.8 \cdot 0.1 \cdot 0.3 = 0.0144$$

$$p(\text{VP}, \text{sees the man with the telescope}) = 0.0288 + 0.0144$$



Probabilities of leftmost derivations:

Probability of a leftmost derivation

Let $G = (N, T, P, S, p)$ be a PCFG, and let $\alpha, \gamma \in (N \cup T)^*$.

- Let $A \rightarrow \beta \in P$. The probability of a leftmost derivation

$\alpha \xRightarrow{l}^{A \rightarrow \beta} \gamma$ is

$$p(\alpha \xRightarrow{l}^{A \rightarrow \beta} \gamma) = p(A \rightarrow \beta)$$

- Let $A_1 \rightarrow \beta_1, \dots, A_m \rightarrow \beta_m \in P$, $m \in \mathbb{N}$. The probability of a leftmost derivation $\alpha \xRightarrow{l}^{A_1 \rightarrow \beta_1} \dots \xRightarrow{l}^{A_m \rightarrow \beta_m} \gamma$ is

$$p(\alpha \xRightarrow{l}^{A_1 \rightarrow \beta_1} \dots \xRightarrow{l}^{A_m \rightarrow \beta_m} \gamma) = \prod_{i=1}^m p(A_i \rightarrow \beta_i)$$

- The probability of leftmost deriving γ from α , $\alpha \xRightarrow{*}_l \gamma$ is defined as the sum over the probabilities of all leftmost derivations of γ from α :

$$p(\alpha \xRightarrow{*}_l \gamma) = \sum_{i=1}^k \prod_{j=1}^m p(A_j^i \rightarrow \beta_j^i)$$

where $k \in \mathbb{N}$ is the number of leftmost derivations of γ from α and $m \in \mathbb{N}$ is the derivation length of the i th derivation and $A_j^i \rightarrow \beta_j^i$ is the j th derivation step of the i th leftmost derivation.

In the following, the subscript l is omitted assuming that derivations are identified with the corresponding leftmost derivation for probabilities.

Consistent PCFG

A PCFG is **consistent** if the sum of the probabilities of all sentences in the language equals 1.

PCFG (6)

Consistent PCFG

A PCFG is **consistent** if the sum of the probabilities of all sentences in the language equals 1.

Example of an inconsistent PCFG

$$.4 S \rightarrow A \quad .6 S \rightarrow B \quad 1 A \rightarrow a \quad 1 B \rightarrow B$$

Problem: probability mass disappears into infinite derivations.

$$\sum_{w \in L(G)} p(w) = p(a) = 0.4$$

PCFG (6)

Consistent PCFG

A PCFG is **consistent** if the sum of the probabilities of all sentences in the language equals 1.

Example of an inconsistent PCFG

$$.4 S \rightarrow A \quad .6 S \rightarrow B \quad 1 A \rightarrow a \quad 1 B \rightarrow B$$

Problem: probability mass disappears into infinite derivations.

$$\sum_{w \in L(G)} p(w) = p(a) = 0.4$$

PCFGs estimated from treebanks are usually consistent.

Inside and outside probability (1)

Given a PCFG and an input $w = w_1 \dots w_n$, determine the likelihood of w , i.e., compute $\sum_{t' \in T(w)} P(t')$.

We don't want to compute the probability of every parse tree separately and then sum over the results. This is too expensive.

Instead, we adopt a computation with tabulation, in order to share the results for common subtrees.

Inside and outside probability (2)

Idea: We fill a $|N| \times |w| \times |w|$ matrix α where the first dimension is the id of a non-terminal, and the second and third are the start and end indices of a span. $\alpha_{A,i,j}$ gives the probability of deriving $w_i \dots w_j$ from A or, put differently, of a parse tree with root label A and yield $w_i \dots w_j$:

$$\alpha_{A,i,j} = P(A \xRightarrow{*} w_i \dots w_j | A)$$

Inside computation

- 1 for all $1 \leq i \leq |w|$ and $A \in N$:
if $A \rightarrow w_i \in P$, then $\alpha_{A,i,i} = p(A \rightarrow w_i)$, else $\alpha_{A,i,i} = 0$
- 2 for all $1 \leq i < j \leq |w|$ and $A \in N$:
$$\alpha_{A,i,j} = \sum_{A \rightarrow BC \in P} \sum_{k=i}^{j-1} p(A \rightarrow BC) \alpha_{B,i,k} \alpha_{C,k+1,j}$$

We have in particular $\alpha_{S,1,|w|} = P(w)$.

Inside and outside probability (3)

Inside computation

0.3: $S \rightarrow AS$ 0.6: $S \rightarrow AX$ 0.1: $S \rightarrow a$ 1: $X \rightarrow SA$ 1: $A \rightarrow a$
input $w = a^4$

j				
4				(1,A), (0.1,S)
3			(1,A), (0.1,S)	
2		(1,A), (0.1,S)		
1	(1,A), (0.1,S)			
	1	2	3	4 i

Inside and outside probability (3)

Inside computation

0.3: $S \rightarrow AS$ 0.6: $S \rightarrow AX$ 0.1: $S \rightarrow a$ 1: $X \rightarrow SA$ 1: $A \rightarrow a$
 input $w = a^4$

j				
4			$(3 \cdot 10^{-2}, S), (0.1, X)$	$(1, A), (0.1, S)$
3		$(3 \cdot 10^{-2}, S), (0.1, X)$	$(1, A), (0.1, S)$	
2	$(3 \cdot 10^{-2}, S), (0.1, X)$	$(1, A), (0.1, S)$		
1	$(1, A), (0.1, S)$			
	1	2	3	4 i

Inside and outside probability (3)

Inside computation

0.3: $S \rightarrow AS$ 0.6: $S \rightarrow AX$ 0.1: $S \rightarrow a$ 1: $X \rightarrow SA$ 1: $A \rightarrow a$
 input $w = a^4$

j				
4		$(6.9 \cdot 10^{-2}, S),$ $(0.03, X)$	$(3 \cdot 10^{-2}, S), (0.1, X)$	$(1, A), (0.1, S)$
3	$(6.9 \cdot 10^{-2}, S),$ $(0.03, X)$	$(3 \cdot 10^{-2}, S), (0.1, X)$	$(1, A), (0.1, S)$	
2	$(3 \cdot 10^{-2}, S), (0.1, X)$	$(1, A), (0.1, S)$		
1	$(1, A), (0.1, S)$			
	1	2	3	4 i

Inside and outside probability (3)

Inside computation

0.3: $S \rightarrow AS$ 0.6: $S \rightarrow AX$ 0.1: $S \rightarrow a$ 1: $X \rightarrow SA$ 1: $A \rightarrow a$
 input $w = a^4$

j				
4	$(3.87 \cdot 10^{-2}, S),$ $(0.069, X)$	$(6.9 \cdot 10^{-2}, S),$ $(0.03, X)$	$(3 \cdot 10^{-2}, S), (0.1, X)$	$(1, A), (0.1, S)$
3	$(6.9 \cdot 10^{-2}, S),$ $(0.03, X)$	$(3 \cdot 10^{-2}, S), (0.1, X)$	$(1, A), (0.1, S)$	
2	$(3 \cdot 10^{-2}, S), (0.1, X)$	$(1, A), (0.1, S)$		
1	$(1, A), (0.1, S)$			
	1	2	3	4 i

Inside and outside probability (3)

Inside computation

0.3: $S \rightarrow AS$ 0.6: $S \rightarrow AX$ 0.1: $S \rightarrow a$ 1: $X \rightarrow SA$ 1: $A \rightarrow a$
 input $w = a^4$

j				
4	$(3.87 \cdot 10^{-2}, S),$ $(0.069, X)$	$(6.9 \cdot 10^{-2}, S),$ $(0.03, X)$	$(3 \cdot 10^{-2}, S), (0.1, X)$	$(1, A), (0.1, S)$
3	$(6.9 \cdot 10^{-2}, S),$ $(0.03, X)$	$(3 \cdot 10^{-2}, S), (0.1, X)$	$(1, A), (0.1, S)$	
2	$(3 \cdot 10^{-2}, S), (0.1, X)$	$(1, A), (0.1, S)$		
1	$(1, A), (0.1, S)$			
	1	2	3	4 i

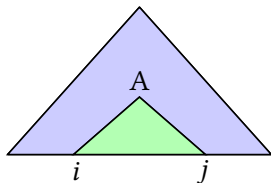
$$P(aaaa) = \alpha_{S,1,4} = 0.0387$$

Inside and outside probability (4)

We can also compute the outside probability of a given non-terminal A with a span from i to j .

Inside: Sum over all possibilities for the tree below A (spanning from i to j).

Outside: Sum over all possibilities for the part of the parse tree outside the tree below A , i.e., over all possibilities to complete a A, i, j tree into a parse tree for the entire sentence.



Outside probability $\beta_{A,i,j}$

Inside probability $\alpha_{A,i,j}$

Inside and outside probability (5)

We fill a $|N| \times |w| \times |w|$ matrix β such that $\beta_{A,i,j}$ gives the probability of deriving $w_1 \dots w_{i-1}Aw_{j+1} \dots w_{|w|}$ from S or, put differently, of deriving a tree with root label S and yield $w_1 \dots w_{i-1}Aw_{j+1} \dots w_{|w|}$:

$$\beta_{A,i,j} = P(S \xRightarrow{*} w_1 \dots w_{i-1}Aw_{j+1} \dots w_{|w|} | S)$$

We need the inside probabilities in order to compute the outside probabilities.

Outside computation

① $\beta_{S,1,|w|} = 1$ and $\beta_{A,1,|w|} = 0$ for all $A \neq S$

② for all $1 \leq i \leq j \leq |w|$ and $A \in N$:

$$\beta_{A,i,j} = \sum_{B \rightarrow AC \in P} \sum_{k=j+1}^n p(B \rightarrow AC) \beta_{B,i,k} \alpha_{C,j+1,k} \\ + \sum_{B \rightarrow CA \in P} \sum_{k=1}^{i-1} p(B \rightarrow CA) \beta_{B,k,j} \alpha_{C,k,i-1}$$

Inside and outside probability (6)

Outside computation

0.3: $S \rightarrow AS$ 0.6: $S \rightarrow AX$ 0.1: $S \rightarrow a$ 1: $X \rightarrow SA$ 1: $A \rightarrow a$
input $w = a^3$

j				
3	(1,S), (0,A), (0,X)			
2				
1				
	1	2	3	i

Inside and outside probability (6)

Outside computation

0.3: $S \rightarrow AS$ 0.6: $S \rightarrow AX$ 0.1: $S \rightarrow a$ 1: $X \rightarrow SA$ 1: $A \rightarrow a$
input $w = a^3$

j				
3	(1,S), (0,A), (0,X)	(0.3,S), (0,A), (0.6,X)		
2	(0,S), (0,X), (0.03,A)			
1				
	1	2	3	i

Inside and outside probability (6)

Outside computation

0.3: $S \rightarrow AS$ 0.6: $S \rightarrow AX$ 0.1: $S \rightarrow a$ 1: $X \rightarrow SA$ 1: $A \rightarrow a$
 input $w = a^3$

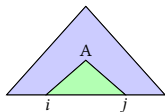
j				
3	(1,S), (0,A), (0,X)	(0.3,S), (0,A), (0.6,X)	($9 \cdot 10^{-2}$,S), (0.18,X), ($3 \cdot 10^{-2}$,A)	
2	(0,S), (0,X), (0.03,A)	(0.6,S), (0,X), ($8.99 \cdot 10^{-3}$,A)		
1	(0,S), (0,X), ($6.9 \cdot 10^{-2}$,A)			
	1	2	3	i

Inside and outside probability (7)

The following holds:

- 1 The probability of a parse tree for w with a node labeled A that spans $w_i \dots w_j$ is

$$P(S \xRightarrow{*} w_1 \dots w_{i-1} A w_{j+1} \dots w_n \xRightarrow{*} w_1 \dots w_n) = \alpha_{A,i,j} \beta_{A,i,j}$$



- 2 In particular: $P(w) = \alpha_{S,1,|w|}$

Parsing (1)

- In PCFG parsing, we want to compute the most probable parse tree (= most probable (leftmost) derivation) given an input sentence w , also called the **Viterbi** parse.
- This means that we are disambiguating: Among several readings, we search for the best.
- Sometimes, the k best are searched for ($k > 1$).
- During parsing, we must make sure that updates on probabilities (because a better derivation has been found for a non-terminal) do not require updates on other parts of the chart. \Rightarrow the order should be such that an item is used within a derivation only when its final probability is reached.

Parsing (2)

We can extend the symbolic CYK parser to a probabilistic one. Instead of summing over all derivations (as in the computation of the inside probability), we keep the best one (\Rightarrow **Viterbi algorithm**).

Assume a three-dimensional chart C (non-terminal, start index, length).

```
 $C_{A,i,l} := 0$  for all  $A, i, l$   
 $C_{A,i,1} := p$  if  $p: A \rightarrow w_i \in P$  scan  
for all  $l \in [1..n]$ :  
  for all  $i \in [1..n-l+1]$ :  
    for every  $p: A \rightarrow B \ C$ :  
      for every  $l_1 \in [1..l-1]$ :  
         $C_{A,i,l} = \max\{C_{A,i,l}, p \cdot C_{B,i,l_1} \cdot C_{C,i+l_1,l-l_1}\}$  complete
```

Parsing (3)

We extend this to a parser.

- The parser can also deal with unary productions $A \rightarrow B$.
- Every chart field has three components, the probability, the rule that has been used and, if the rule is binary, the length l_1 of the first righthand side element.
- We assume that the grammar does not contain any loops $A \xRightarrow{+} A$.

Parsing (4)

```
 $C_{A,i,1} = \langle p, A \rightarrow w_i, - \rangle$  if  $p: A \rightarrow w_i \in P$  scan  
for all  $l \in [1..n]$  and for all  $i \in [1..n-l]$ :  
  for all  $p: A \rightarrow B C$  and for all  $l_1 \in [1..l-1]$ :  
    for all  $l_1 \in [1..l-1]$ :  
      if  $C_{B,i,l_1} \neq \emptyset$  and  $C_{C,i+l_1,l-l_1} \neq \emptyset$  then:  
         $p_{new} = p \cdot C_{B,i,l_1}[1] \cdot C_{C,i+l_1,l-l_1}[1]$   
        if  $C_{A,i,l} == \emptyset$  or  $C_{A,i,l}[1] < p_{new}$  then:  
           $C_{A,i,l} = \langle p_{new}, A \rightarrow BC, l_1 \rangle$  binary complete  
    repeat until  $C$  does not change any more:  
      for every  $p: A \rightarrow B$ :  
        if  $C_{B,i,l} \neq \emptyset$  then:  
           $p_{new} = p \cdot C_{B,i,l}[1]$   
          if  $C_{A,i,l} == \emptyset$  or  $C_{A,i,l}[1] < p_{new}$  then:  
             $C_{A,i,l} = \langle p_{new}, A \rightarrow B, - \rangle$  unary complete  
return build_tree( $S, 1, n$ )
```

Parsing (5)

Example

- .1 VP \rightarrow VP NP 1 NP \rightarrow Det N .3 V \rightarrow eats
.6 VP \rightarrow V NP .3 V \rightarrow sees 1 Det \rightarrow this
.3 VP \rightarrow V .4 V \rightarrow comes .5 N \rightarrow morning
.5 N \rightarrow apple

Start symbol VP, input $w = \text{eats this morning}$

<i>l</i>				
3				
2				
1				
	1	2	3	<i>i</i>

Parsing (5)

Example

- .1 VP \rightarrow VP NP 1 NP \rightarrow Det N .3 V \rightarrow eats
.6 VP \rightarrow V NP .3 V \rightarrow sees 1 Det \rightarrow this
.3 VP \rightarrow V .4 V \rightarrow comes .5 N \rightarrow morning
.5 N \rightarrow apple

Start symbol VP, input $w = \text{eats this morning}$

<i>l</i>			
3			
2			
1	.3, V \rightarrow eats	1, Det \rightarrow this	.5, N \rightarrow morning
	1	2	3
			<i>i</i>

Parsing (5)

Example

- .1 VP \rightarrow VP NP 1 NP \rightarrow Det N .3 V \rightarrow eats
 .6 VP \rightarrow V NP .3 V \rightarrow sees 1 Det \rightarrow this
 .3 VP \rightarrow V .4 V \rightarrow comes .5 N \rightarrow morning
 .5 N \rightarrow apple

Start symbol VP, input $w = \text{eats this morning}$

l			
3			
2			
	.09, VP \rightarrow V		
1	.3, V \rightarrow eats	1, Det \rightarrow this	.5, N \rightarrow morning
	1	2	3
			i

Parsing (5)

Example

- .1 VP \rightarrow VP NP 1 NP \rightarrow Det N .3 V \rightarrow eats
 .6 VP \rightarrow V NP .3 V \rightarrow sees 1 Det \rightarrow this
 .3 VP \rightarrow V .4 V \rightarrow comes .5 N \rightarrow morning
 .5 N \rightarrow apple

Start symbol VP, input $w = \text{eats this morning}$

l			
3			
2		.5, NP \rightarrow Det N, 1	
	.09, VP \rightarrow V		
1	.3, V \rightarrow eats	1, Det \rightarrow this	.5, N \rightarrow morning
	1	2	3
			i

Parsing (5)

Example

- .1 VP \rightarrow VP NP 1 NP \rightarrow Det N .3 V \rightarrow eats
 .6 VP \rightarrow V NP .3 V \rightarrow sees 1 Det \rightarrow this
 .3 VP \rightarrow V .4 V \rightarrow comes .5 N \rightarrow morning
 .5 N \rightarrow apple

Start symbol VP, input $w = \text{eats this morning}$

l			
3	.0045, VP \rightarrow VP NP, 1		
2		.5, NP \rightarrow Det N, 1	
	.09, VP \rightarrow V		
1	.3, V \rightarrow eats	1, Det \rightarrow this	.5, N \rightarrow morning
	1	2	3
			i

Parsing (5)

Example

- .1 VP \rightarrow VP NP 1 NP \rightarrow Det N .3 V \rightarrow eats
 .6 VP \rightarrow V NP .3 V \rightarrow sees 1 Det \rightarrow this
 .3 VP \rightarrow V .4 V \rightarrow comes .5 N \rightarrow morning
 .5 N \rightarrow apple

Start symbol VP, input $w = \text{eats this morning}$

l			
3	.09, VP \rightarrow V NP, 1		
2		.5, NP \rightarrow Det N, 1	
	.09, VP \rightarrow V		
1	.3, V \rightarrow eats	1, Det \rightarrow this	.5, N \rightarrow morning
	1	2	3

(The analysis of the VP gets revised since a better parse tree has been found.)

Jurafsky, D. and Martin, J. H. (2009). *Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall Series in Artificial Intelligence. Pearson Education International, second edition edition.