Parsing with PCFGs

Joakim Nivre

Uppsala University
Department of Linguistics and Philology
joakim.nivre@lingfil.uu.se
Probabilistic Context-Free Grammar (PCFG)

1. Grammar Formalism
2. Parsing Model
3. Parsing Algorithms
4. Learning with a Treebank
5. Learning without a Treebank
$G = (N, \Sigma, R, S, Q)$

- $N$ is a finite (non-terminal) alphabet
- $\Sigma$ is a finite (terminal) alphabet
- $R$ is a finite set of rules $A \rightarrow \alpha$ ($A \in N$, $\alpha \in (\Sigma \cup N)^*$)
- $S \in N$ is the start symbol
- $Q$ is function from $R$ to the real numbers in the interval $[0, 1]$
S → NP VP PU 1.00
VP → VP PP 0.33
VP → VBD NP 0.67
NP → NP PP 0.14
NP → JJ NN 0.57
NP → JJ NNS 0.29
PP → IN NP 1.00
PU → . 1.00
JJ → Economic 0.33
JJ → little 0.33
JJ → financial 0.33
NN → news 0.50
NN → effect 0.50
NNS → markets 1.00
VBD → had 1.00
IN → on 1.00
\[ L(G) = \{ x \in \Sigma^* | S \Rightarrow^* x \} \]

\[ T(G) = \text{set of parse trees for } x \in L(G) \]

For parse tree \( y \in T(G) \):

- yield\((y)\) = terminal string associated with \( y \)
- count\((i, y)\) = number of times the \( r_i \in R \) is used to derive \( y \)
- \( \text{lhs}(i) \) = nonterminal symbol in the left-hand side of \( r_i \)
- \( Q(i) = q_i = \text{probability of } r_i \)
Probability $P(y)$ of a parse tree $y \in T(G)$:

$$P(y) = \prod_{i=1}^{\mid R \mid} q_i^{\text{count}(i, y)}$$

Probability $P(x, y)$ of a string $x$ and parse tree $y$:

$$P(x, y) = \begin{cases} P(y) & \text{if } \text{yield}(y) = x \\ 0 & \text{otherwise} \end{cases}$$

The probability $P(x)$ of a string $x \in L(G)$:

$$P(x) = \sum_{y \in T(G) : \text{yield}(y) = x} P(y)$$
A PCFG is **proper** iff for every nonterminal $A \in N$

$$\sum_{r_i \in R: \text{lhs}(i) = A} q_i = 1$$

A PCFG is **consistent** iff

$$\sum_{y \in T(G)} P(y) = 1$$
1. \( \mathcal{X} = \Sigma^* \)
2. \( \mathcal{Y} = R^* \) [parse trees = leftmost derivations]
3. \( \text{GEN}(x) = \{ y \in T(G) \mid \text{yield}(y) = x \} \)
4. \( \text{EVAL}(y) = P(y) = \prod_{i=1}^{R} q_{\text{count}(i,y)} \)

**NB:** Joint probability is proportional to conditional probability:

\[
P(y|x) = \frac{P(x, y)}{\sum_{y' \in \text{GEN}(x)} P(y')}\]
S → NP VP PU 1.00
VP → VP PP 0.33
VP → VBD NP 0.67
NP → NP PP 0.14
NP → JJ NN 0.57
NP → JJ NNS 0.29
PP → IN NP 1.00
PU → . 1.00
JJ → Economic 0.33
JJ → little 0.33
JJ → financial 0.33
NN → news 0.50
NN → effect 0.50
NNS → markets 1.00
VBD → had 1.00
IN → on 1.00
Parsing (decoding) problem for PCFG $G$ and input $x$:
- Compute $\text{GEN}(x)$
- Compute $\text{EVAL}(y)$ for $y \in \text{GEN}(x)$

Standard algorithms for CFG can be adapted to PCFG:
- CKY
- Earley

Viterbi parsing:

$$\arg\max_{y \in \text{GEN}(x)} \text{EVAL}(y)$$
Fencepost positions
PARSE(G, x)

for j from 1 to n do
    for all A : A → a ∈ R and a = j − 1 w_j
        C[j − 1, j, A] := Q(A → a)

for j from 2 to n do
    for i from j − 2 downto 0 do
        for k from i + 1 to j − 1 do
            for all A : A → BC ∈ R and C[i, k, B] > 0 and C[k, j, C] > 0
                if (C[i, j, A] < Q(A → BC) · C[i, k, B] · C[k, j, C]) then
                    C[i, j, A] := Q(A → BC) · C[i, k, B] · C[k, j, C]
                    B[i, j, A] := {k, B, C}

return BUILD-TREE(B[0, n, S]), C[0, n, S]
Training set:
- Treebank $Y = \{y^1, \ldots, y^m\}$

Extract grammar $G = (N, \Sigma, R, S)$:
- $N =$ the set of all nonterminals occurring in some $y_i \in Y$
- $\Sigma =$ the set of all terminals occurring in some $y_i \in Y$
- $R =$ the set of all rules needed to derive some $y_i \in Y$
- $S =$ the nonterminal at the root of every $y_i \in Y$

Estimate $Q$ using relative frequencies (MLE):

$$q_i = \frac{\sum_{j=1}^{m} \text{count}(i, y^j)}{\sum_{j=1}^{m} \sum_{r_k \in R: \text{lhs}(r_k) = \text{lhs}(r_i)} \text{count}(k, y^j)}$$
S → NP VP PU 1.00
VP → VP PP 0.33
VP → VBD NP 0.67
NP → NP PP 0.14
NP → JJ NN 0.57
NP → JJ NNS 0.29
PP → IN NP 1.00
PU → . 1.00
JJ → Economic 0.33
JJ → little 0.33
JJ → financial 0.33
NN → news 0.50
NN → effect 0.50
NNS → markets 1.00
VBD → had 1.00
IN → on 1.00

Economic news had little effect on financial markets .

Economic news had little effect on financial markets .
Training set:
- Corpus $X = \{x^1, \ldots, x^m\}$
- Grammar $G = (N, \Sigma, R, S)$

Estimate $Q$ using expectation-maximization (EM):
1. Guess a probability $q_i$ for each rule $r_i \in R$
2. Repeat until convergence:
   2.1 E-step: Compute the expected count $f(r_i)$ of each rule $r_i \in R$:
   $$ f(r_i) = \sum_{j=1}^{m} \sum_{y \in \text{GEN}(x^j)} P(y | x^j, Q) \cdot \text{count}(i, y) $$
   2.2 M-step: Reestimate the probability $q_i$ of each rule $r_i$ to maximize the marginal likelihood given expected counts:
   $$ q_i = \frac{f(r_i)}{\sum_{r_j \in R: \text{lhs}(r_j) = \text{lhs}(r_i)} f(r_j)} $$