Lecture 16: PCFG Parsing (updated)

Julia Hockenmaier
juliahmr@illinois.edu
3324 Siebel Center
Overview
Where we’re at

Previous lecture:
Standard **CKY** (for non-probabilistic CFGs)
The CKY algorithm finds all possible parse trees $\tau$ for a sentence $S = w^{(1)} \ldots w^{(n)}$ under a CFG $G$ in Chomsky Normal Form.

Today’s lecture:
**Probabilistic Context-Free Grammars (PCFGs)**
− CFGs in which each rule is associated with a probability

**CKY for PCFGs (Viterbi):**
− CKY for PCFGs finds the most likely parse tree
  $\tau^* = \text{argmax } P(\tau | S)$ for the sentence $S$ under a PCFG.

**Shortcomings of PCFGs** (and ways to overcome them)

**Penn Treebank Parsing**

**Evaluating PCFG parsers**
CKY: filling the chart
CKY: filling one cell

chart[2][6]:

w₁ W₂ W₃ W₄ W₅ W₆ W₇
CKY for standard CFGs

CKY is a bottom-up chart parsing algorithm that finds all possible parse trees \( \tau \) for a sentence \( S = w^{(1)} \ldots w^{(n)} \) under a CFG \( G \) in Chomsky Normal Form (CNF).

- **CNF**: \( G \) has two types of rules: \( X \rightarrow Y \ Z \) and \( X \rightarrow w \)  
  (\( X, Y, Z \) are nonterminals, \( w \) is a terminal)

- CKY is a **dynamic programming** algorithm

- The **parse chart** is an \( n \times n \) upper triangular matrix:  
  Each cell \( \text{chart}[i][j] \) \((i \leq j)\) stores all subtrees for \( w^{(i)} \ldots w^{(j)} \)

- Each cell \( \text{chart}[i][j] \) has at most one entry for each nonterminal \( X \) (and pairs of backpointers) to each pair of \( (Y, Z) \) entry in cells \( \text{chart}[i][k] \ \text{chart}[k+1][j] \) from which an \( X \) can be formed

- Time Complexity: \( O(n^3 \ |G|) \)
Grammars are ambiguous

A grammar might generate multiple trees for a sentence:

What’s the most likely parse $\tau$ for sentence $S$ ?

We need a model of $P(\tau | S)$
Computing $P(\tau \mid S)$

Using Bayes’ Rule:

$$\arg \max_{\tau} P(\tau \mid S) = \arg \max_{\tau} \frac{P(\tau, S)}{P(S)}$$

$$= \arg \max_{\tau} P(\tau, S)$$

$$= \arg \max_{\tau} P(\tau) \text{ if } S = \text{yield}(\tau)$$

The **yield of a tree** is the string of terminal symbols that can be read off the leaf nodes

$\text{yield}(\text{eat with tuna}, \text{sushi}) = \text{eat sushi with tuna}$
Computing $P(\tau)$

$T$ is the (infinite) set of all trees in the language:

$$L = \{ s \in \Sigma^* | \exists \tau \in T : \text{yield}(\tau) = s \}$$

We need to define $P(\tau)$ such that:

$$\forall \tau \in T : 0 \leq P(\tau) \leq 1$$

$$\sum_{\tau \in T} P(\tau) = 1$$

The set $T$ is generated by a context-free grammar

$$S \rightarrow \text{NP VP} \quad \text{VP} \rightarrow \text{Verb NP} \quad \text{NP} \rightarrow \text{Det Noun}$$

$$S \rightarrow S \text{ conj } S \quad \text{VP} \rightarrow \text{VP PP} \quad \text{NP} \rightarrow \text{NP PP}$$

$$S \rightarrow ...... \quad \text{VP} \rightarrow ...... \quad \text{NP} \rightarrow ......$$
Proabilistic Context-Free Grammars

For every nonterminal \( X \), define a probability distribution \( P(X \rightarrow \alpha | X) \) over all rules with the same LHS symbol \( X \):

<table>
<thead>
<tr>
<th>Rule</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S \rightarrow NP \ VP )</td>
<td>0.8</td>
</tr>
<tr>
<td>( S \rightarrow S \ conj \ S )</td>
<td>0.2</td>
</tr>
<tr>
<td>( NP \rightarrow Noun )</td>
<td>0.2</td>
</tr>
<tr>
<td>( NP \rightarrow Det \ Noun )</td>
<td>0.4</td>
</tr>
<tr>
<td>( NP \rightarrow NP \ PP )</td>
<td>0.2</td>
</tr>
<tr>
<td>( NP \rightarrow NP \ conj \ NP )</td>
<td>0.2</td>
</tr>
<tr>
<td>( VP \rightarrow Verb )</td>
<td>0.4</td>
</tr>
<tr>
<td>( VP \rightarrow Verb \ NP )</td>
<td>0.3</td>
</tr>
<tr>
<td>( VP \rightarrow Verb \ NP \ NP )</td>
<td>0.1</td>
</tr>
<tr>
<td>( VP \rightarrow VP \ PP )</td>
<td>0.2</td>
</tr>
<tr>
<td>( PP \rightarrow P \ NP )</td>
<td>1.0</td>
</tr>
</tbody>
</table>
Computing $P(\tau)$ with a PCFG

The probability of a tree $\tau$ is the product of the probabilities of all its rules:

$$P(\tau) = 0.8 \times 0.3 \times 0.2 \times 1.0 \times 0.2^3 \approx 0.00384$$
Learning the parameters of a PCFG

If we have a treebank (a corpus in which each sentence is associated with a parse tree), we can just count the number of times each rule appears, e.g.:

\[
S \rightarrow NP \ VP . \quad (\text{count} = 1000)
\]

\[
S \rightarrow S \ \text{conj} \ S . \quad (\text{count} = 220)
\]

\[
PP \rightarrow \text{IN} \ NP \quad (\text{count} = 700)
\]

and then we divide the count (observed frequency) of each rule \( X \rightarrow Y \ Z \) by the sum of the frequencies of all rules with the same LHS \( X \) to turn these counts into probabilities:

\[
S \rightarrow NP \ VP . \quad (p = 1000/1220)
\]

\[
S \rightarrow S \ \text{conj} \ S . \quad (p = 220/1220)
\]

\[
PP \rightarrow \text{IN} \ NP \quad (p = 700/700)
\]
More on probabilities:

Computing $P(s)$:
If $P(\tau)$ is the probability of a tree $\tau$, the probability of a sentence $s$ is the sum of the probabilities of all its parse trees:

$$P(s) = \sum_{\tau: \text{yield}(\tau) = s} P(\tau)$$

How do we know that $P(L) = \sum_{\tau} P(\tau) = 1$?
If we have learned the PCFG from a corpus via MLE, this is guaranteed to be the case.

But if we set the probabilities by hand, we could run into trouble:
In this PCFG, the probability mass of all finite trees is less than 1:

$$S \rightarrow SS \ (0.9) \quad S \rightarrow w \ (0.1)$$

$$P(L) = P(“w”) + P(“ww”) + P(“w[ww]”) + P(“[ww]w”) + \ldots
\approx .1 + .009 + 0.00081 + 0.00081 + \ldots \approx 1$$
PCFG Decoding: CKY with Viterbi
How do we handle flat rules?

Binarize each flat rule by adding a unique dummy nonterminal \((\text{ConjS})\), and setting the probability of the new rule with the dummy nonterminal on the LHS to 1.
How do we handle flat rules?

S → NP VP  0.8
S → S conj S  0.2
NP → Noun  0.2
NP → Det Noun  0.4
NP → NP PP  0.2
NP → NP conj NP  0.2
VP → Verb  0.3
VP → Verb NP  0.3
VP → Verb NP NP  0.1
VP → VP PP  0.3
PP → PP NP  1.0
Prep → P  1.0
Noun → N  1.0
Verb → V  1.0

ConjS → conj S  1.0
ConjNP → conj NP  1.0
NPNP → NP NP  1.0
Probabilistic CKY: Viterbi

Like standard CKY, but with probabilities.
Finding the most likely tree is similar to Viterbi for HMMs:

**Initialization:**
- [optional] Every chart entry that corresponds to a **terminal** (entry \(w\) in \(\text{cell}[i][i]\)) has a Viterbi probability \(P_{\text{VIT}}(w[i][i]) = 1\) (*)
- Every entry for a **non-terminal** \(X\) in \(\text{cell}[i][i]\) has Viterbi probability \(P_{\text{VIT}}(X[i][i]) = P(X \rightarrow w \mid X)\) [and a single backpointer to \(w[i][i]\) (*)]

**Recurrence:** For every entry that corresponds to a **non-terminal** \(X\) in \(\text{cell}[i][j]\), keep only the highest-scoring pair of backpointers to any pair of children (\(Y\) in \(\text{cell}[i][k]\) and \(Z\) in \(\text{cell}[k+1][j]\)): \(P_{\text{VIT}}(X[i][j]) = \arg\max_{Y,Z,k} P_{\text{VIT}}(Y[i][k]) \times P_{\text{VIT}}(Z[k+1][j]) \times P(X \rightarrow Y Z \mid X)\)

**Final step:** Return the Viterbi parse for the start symbol \(S\) in the top \(\text{cell}[1][n]\).

*this is unnecessary for simple PCFGs, but can be helpful for more complex probability models
Probabilistic CKY

Input: POS-tagged sentence
John_N eats_V pie_N with_P cream_N

S → NP VP 0.8
S → S ConjS 0.2
NP → Noun 0.2
NP → Det Noun 0.4
NP → NP PP 0.2
NP → NP ConjNP 0.2
VP → Verb 0.3
VP → Verb NP 0.3
VP → Verb NPNP 0.1
VP → VP PP 0.3
PP → PP NP 1.0
Prep → P 1.0
Noun → N 1.0
Verb → V 1.0
ConjS → conj S 1.0
ConjNP → conj NP 1.0
NPNP → NP NP 1.0
Probabilistic CKY

Input: POS-tagged sentence
John_N eats_V pie_N with_P cream_N

<table>
<thead>
<tr>
<th></th>
<th>John</th>
<th>eats</th>
<th>pie</th>
<th>with</th>
<th>cream</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>John</td>
</tr>
<tr>
<td>eats</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>eats</td>
</tr>
<tr>
<td>pie</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pie</td>
</tr>
<tr>
<td>with</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>with</td>
</tr>
<tr>
<td>cream</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>cream</td>
</tr>
</tbody>
</table>

S  → NP VP     0.8  
S  → S ConjS   0.2  
NP → Noun      0.2  
NP → Det Noun  0.4  
NP → NP PP     0.2  
NP → NP ConjNP 0.2  
VP → Verb      0.3  
VP → Verb NP   0.3  
VP → Verb NPNP 0.1  
PP → PP NP     1.0  
Prep → P       1.0  
Noun → N       1.0  
Verb → V       1.0  
ConjS → conj S  1.0  
ConjNP → conj NP 1.0 
NPNP → NP NP  1.0
# Probabilistic CKY

## Input: POS-tagged sentence

John_N eats_V pie_N with_P cream_N

<table>
<thead>
<tr>
<th></th>
<th>John</th>
<th>eats</th>
<th>pie</th>
<th>with</th>
<th>cream</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>NP</th>
<th>VP</th>
<th>PP</th>
<th>ConjNP</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>NP</td>
<td>VP</td>
<td>0.8</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>ConjS</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td>Noun</td>
<td>0.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td>Det Noun</td>
<td>0.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td>NP</td>
<td>PP</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td>NP</td>
<td>ConjNP</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>VP</td>
<td>Verb</td>
<td>0.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VP</td>
<td>Verb</td>
<td>NP</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>VP</td>
<td>Verb</td>
<td>NPNP</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>PP</td>
<td>PP</td>
<td>NP</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>Prep</td>
<td>P</td>
<td>1.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>NPNP</th>
<th>ConjS</th>
<th>ConjNP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun</td>
<td>N</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>Verb</td>
<td>V</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>ConjS</td>
<td>conj S</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>ConjNP</td>
<td>conj NP</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>NPNP</td>
<td>NP</td>
<td>NP</td>
<td>1.0</td>
</tr>
</tbody>
</table>
Probabilistic CKY

Input: POS-tagged sentence
John_N eats_V pie_N with_P cream_N

<table>
<thead>
<tr>
<th>John</th>
<th>eats</th>
<th>pie</th>
<th>with</th>
<th>cream</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun</td>
<td>NP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Verb     | VP     |       |       |       |
| 1.0      | 0.3    |       |       |       |

<table>
<thead>
<tr>
<th>S</th>
<th>NP VP</th>
<th>0.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>S ConjS</td>
<td>0.2</td>
</tr>
<tr>
<td>NP</td>
<td>Noun</td>
<td>0.2</td>
</tr>
<tr>
<td>NP</td>
<td>Det Noun</td>
<td>0.4</td>
</tr>
<tr>
<td>NP</td>
<td>NP PP</td>
<td>0.2</td>
</tr>
<tr>
<td>NP</td>
<td>NP ConjNP</td>
<td>0.2</td>
</tr>
<tr>
<td>VP</td>
<td>Verb</td>
<td>0.3</td>
</tr>
<tr>
<td>VP</td>
<td>Verb NP</td>
<td>0.3</td>
</tr>
<tr>
<td>VP</td>
<td>Verb NPNP</td>
<td>0.1</td>
</tr>
<tr>
<td>PP</td>
<td>PP NP</td>
<td>1.0</td>
</tr>
<tr>
<td>Prep</td>
<td>P</td>
<td>1.0</td>
</tr>
<tr>
<td>Noun</td>
<td>N</td>
<td>1.0</td>
</tr>
<tr>
<td>Verb</td>
<td>V</td>
<td>1.0</td>
</tr>
<tr>
<td>ConjS</td>
<td>conj S</td>
<td>1.0</td>
</tr>
<tr>
<td>ConjNP</td>
<td>conj NP</td>
<td>1.0</td>
</tr>
<tr>
<td>NPNP</td>
<td>NP NP</td>
<td>1.0</td>
</tr>
</tbody>
</table>
## Probabilistic CKY

### Input: POS-tagged sentence

\( \text{John}_N \text{ eats}_V \text{ pie}_N \text{ with}_P \text{ cream}_N \)

<table>
<thead>
<tr>
<th></th>
<th>John</th>
<th>eats</th>
<th>pie</th>
<th>with</th>
<th>cream</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun</td>
<td>NP</td>
<td>1.0</td>
<td>0.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Noun</td>
<td>NP</td>
<td>1.0</td>
<td>0.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Verb</td>
<td>VP</td>
<td>1.0</td>
<td>0.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Noun</td>
<td>NP</td>
<td>1.0</td>
<td>0.2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Rules

\[
\begin{align*}
    S & \rightarrow \text{NP} \ \text{VP} \quad 0.8 \\
    S & \rightarrow S \ \text{ConjS} \quad 0.2 \\
    \text{NP} & \rightarrow \text{Noun} \quad 0.2 \\
    \text{NP} & \rightarrow \text{Det} \ \text{Noun} \quad 0.4 \\
    \text{NP} & \rightarrow \text{NP} \ \text{PP} \quad 0.2 \\
    \text{NP} & \rightarrow \text{NP} \ \text{ConjNP} \quad 0.2 \\
    \text{VP} & \rightarrow \text{Verb} \quad 0.3 \\
    \text{VP} & \rightarrow \text{Verb} \ \text{NP} \quad 0.3 \\
    \text{VP} & \rightarrow \text{Verb} \ \text{NPNP} \quad 0.1 \\
    \text{VP} & \rightarrow \text{VP} \ \text{PP} \quad 0.3 \\
    \text{PP} & \rightarrow \text{PP} \ \text{NP} \quad 1.0 \\
    \text{Prep} & \rightarrow \text{P} \quad 1.0 \\
    \text{Noun} & \rightarrow \text{N} \quad 1.0 \\
    \text{Verb} & \rightarrow \text{V} \quad 1.0 \\
    \text{ConjS} & \rightarrow \text{conj} \ \text{S} \quad 1.0 \\
    \text{ConjNP} & \rightarrow \text{conj} \ \text{NP} \quad 1.0 \\
    \text{NPNP} & \rightarrow \text{NP} \ \text{NP} \quad 1.0
\end{align*}
\]
## Probabilistic CKY

**Input: POS-tagged sentence**

\[\text{John}_N \text{ eats}_V \text{ pie}_N \text{ with}_P \text{ cream}_N\]

<table>
<thead>
<tr>
<th></th>
<th>John</th>
<th>eats</th>
<th>pie</th>
<th>with</th>
<th>cream</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun</td>
<td>NP</td>
<td>1.0</td>
<td>0.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Verb</td>
<td>VP</td>
<td>1.0</td>
<td>0.3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| S          | NP VP          | 0.8  |
| S          | S ConjS        | 0.2  |
| NP         | Noun           | 0.2  |
| NP         | Det Noun       | 0.4  |
| NP         | NP PP          | 0.2  |
| NP         | NP ConjNP      | 0.2  |
| VP         | Verb           | 0.3  |
| VP         | Verb NP        | 0.3  |
| VP         | Verb NPNP      | 0.1  |
| PP         | PP NP          | 1.0  |
| Prep       | P              | 1.0  |
| ConjS      | conj S         | 1.0  |
| ConjNP     | conj NP        | 1.0  |
| NPNP       | NP NP          | 1.0  |
## Probabilistic CKY

**Input: POS-tagged sentence**

\[ \text{John\_N eats\_V pie\_N with\_P cream\_N} \]

<table>
<thead>
<tr>
<th></th>
<th>John</th>
<th>eats</th>
<th>pie</th>
<th>with</th>
<th>cream</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>John</strong></td>
<td>Noun</td>
<td>VP</td>
<td>NOUN</td>
<td>VP</td>
<td>NOUN</td>
<td>1.0</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>eats</strong></td>
<td>Noun</td>
<td>VP</td>
<td>NOUN</td>
<td>VP</td>
<td>NOUN</td>
<td>1.0</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>pie</strong></td>
<td>Noun</td>
<td>VP</td>
<td>NOUN</td>
<td>VP</td>
<td>NOUN</td>
<td>1.0</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>with</strong></td>
<td>Noun</td>
<td>VP</td>
<td>NOUN</td>
<td>VP</td>
<td>NOUN</td>
<td>1.0</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>cream</strong></td>
<td>Noun</td>
<td>VP</td>
<td>NOUN</td>
<td>VP</td>
<td>NOUN</td>
<td>1.0</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Rules:**

- **S** → NP VP 0.8
- **S** → S ConjS 0.2
- **NP** → Noun 0.2
- **NP** → Det Noun 0.4
- **NP** → NP PP 0.2
- **NP** → NP ConjNP 0.2
- **VP** → Verb 0.3
- **VP** → Verb NP 0.3
- **VP** → Verb NPNP 0.1
- **VP** → VP PP 0.3
- **PP** → PP NP 1.0
- **Prep** → P 1.0
- **Noun** → N 1.0
- **Verb** → V 1.0
- **ConjS** → conj S 1.0
- **ConjNP** → conj NP 1.0
- **NPNP** → NP NP 1.0
Probabilistic CKY

Input: POS-tagged sentence
John_N eats_V pie_N with_P cream_N

S ⟶ NP VP 0.8
S ⟶ S ConjS 0.2
NP ⟶ Noun 0.2
NP ⟶ Det Noun 0.4
NP ⟶ NP PP 0.2
NP ⟶ NP ConjNP 0.2
VP ⟶ Verb 0.3
VP ⟶ Verb NP 0.3
VP ⟶ Verb NPNP 0.1
VP ⟶ VP PP 0.3
PP ⟶ PP NP 1.0
Prep ⟶ P 1.0
Noun ⟶ N 1.0
Verb ⟶ V 1.0
ConjS ⟶ conj S 1.0
ConjNP ⟶ conj NP 1.0
NPNP ⟶ NP NP 1.0
Probabilistic CKY

Input: POS-tagged sentence
John_N eats_V pie_N with_P cream_N

John Noun 1.0 NP 0.2
   VP Verb 1.0 0.3
      PP Prep 1.0
        S Noun 1.0 NP 0.2
           pie Noun 1.0
           with Prep 1.0
              cream Noun 1.0

S → NP VP 0.8
S → S ConjS 0.2
NP → Noun 0.2
NP → Det Noun 0.4
NP → NP PP 0.2
NP → NP ConjNP 0.2
VP → Verb 0.3
VP → Verb NPNP 0.1
VP → VP PP 0.3
PP → PP NP 1.0
Prep → P 1.0
Noun → N 1.0
Verb → V 1.0
ConjS → conj S 1.0
ConjNP → conj NP 1.0
NPNP → NP NP 1.0
Probabilistic CKY

**Input: POS-tagged sentence**

\[ \text{John}_N \text{ eats}_V \text{ pie}_N \text{ with}_P \text{ cream}_N \]

<table>
<thead>
<tr>
<th>John</th>
<th>eats</th>
<th>pie</th>
<th>with</th>
<th>cream</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>1.0</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
</tr>
</tbody>
</table>

```
S  \rightarrow \text{ NP VP} \quad 0.8
S  \rightarrow \text{ S ConjS} \quad 0.2
NP \rightarrow \text{ Noun} \quad 0.2
NP \rightarrow \text{ Det Noun} \quad 0.4
NP \rightarrow \text{ NP PP} \quad 0.2
NP \rightarrow \text{ NP ConjNP} \quad 0.2
VP \rightarrow \text{ Verb} \quad 0.3
VP \rightarrow \text{ Verb NP} \quad 0.3
VP \rightarrow \text{ Verb NPNP} \quad 0.1
PP \rightarrow \text{ PP NP} \quad 1.0
Prep \rightarrow \text{ P} \quad 1.0
Noun \rightarrow \text{ N} \quad 1.0
Verb \rightarrow \text{ V} \quad 1.0
ConjS \rightarrow \text{ conj S} \quad 1.0
ConjNP \rightarrow \text{ conj NP} \quad 1.0
NPNP \rightarrow \text{ NP NP} \quad 1.0
```
Probabilistic CKY

Input: POS-tagged sentence
John_N eats_V pie_N with_P cream_N

<table>
<thead>
<tr>
<th></th>
<th>John</th>
<th>eats</th>
<th>pie</th>
<th>with</th>
<th>cream</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun</td>
<td>NP</td>
<td>S</td>
<td>VP</td>
<td>VP</td>
<td>John</td>
</tr>
<tr>
<td>1.0</td>
<td>0.2</td>
<td>0.8·0.2·0.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Verb</td>
<td>VP</td>
<td>VP</td>
<td>VP</td>
<td>VP</td>
<td>eats</td>
</tr>
<tr>
<td>1.0</td>
<td>0.3</td>
<td>1·0.3·0.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Noun</td>
<td>NP</td>
<td></td>
<td></td>
<td></td>
<td>pie</td>
</tr>
<tr>
<td>1.0</td>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prep</td>
<td>PP</td>
<td></td>
<td></td>
<td></td>
<td>with</td>
</tr>
<tr>
<td>1.0</td>
<td></td>
<td>1·1·0.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Noun</td>
<td>NP</td>
<td></td>
<td></td>
<td></td>
<td>cream</td>
</tr>
<tr>
<td>1.0</td>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Production Rules:
- S → NP VP 0.8
- S → S ConjS 0.2
- NP → Noun 0.2
- NP → Det Noun 0.4
- NP → NP PP 0.2
- NP → NP ConjNP 0.2
- VP → Verb 0.3
- VP → Verb NP 0.3
- VP → Verb NPNP 0.1
- VP → VP PP 0.3
- PP → PP NP 1.0
- Prep → P 1.0
- Noun → N 1.0
- Verb → V 1.0
- ConjS → conj S 1.0
- ConjNP → conj NP 1.0
- NPNP → NP NP 1.0
Probabilistic CKY

Input: POS-tagged sentence
John_N eats_V pie_N with_P cream_N

<table>
<thead>
<tr>
<th></th>
<th>John</th>
<th>eats</th>
<th>pie</th>
<th>with</th>
<th>cream</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun</td>
<td>NP</td>
<td>0.2</td>
<td>S</td>
<td>0.8</td>
<td>0.2</td>
</tr>
<tr>
<td>Verb</td>
<td>VP</td>
<td>1.0</td>
<td>1.0</td>
<td>0.3</td>
<td>0.2</td>
</tr>
<tr>
<td>PP</td>
<td>1.0</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prep</td>
<td>PP</td>
<td>1.0</td>
<td>1.0</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>Noun</td>
<td>NP</td>
<td>1.0</td>
<td>0.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Noun</td>
<td>NP</td>
<td>1.0</td>
<td>0.2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Production rules:

- \( S \rightarrow NP \ VP \ 0.8 \)
- \( S \rightarrow S \ ConjS \ 0.2 \)
- \( NP \rightarrow Noun \ 0.2 \)
- \( NP \rightarrow Det \ Noun \ 0.4 \)
- \( NP \rightarrow NP \ PP \ 0.2 \)
- \( NP \rightarrow NP \ ConjNP \ 0.2 \)
- \( VP \rightarrow Verb \ 0.3 \)
- \( VP \rightarrow Verb \ NP \ 0.3 \)
- \( VP \rightarrow Verb \ NPNP \ 0.1 \)
- \( VP \rightarrow VP \ PP \ 0.3 \)
- \( PP \rightarrow PP \ NP \ 1.0 \)
- \( Prep \rightarrow P \ 1.0 \)
- \( Noun \rightarrow N \ 1.0 \)
- \( Verb \rightarrow V \ 1.0 \)
- \( ConjS \rightarrow conj \ S \ 1.0 \)
- \( ConjNP \rightarrow conj \ NP \ 1.0 \)
- \( NPNP \rightarrow NP \ NP \ 1.0 \)
**Probabilistic CKY**

**Input: POS-tagged sentence**

<table>
<thead>
<tr>
<th>John</th>
<th>eats</th>
<th>pie</th>
<th>with</th>
<th>cream</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun</td>
<td>NP</td>
<td>S</td>
<td>S</td>
<td>John</td>
</tr>
<tr>
<td>1.0</td>
<td>0.2</td>
<td>0.8·0.2·0.3</td>
<td>0.8·0.2·0.06</td>
<td>0.8</td>
</tr>
<tr>
<td>Verb</td>
<td>VP</td>
<td>VP</td>
<td>Noun</td>
<td>1.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.3</td>
<td>1·0.3·0.2</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>Prep</td>
<td>PP</td>
<td>with</td>
<td>Noun</td>
<td>1.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.2</td>
<td>1·1·0.2</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>Prep</td>
<td>PP</td>
<td>cream</td>
<td>Noun</td>
<td>1.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.2</td>
<td>1·1·0.2</td>
<td>0.2</td>
<td></td>
</tr>
</tbody>
</table>

Production Rules:

- **S** → NP VP 0.8
- **S** → S ConjS 0.2
- **NP** → Noun 0.2
- **NP** → Det Noun 0.4
- **NP** → NP PP 0.2
- **NP** → NP ConjNP 0.2
- **VP** → Verb 0.3
- **VP** → Verb NP 0.3
- **VP** → Verb NPNP 0.1
- **PP** → PP NP 1.0
- **ConjS** → conj S 1.0
- **ConjNP** → conj NP 1.0
- **NPNP** → NP NP 1.0
## Probabilistic CKY

### Input: POS-tagged sentence

\[\text{John}_N \text{ eats}_V \text{ pie}_N \text{ with}_P \text{ cream}_N\]

<table>
<thead>
<tr>
<th></th>
<th>John</th>
<th>eats</th>
<th>pie</th>
<th>with</th>
<th>cream</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Noun</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.06</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Noun</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.06</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prep</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Noun</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Probabilistic CKY Rules

- \(\text{S} \rightarrow \text{NP} \text{ VP} \) 0.8
- \(\text{S} \rightarrow \text{S Conjunction S} \) 0.2
- \(\text{NP} \rightarrow \text{Noun} \) 0.2
- \(\text{NP} \rightarrow \text{Det Noun} \) 0.4
- \(\text{NP} \rightarrow \text{NP PP} \) 0.2
- \(\text{NP} \rightarrow \text{NP Conjunction NP} \) 0.2
- \(\text{VP} \rightarrow \text{Verb} \) 0.3
- \(\text{VP} \rightarrow \text{Verb NP} \) 0.3
- \(\text{VP} \rightarrow \text{Verb NPNP} \) 0.1
- \(\text{PP} \rightarrow \text{PP NP} \) 1.0
- \(\text{Prep} \rightarrow \text{P} \) 1.0
- \(\text{Noun} \rightarrow \text{N} \) 1.0
- \(\text{Verb} \rightarrow \text{V} \) 1.0
- \(\text{Conjunction S} \rightarrow \text{Conjunction S} \) 1.0
- \(\text{Conjunction NP} \rightarrow \text{Conjunction NP} \) 1.0
- \(\text{NPNP} \rightarrow \text{NP NP} \) 1.0
Probabilistic CKY

Input: POS-tagged sentence
John_N eats_V pie_N with_P cream_N

John  eats  pie  with  cream
      NP  1.0  0.2  
Noun  S  0.8 0.2 0.3  
Verb  VP  1.0  0.3  
Noun  NP  1.0  0.2

with
      PP  1.0  
Prep  PP  0.2  
Noun  NP  1.0  0.2

cream
      Noun  NP  1.0  0.2

S  →  NP VP  0.8
S  →  S ConjS  0.2
NP  →  Noun  0.2
NP  →  Det Noun  0.4
NP  →  NP PP  0.2
NP  →  NP ConjNP  0.2
VP  →  Verb  0.3
VP  →  Verb NP  0.3
VP  →  Verb NPNP  0.1
PP  →  PP NP  1.0
Prep  →  P  1.0
Noun  →  N  1.0
Verb  →  V  1.0
ConjS  →  conj S  1.0
ConjNP  →  conj NP  1.0
NPNP  →  NP NP  1.0
Probabilistic CKY

Input: POS-tagged sentence
John_N eats_V pie_N with_P cream_N

S → NP VP 0.8
S → S ConjS 0.2
NP → Noun 0.2
NP → Det Noun 0.4
NP → NP PP 0.2
NP → NP ConjNP 0.2
NP → VP NP 0.3
VP → Verb NP 0.3
VP → Verb NPNP 0.1
VP → VP PP 0.3
PP → PP NP 1.0
Prep → P 1.0
Noun → N 1.0
Verb → V 1.0
ConjS → conj S 1.0
ConjNP → conj NP 1.0
NPNP → NP NP 1.0
## Probabilistic CKY

**Input: POS-tagged sentence**

\[\text{John}_N \ \text{eats}_V \ \text{pie}_N \ \text{with}_P \ \text{cream}_N\]

<table>
<thead>
<tr>
<th></th>
<th>John</th>
<th>eats</th>
<th>pie</th>
<th>with</th>
<th>cream</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun</td>
<td>S</td>
<td>S</td>
<td></td>
<td></td>
<td>John</td>
</tr>
<tr>
<td>1.0</td>
<td>0.2</td>
<td>0.8</td>
<td>0.8</td>
<td>0.2</td>
<td>0.06</td>
</tr>
<tr>
<td>Verb</td>
<td>VP</td>
<td>VP</td>
<td></td>
<td></td>
<td>eats</td>
</tr>
<tr>
<td>1.0</td>
<td>0.3</td>
<td>1.0</td>
<td>1.0</td>
<td>0.2</td>
<td>0.06</td>
</tr>
<tr>
<td>Noun</td>
<td>NP</td>
<td>NP</td>
<td></td>
<td></td>
<td>pie</td>
</tr>
<tr>
<td>1.0</td>
<td>0.2</td>
<td>0.2</td>
<td></td>
<td>1.0</td>
<td>0.008</td>
</tr>
<tr>
<td>Prep</td>
<td>PP</td>
<td></td>
<td></td>
<td></td>
<td>with</td>
</tr>
<tr>
<td>1.0</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Noun</td>
<td>NP</td>
<td></td>
<td></td>
<td></td>
<td>cream</td>
</tr>
<tr>
<td>1.0</td>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Production rules:

- \( S \rightarrow \text{NP} \ \text{VP} \) with probability 0.8
- \( S \rightarrow \text{S} \ \text{ConjS} \) with probability 0.2
- \( \text{NP} \rightarrow \text{Noun} \) with probability 0.2
- \( \text{NP} \rightarrow \text{Det} \ \text{Noun} \) with probability 0.4
- \( \text{NP} \rightarrow \text{NP} \ \text{PP} \) with probability 0.2
- \( \text{NP} \rightarrow \text{NP} \ \text{ConjNP} \) with probability 0.2
- \( \text{VP} \rightarrow \text{Verb} \) with probability 0.3
- \( \text{VP} \rightarrow \text{Verb} \ \text{NP} \) with probability 0.3
- \( \text{VP} \rightarrow \text{Verb} \ \text{NNP} \) with probability 0.1
- \( \text{PP} \rightarrow \text{PP} \ \text{NP} \) with probability 1.0
- \( \text{Prep} \rightarrow \text{P} \) with probability 1.0
- \( \text{Noun} \rightarrow \text{N} \) with probability 1.0
- \( \text{Verb} \rightarrow \text{V} \) with probability 1.0
- \( \text{ConjS} \rightarrow \text{conj} \ \text{S} \) with probability 1.0
- \( \text{ConjNP} \rightarrow \text{conj} \ \text{NP} \) with probability 1.0
- \( \text{NPNP} \rightarrow \text{NP} \ \text{NP} \) with probability 1.0
Probabilistic CKY

Input: POS-tagged sentence
John_N eats_V pie_N with_P cream_N

S → NP VP 0.8
S → S ConjS 0.2
NP → Noun 0.2
NP → Det Noun 0.4
NP → NP PP 0.2
NP → NP ConjNP 0.2
VP → Verb 0.3
VP → Verb NP 0.3
VP → Verb NPNP 0.1
VP → VP PP 0.3
PP → PP NP 1.0
Prep → P 1.0
Noun → N 1.0
Verb → V 1.0
ConjS → conj S 1.0
ConjNP → conj NP 1.0
NPNP → NP NP 1.0
Probabilistic CKY

Input: POS-tagged sentence
John_N eats_V pie_N with_P cream_N

<table>
<thead>
<tr>
<th></th>
<th>John</th>
<th>eats</th>
<th>pie</th>
<th>with</th>
<th>cream</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun</td>
<td>NP</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>John</td>
</tr>
<tr>
<td>1.0</td>
<td>0.2</td>
<td>0.8</td>
<td>0.2</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>Verb</td>
<td>VP</td>
<td>VP</td>
<td>NP</td>
<td>PP</td>
<td>eats</td>
</tr>
<tr>
<td>1.0</td>
<td>0.3</td>
<td>1.0</td>
<td>1.0</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>Noun</td>
<td>NP</td>
<td>NP</td>
<td>NP</td>
<td>PP</td>
<td>pie</td>
</tr>
<tr>
<td>1.0</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prep</td>
<td>PP</td>
<td>PP</td>
<td>PP</td>
<td>PP</td>
<td>with</td>
</tr>
<tr>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Noun</td>
<td>NP</td>
<td>NP</td>
<td>cream</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>0.2</td>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Production Rules:
- S → NP VP 0.8
- S → S ConjS 0.2
- NP → Noun 0.2
- NP → Det Noun 0.4
- NP → NP PP 0.2
- NP → NP ConjNP 0.2
- VP → Verb 0.3
- VP → Verb NP 0.3
- VP → Verb NPNP 0.1
- PP → PP NP 1.0
- Prep → P 1.0
- Noun → N 1.0
- Verb → V 1.0
- ConjS → conj S 1.0
- ConjNP → conj NP 1.0
- NPNP → NP NP 1.0
## Probabilistic CKY

**Input:** POS-tagged sentence

\[ \text{John}_N \text{ eats}_V \text{ pie}_N \text{ with}_P \text{ cream}_N \]

<table>
<thead>
<tr>
<th></th>
<th>John</th>
<th>eats</th>
<th>pie</th>
<th>with</th>
<th>cream</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NP</strong></td>
<td>Noun 1.0</td>
<td><strong>VP</strong></td>
<td>1.0</td>
<td>0.2 [0.8 \cdot 0.2 \cdot 0.3]</td>
<td>0.8 \cdot 0.2 \cdot 0.06</td>
</tr>
<tr>
<td></td>
<td>VP 1.0</td>
<td><strong>NP</strong></td>
<td>1.0</td>
<td>0.3</td>
<td>1 \cdot 0.3 \cdot 0.2 = 0.06</td>
</tr>
<tr>
<td>Prep 1.0</td>
<td><strong>PP</strong></td>
<td>1 \cdot 1 \cdot 0.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Noun</strong></td>
<td>1.0</td>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Derivation:**

\[
S \rightarrow NP \ VP \quad 0.8 \\
S \rightarrow S \ ConjS \quad 0.2 \\
NP \rightarrow Noun \quad 0.2 \\
NP \rightarrow Det \ Noun \quad 0.4 \\
NP \rightarrow NP \ PP \quad 0.2 \\
NP \rightarrow NP \ ConjNP \quad 0.2 \\
VP \rightarrow Verb \quad 0.3 \\
VP \rightarrow Verb \ NP \quad 0.3 \\
VP \rightarrow Verb \ NPNP \quad 0.1 \\
PP \rightarrow PP \ NP \quad 1.0 \\
Prep \rightarrow P \quad 1.0 \\
Noun \rightarrow N \quad 1.0 \\
Verb \rightarrow V \quad 1.0 \\
ConjS \rightarrow conj \ S \quad 1.0 \\
ConjNP \rightarrow conj \ NP \quad 1.0 \\
NPNP \rightarrow NP \ NP \quad 1.0 \\
\]
## Probabilistic CKY

### Input: POS-tagged sentence

John\_N eats\_V pie\_N with\_P cream\_N

<table>
<thead>
<tr>
<th></th>
<th>John</th>
<th>eats</th>
<th>pie</th>
<th>with</th>
<th>cream</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun</td>
<td>NP</td>
<td>S</td>
<td>S</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Verb</td>
<td>VP</td>
<td>VP</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
\text{S} \rightarrow \text{NP} \text{ VP} \quad 0.8
\]

\[
\text{S} \rightarrow \text{S ConjS} \quad 0.2
\]

\[
\text{NP} \rightarrow \text{Noun} \quad 0.2
\]

\[
\text{NP} \rightarrow \text{Det Noun} \quad 0.4
\]

\[
\text{NP} \rightarrow \text{NP PP} \quad 0.2
\]

\[
\text{NP} \rightarrow \text{NP ConjNP} \quad 0.2
\]

\[
\text{VP} \rightarrow \text{Verb} \quad 0.3
\]

\[
\text{VP} \rightarrow \text{Verb NP} \quad 0.3
\]

\[
\text{VP} \rightarrow \text{Verb NPNP} \quad 0.1
\]

\[
\text{PP} \rightarrow \text{PP NP} \quad 1.0
\]

\[
\text{Prep} \rightarrow \text{P} \quad 1.0
\]

\[
\text{Noun} \rightarrow \text{N} \quad 1.0
\]

\[
\text{Verb} \rightarrow \text{V} \quad 1.0
\]

\[
\text{ConjS} \rightarrow \text{conj S} \quad 1.0
\]

\[
\text{ConjNP} \rightarrow \text{conj NP} \quad 1.0
\]

\[
\text{NPNP} \rightarrow \text{NP NP} \quad 1.0
\]

---

CS447 Natural Language Processing (J. Hockenmaier) https://courses.grainger.illinois.edu/cs447/
**Probabilistic CKY**

**Input: POS-tagged sentence**

John_N eats_V pie_N with_P cream_N

<table>
<thead>
<tr>
<th></th>
<th>John</th>
<th>eats</th>
<th>pie</th>
<th>with</th>
<th>cream</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Noun</strong></td>
<td>NP</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>0.8 · 0.2 · 0.3</td>
<td>0.8 · 0.2 · 0.06</td>
<td>0.2 · 0.0036 · 0.8</td>
<td>0.2 · 0.0036 · 0.008</td>
</tr>
<tr>
<td><strong>Verb</strong></td>
<td>VP</td>
<td>VP</td>
<td>VP</td>
<td>VP</td>
<td>VP</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>1.0 · 0.3 · 0.2</td>
<td>1.0 · 0.3 · 0.2</td>
<td>1.0 · 0.3 · 0.2</td>
<td>1.0 · 0.3 · 0.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.008</td>
<td></td>
<td>0.008</td>
</tr>
</tbody>
</table>

**Probabilistic Rules**

- \( S \rightarrow NP \ VP \ 0.8 \)
- \( S \rightarrow S \ ConjS \ 0.2 \)
- \( NP \rightarrow Noun \ 0.2 \)
- \( NP \rightarrow Det \ Noun \ 0.4 \)
- \( NP \rightarrow NP \ PP \ 0.2 \)
- \( NP \rightarrow NP \ ConjNP \ 0.2 \)
- \( VP \rightarrow Verb \ 0.3 \)
- \( VP \rightarrow Verb \ NP \ 0.3 \)
- \( VP \rightarrow Verb \ NPNP \ 0.1 \)
- \( VP \rightarrow VP \ PP \ 0.3 \)
- \( PP \rightarrow PP \ NP \ 1.0 \)
- \( Prep \rightarrow P \ 1.0 \)
- \( Noun \rightarrow N \ 1.0 \)
- \( Verb \rightarrow V \ 1.0 \)
- \( ConjS \rightarrow conj \ S \ 1.0 \)
- \( ConjNP \rightarrow conj \ NP \ 1.0 \)
- \( NPNP \rightarrow NP \ NP \ 1.0 \)
Probabilistic CKY

Input: POS-tagged sentence
John_N eats_V pie_N with_P cream_N

<table>
<thead>
<tr>
<th></th>
<th>John</th>
<th>eats</th>
<th>pie</th>
<th>with</th>
<th>cream</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun</td>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Verb</td>
<td>1.0</td>
<td>0.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Noun</td>
<td>1.0</td>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prep</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Noun</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

John \(\rightarrow\) NP VP \(0.8\)
S \(\rightarrow\) NP VP \(0.8\)
S \(\rightarrow\) S ConjS \(0.2\)
NP \(\rightarrow\) Noun \(0.2\)
NP \(\rightarrow\) Det Noun \(0.4\)
NP \(\rightarrow\) NP PP \(0.2\)
NP \(\rightarrow\) NP ConjNP \(0.2\)
VP \(\rightarrow\) Verb \(0.3\)
VP \(\rightarrow\) Verb NP \(0.3\)
VP \(\rightarrow\) Verb NPNP \(0.1\)
PP \(\rightarrow\) PP NP \(1.0\)
Prep \(\rightarrow\) P \(1.0\)
Noun \(\rightarrow\) N \(1.0\)
Verb \(\rightarrow\) V \(1.0\)
ConjS \(\rightarrow\) conj S \(1.0\)
ConjNP \(\rightarrow\) conj NP \(1.0\)
NPNP \(\rightarrow\) NP NP \(1.0\)
## Probabilistic CKY

**Input:** POS-tagged sentence

John\_N eats\_V pie\_N with\_P cream\_N

<table>
<thead>
<tr>
<th></th>
<th>John</th>
<th>eats</th>
<th>pie</th>
<th>with</th>
<th>cream</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Noun \ 0.2</td>
<td>S \ 0.8 \ 0.2 \ 0.3</td>
<td>S \ 0.8 \ 0.2 \ 0.06</td>
<td>S \ 0.2 \ 0.0036 \ 0.8</td>
<td></td>
</tr>
<tr>
<td>Verb</td>
<td>VP \ 1.0 \ 0.3</td>
<td>VP \ 1 \ 0.3 \ 0.2</td>
<td>VP \ 0.0036</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Noun</td>
<td>NP \ 1.0 \ 0.2</td>
<td>NP \ 0.2 \ 0.2 \ 0.2</td>
<td>NP \ 0.2 \ 0.2 \ 0.2</td>
<td>NP \ 0.2 \ 0.2 \ 0.2</td>
<td></td>
</tr>
</tbody>
</table>

The probabilistic rules for CKY are:

- $S \rightarrow NP \ VP \ 0.8$
- $S \rightarrow S \ ConjS \ 0.2$
- $NP \rightarrow Noun \ 0.2$
- $NP \rightarrow Det \ Noun \ 0.4$
- $NP \rightarrow NP \ PP \ 0.2$
- $NP \rightarrow NP \ ConjNP \ 0.2$
- $VP \rightarrow Verb \ 0.3$
- $VP \rightarrow Verb \ NP \ 0.3$
- $VP \rightarrow Verb \ NPNP \ 0.1$
- $VP \rightarrow VP \ PP \ 0.3$
- $PP \rightarrow PP \ NP \ 1.0$
- $Prep \rightarrow P \ 1.0$
- $Noun \rightarrow N \ 1.0$
- $Verb \rightarrow V \ 1.0$
- $ConjS \rightarrow conj \ S \ 1.0$
- $ConjNP \rightarrow conj \ NP \ 1.0$
- $NPNP \rightarrow NP \ NP \ 1.0$
# Probabilistic CKY

**Input: POS-tagged sentence**

\[ \text{John}_N \text{ eats}_V \text{ pie}_N \text{ with}_P \text{ cream}_N \]

<table>
<thead>
<tr>
<th></th>
<th>John</th>
<th>eats</th>
<th>pie</th>
<th>with</th>
<th>cream</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun</td>
<td>1.0</td>
<td>0.2</td>
<td>S</td>
<td>0.8 · 0.2 · 0.3</td>
<td>0.8 · 0.2 · 0.06</td>
</tr>
<tr>
<td>Verb</td>
<td>1.0</td>
<td>0.3</td>
<td>VP</td>
<td>1.0 · 0.3 · 0.2</td>
<td>0.06</td>
</tr>
<tr>
<td>Noun</td>
<td>1.0</td>
<td>0.2</td>
<td>NP</td>
<td>0.2 · 0.2 · 0.2</td>
<td>0.008</td>
</tr>
<tr>
<td>Prep</td>
<td>1.0</td>
<td>PP</td>
<td>VP</td>
<td>1.0 · 1.0 · 0.2</td>
<td>0.0036</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
S & \rightarrow \text{NP VP} & 0.8 \\
S & \rightarrow \text{S ConjS} & 0.2 \\
\text{NP} & \rightarrow \text{Noun} & 0.2 \\
\text{NP} & \rightarrow \text{Det Noun} & 0.4 \\
\text{NP} & \rightarrow \text{NP PP} & 0.2 \\
\text{NP} & \rightarrow \text{NP ConjNP} & 0.2 \\
\text{VP} & \rightarrow \text{Verb} & 0.3 \\
\text{VP} & \rightarrow \text{Verb NP} & 0.3 \\
\text{VP} & \rightarrow \text{Verb NPNP} & 0.1 \\
\text{PP} & \rightarrow \text{PP NP} & 1.0 \\
\text{Prep} & \rightarrow \text{P} & 1.0 \\
\text{Noun} & \rightarrow \text{N} & 1.0 \\
\text{Verb} & \rightarrow \text{V} & 1.0 \\
\text{ConjS} & \rightarrow \text{conj S} & 1.0 \\
\text{ConjNP} & \rightarrow \text{conj NP} & 1.0 \\
\text{NPNP} & \rightarrow \text{NP NP} & 1.0
\end{align*}
\]
Extracting the final tree

**Input: POS-tagged sentence**

\[\text{John\_N eats\_V pie\_N with\_P cream\_N}\]

<table>
<thead>
<tr>
<th></th>
<th>John</th>
<th>eats</th>
<th>pie</th>
<th>with</th>
<th>cream</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun</td>
<td>1.0</td>
<td>NP</td>
<td>0.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>0.8</td>
<td>0.2</td>
<td>0.3</td>
<td>John</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>0.8</td>
<td>0.2</td>
<td>0.06</td>
<td></td>
</tr>
<tr>
<td>Verb</td>
<td>1.0</td>
<td>VP</td>
<td>0.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>VP</td>
<td>1.0</td>
<td>0.3</td>
<td>0.2</td>
<td>eats</td>
</tr>
<tr>
<td></td>
<td>VP</td>
<td>1.0</td>
<td>0.3</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NP</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>pie</td>
</tr>
<tr>
<td></td>
<td>NP</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>Prep</td>
<td>1.0</td>
<td>PP</td>
<td>0.2</td>
<td></td>
<td>with</td>
</tr>
<tr>
<td></td>
<td>PP</td>
<td>1.0</td>
<td>1.0</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>Prep</td>
<td>1.0</td>
<td>PP</td>
<td>0.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PP</td>
<td>1.0</td>
<td>1.0</td>
<td>0.2</td>
<td>cream</td>
</tr>
<tr>
<td></td>
<td>PP</td>
<td>1.0</td>
<td>1.0</td>
<td>0.2</td>
<td></td>
</tr>
</tbody>
</table>
Extracting the final tree

Input: POS-tagged sentence
John_N eats_V pie_N with_P cream_N

(S (NP (N John))
  (VP (VP (Verb eats)
    (NP (Noun pie)))
  (PP (Prep with)
    (NP (Noun cream))))
Shortcomings of PCFGs
How well can a PCFG model the distribution of trees?

PCFGs make **independence assumptions**: Only the label of a node determines what children it has.

Factors that influence these assumptions:

- **Shape** of the trees:
  A corpus with **flat trees** (i.e. few nodes/sentence) results in a model with few independence assumptions.

- **Labeling** of the trees:
  A corpus with **many node labels** (nonterminals) results in a model with few independence assumptions.
Example 1: flat trees

What sentences would a PCFG estimated from this corpus generate?
Example 2: deep trees, few labels

What sentences would a PCFG estimated from this corpus generate?
Example 3: deep trees, many labels

What sentences would a PCFG estimated from this corpus generate?
Aside: Bias/Variance tradeoff

A probability model has low bias if it makes few independence assumptions.
⇒ It can capture the structures in the training data.

But: this typically leads to a more fine-grained partitioning of the training data.

Hence, fewer data points are available to estimate the model parameters. This yields poor estimates of the distribution.

That is, such models have high variance
Two ways to improve performance

... change the (internal) grammar:
- **Parent annotation/state splits:**
  Not all NPs/VPs/DTs/... are the same.
  It matters where they are in the tree

... change the probability model:
- **Lexicalization:**
  Words matter!
- **Markovization:**
  Generalizing the rules
The parent transformation

PCFGs assume the expansion of any nonterminal is independent of its parent.

But this is not true: NP subjects more likely to be modified than objects.

We can change the grammar by adding the name of the parent node to each nonterminal

(a) \[ \text{VP} \]
   \[ \text{V} \quad \text{NP} \]
   \[ \text{NP} \quad \text{PP} \]
   \[ \text{Det} \quad \text{N} \quad \text{P} \quad \text{NP} \]

(b) \[ \text{VP}^{\text{S}} \]
   \[ \text{V} \quad \text{NP}^{\text{VP}} \]
   \[ \text{NP}^{\text{NP}} \quad \text{PP}^{\text{NP}} \]
   \[ \text{Det} \quad \text{N} \quad \text{P} \quad \text{NP}^{\text{PP}} \]
Lexicalization

PCFGs can’t distinguish between “eat sushi with chopsticks” and “eat sushi with tuna”.

We need to take words into account!

\[
P(\text{VP}_{\text{eat}} \rightarrow \text{VP PP}_{\text{with chopsticks}} \mid \text{VP}_{\text{eat}}) \\
\text{vs. } P(\text{VP}_{\text{eat}} \rightarrow \text{VP PP}_{\text{with tuna}} \mid \text{VP}_{\text{eat}})
\]

Problem: sparse data \((\text{PP}_{\text{with fatty|white|... tuna|...}})\)
Solution: only take **head words** into account!

Assumption: each constituent has one head word.
Lexicalized PCFGs

At the root (start symbol $S$), generate the head word of the sentence, $w_s$, with $P(w_s)$

Lexicalized rule probabilities:
Every nonterminal is lexicalized: $X_{w_x}$
Condition rules $X_{w_x} \rightarrow \alpha Y \beta$ on the lexicalized LHS $X_{w_x}$
$P(X_{w_x} \rightarrow \alpha Y \beta \mid X_{w_x})$

Word-word dependencies:
For each nonterminal $Y$ in RHS of a rule $X_{w_x} \rightarrow \alpha Y \beta$, condition $w_Y$ (the head word of $Y$) on $X$ and $w_x$:
$P(w_Y \mid Y, X, w_x)$
Dealing with unknown words

A lexicalized PCFG assigns zero probability to any word that does not appear in the training data.

Solution:

Training: Replace rare words in training data with a token ‘UNKNOWN’.

Testing: Replace unseen words with ‘UNKNOWN’
Markov PCFGs (Collins parser)

The RHS of each CFG rule consists of:
one head $H_X$, $n$ left sisters $L_i$ and $m$ right sisters $R_i$:

$$X \rightarrow L_n \ldots L_1 \underbrace{H_X}_{\text{left sisters}} R_1 \ldots R_m \underbrace{\text{right sisters}}_{\text{right sisters}}$$

Replace rule probabilities with a generative process:
For each nonterminal $X$

- generate its head $H_X$ (nonterminal or terminal)
- then generate its left sisters $L_{1..n}$ and a STOP symbol conditioned on $H_X$
- then generate its right sisters $R_{1..n}$ and a STOP symbol conditioned on $H_X$
Penn Treebank Parsing
The Penn Treebank

The first publicly available syntactically annotated corpus
Wall Street Journal (50,000 sentences, 1 million words)
also Switchboard, Brown corpus, ATIS

The annotation:
- POS-tagged (Ratnaparkhi’s MXPOST)
- Manually annotated with phrase-structure trees
- Richer than standard CFG: *Traces* and other *null elements* used to represent non-local dependencies (designed to allow extraction of predicate-argument structure), although these are typically removed when we do parsing
[more on non-local dependencies and traces later in the semester]

The standard data set for English phrase-structure parsers
The Treebank label set

48 preterminals (tags):
- 36 POS tags, 12 other symbols (punctuation etc.)
- Simplified version of Brown tagset (87 tags)
  (cf. Lancaster-Oslo/Bergen (LOB) tag set: 126 tags)

14 nonterminals:
Standard inventory (S, NP, VP, PP, ADJP, ADVP, SBAR,…)  
Many nonterminals have function tags indicating their syntactic roles (NP-SBJ: subject NP) or what role they play
(e.g. PP-LOC: locative PP, i.e. indicating a location [“in NYC”]
  PP-DIR: directional PP, indicating a direction [“to NYC”],
  ADVP-MNR: manner adverb [“slowly”]).

For historical reasons, these function tags are typically removed before parsing.
A simple example

Relatively flat structures:
- There is no noun level
- VP arguments and adjuncts appear at the same level

Function tags, e.g. -SBJ (subject), -MNR (manner)
A more realistic (partial) example

*Until Congress acts, the government hasn't any authority to issue new debt obligations of any kind, the Treasury said ....*
The Penn Treebank CFG

The Penn Treebank uses very flat rules, e.g.:

```
NP → DT JJ NN
NP → DT JJ NNS
NP → DT JJ NN NN
NP → DT JJ JJ NN
NP → DT JJ CD NNS
NP → RB DT JJ NN NN
NP → RB DT JJ JJ NNS
NP → DT JJ JJ NNP NNS
NP → DT NNP NNP NNP NNP JJ NN
NP → DT JJ NNP CC JJ JJ NN NNS
NP → RB DT JJS NN NN SBAR
NP → DT VBG JJ NNP NNP CC NNP
NP → DT JJ NNS , NNS CC NN NNS NN
NP → DT JJ JJ VBG NN NNP NNP FW NNP
NP → NP JJ , JJ `` SBAR '' NNS
```

Basic PCFGs don’t work well on the Penn Treebank

- Many of these rules appear only once.
- But many of these rules are very similar.
  Can we generalize by not treating each rule as atomic?
Summary

The Penn Treebank has a large number of very flat rules.

Accurate parsing requires modifications to basic PCFG models:
— Generalizing across similar rules ("Markov PCFGs")
— Modeling word-word dependencies
  (although this does not help as much as people used to think)
— Refining the nonterminals to capture more context

How much of this transfers to other treebanks or languages?
CFG Parser Evaluation
Precision and recall

Precision and recall were originally developed as evaluation metrics for information retrieval:

- **Precision**: What percentage of retrieved documents are relevant to the query?
- **Recall**: What percentage of relevant documents were retrieved?

In NLP, they are often used in addition to accuracy:

- **Precision**: What percentage of items that were assigned label X do actually have label X in the test data?
- **Recall**: What percentage of items that have label X in the test data were assigned label X by the system?

Particularly useful when there are more than two labels.
True vs. false positives, false negatives

- True positives: Items that were labeled X by the system, and should be labeled X.
- False positives: Items that were labeled X by the system, but should not be labeled X.
- False negatives: Items that were not labeled X by the system, but should be labeled X.
Precision, recall, f-measure

**Items labeled X in the gold standard**

-items labeled X ('truth')

\[ = TP + FN \]

**False Negatives (FN)**

**True Positives (TP)**

**False Positives (FP)**

**Precision:**

\[ P = \frac{TP}{TP + FP} \]

**Recall:**

\[ R = \frac{TP}{TP + FN} \]

**F-measure:**

harmonic mean of precision and recall

\[ F = \frac{(2 \cdot P \cdot R)}{(P + R)} \]
Evalb ("Parseeval")

Measures recovery of phrase-structure trees.

**Labeled:** span and label (NP, PP,...) has to be right.
[Earlier variant— unlabeled: span of nodes has to be right]

Two aspects of evaluation

**Precision:** *How many of the predicted nodes are correct?*

**Recall:** *How many of the correct nodes were predicted?*

*Usually combined into one metric (F-measure):*

\[ P = \frac{\text{#correctly predicted nodes}}{\text{#predicted nodes}} \]

\[ R = \frac{\text{#correctly predicted nodes}}{\text{#correct nodes}} \]

\[ F = \frac{2PR}{P + R} \]
Parseval in practice

Gold standard

Parser output

eat sushi with tuna: Precision: 4/5 Recall: 4/5
eat sushi with chopsticks: Precision: 4/5 Recall: 4/5