CKY algorithm / PCFGs

CS 685, Spring 2022
Introduction to Natural Language Processing
http://people.cs.umass.edu/~miyyer/cs685/

Mohit Iyyer
College of Information and Computer Sciences
University of Massachusetts Amherst

some slides from Brendan O’Connor
today we’ll be doing parsing: given a grammar, how do we use it to parse a sentence?
A mouse eats a cat.
why parsing?

• historically: good way to obtain features for downstream tasks
• today: can sometimes (not always) use syntax to improve neural models
• always useful for chunking text into phrases
• parsing makes for good probe tasks on top of neural models
• useful for psycholinguistics experiments
Ambiguity in parsing

Syntactic ambiguity is endemic to natural language:\(^1\)

- **Attachment ambiguity:** we eat sushi with chopsticks, I shot an elephant in my pajamas.
- **Modifier scope:** southern food store
- **Particle versus preposition:** The puppy tore up the staircase.
- **Complement structure:** The tourists objected to the guide that they couldn’t hear.
- **Coordination scope:** “I see,” said the blind man, as he picked up the hammer and saw.
- **Multiple gap constructions:** The chicken is ready to eat

\(^1\)Examples borrowed from Dan Klein
Formal Definition of Context-Free Grammar

• A context-free grammar \( G \) is defined by four parameters: \( N, \Sigma, R, S \)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N )</td>
<td>a set of <strong>non-terminal symbols</strong> (or <strong>variables</strong>)</td>
</tr>
<tr>
<td>( \Sigma )</td>
<td>a set of <strong>terminal symbols</strong> (disjoint from ( N ))</td>
</tr>
<tr>
<td>( R )</td>
<td>a set of <strong>rules</strong> or productions, each of the form ( A \rightarrow \beta ), where ( A ) is a non-terminal, ( \beta ) is a string of symbols from the infinite set of strings ((\Sigma \cup N)^*)</td>
</tr>
<tr>
<td>( S )</td>
<td>a designated <strong>start symbol</strong> and a member of ( N )</td>
</tr>
</tbody>
</table>
let’s start with a simple CFG

- $S \rightarrow NP\ VP$
- $NN \rightarrow \text{“dog”}$
- $NP \rightarrow DT\ JJ\ NN$
first, let’s convert this to Chomsky Normal Form (CNF)

\[
\begin{align*}
N & \quad \text{a set of non-terminal symbols (or variables)} \\
\Sigma & \quad \text{a set of terminal symbols (disjoint from } N) \\
R & \quad \text{a set of rules or productions, each of the form } A \to \beta \\
S & \quad \text{a designated start symbol and a member of } N
\end{align*}
\]

\(\beta\) is either a single terminal from \(\Sigma\) or a pair of non-terminals from \(N\)
converting the simple CFG

- $S \rightarrow NP \ VP$
- $NN \rightarrow \text{“dog”}$
- $NP \rightarrow DT \ JJ \ NN$
  - $NP \rightarrow X \ NN$
  - $X \rightarrow DT \ JJ$

we can convert any CFG to a CNF. this is a necessary preprocessing step for the basic CKY alg., produces binary trees!
Parsing!

• Given a sentence and a CNF, we want to search through the space of all possible parses for that sentence to find:
  • any valid parse for that sentence
  • all valid parses
  • the most probable parse

• Two approaches
  • bottom-up: start from the words and attempt to construct the tree
  • top-down: start from START symbol and keep expanding until you can construct the sentence

Pros and cons of each?
today: CKY algorithm

- Cocke-Kasami-Younger (independently discovered, also known as CYK)
- a bottom-up parser for CFGs (and PCFGs)

“I shot an elephant in my pajamas. How he got into my pajamas, I'll never know.”
— Groucho Marx
let’s say I have this CNF

- $S \rightarrow NP \ VP$
- $PP \rightarrow IN \ NP$
- $NP \rightarrow DET \ NP$
- $NP \rightarrow NP \ PP$
- $VP \rightarrow VBD \ NP$
- $VP \rightarrow VP \ PP$
- $NP \rightarrow PRP$\$ \ NP$

- $DET \rightarrow “an”$
- $VBD \rightarrow “shot”$
- $NP \rightarrow “pajamas”$
- $NP \rightarrow “elephant”$
- $NP \rightarrow “I”$
- $PRP \rightarrow “I”$
- $IN \rightarrow “in”$
- $PRP$\$ \rightarrow “my”$
I shot an elephant in my pajamas

build a chart!
top-right is root
I shot an elephant in my pajamas

NP / PRP

VBD

DET

NP

IN

PRP$

NP

fill in first level (words) with possible derivations
I shot an elephant in my pajamas onto the second level!
I shot an elephant in my pajamas onto the second level!

This cell spans the phrase “I shot”
I shot an elephant in my pajamas onto the second level!

what does this cell span?
I shot an elephant in my pajamas

<table>
<thead>
<tr>
<th>NP / PRP</th>
<th>VBD</th>
<th>DET</th>
</tr>
</thead>
</table>

onto the second level!

do any rules produce NP VBD or PRP VBD?

- S ▸ NP VP
- PP ▸ IN NP
- NP ▸ DET NP
- NP ▸ NP PP
- VP ▸ VBD NP
- VP ▸ VP PP
- NP ▸ PRP$ NP
I shot an elephant in my pajamas onto the second level!

Do any rules produce VBD DET?
I shot an elephant in my pajamas

onto the second level!
do any rules produce DET NP?
I shot an elephant in my pajamas onto the second level!

**do any rules produce DET NP?** Yes!

**onto the second level!**
I shot an elephant in my pajamas onto the third level!
I shot an elephant in my pajamas

onto the third level!

two ways to form
“I shot an”:
I + shot an
I shot an elephant in my pajamas onto the third level!

two ways to form “I shot an”:
  I + shot an
  I shot + an
I shot an elephant in my pajamas onto the third level!

what about this cell?
I shot an elephant in my pajamas onto the third level!

what about this cell?
I shot an elephant in my pajamas onto the third level!
| NP / PRP | Ø   | Ø   |   |   |   |   |
| VBD     | Ø   | VP  |   |   |   |   |
| DET     | NP  | Ø   |   |   |   |   |
| NP      | Ø   | Ø   |   |   |   |   |
| IN      | Ø   | PP  |   |   |   |   |
| PRP$    | NP  | NP  |   |   |   |   |
|         |     |     | I|  |  |   |

I shot an elephant in my pajamas
I shot an elephant in my pajamas onto the fourth level!

what are our options here?
I shot an elephant in my pajamas onto the fourth level!

what are our options here?
I shot an elephant in my pajamas onto the fourth level!
I shot an elephant in my pajamas

<table>
<thead>
<tr>
<th>NP / PRP</th>
<th>Ø</th>
<th>Ø</th>
<th>S</th>
<th>Ø</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD</td>
<td>Ø</td>
<td>Ø</td>
<td>VP</td>
<td>Ø</td>
</tr>
<tr>
<td>DET</td>
<td>NP</td>
<td>Ø</td>
<td>Ø</td>
<td>NP</td>
</tr>
<tr>
<td>NP</td>
<td>Ø</td>
<td>Ø</td>
<td>IN</td>
<td>Ø</td>
</tr>
<tr>
<td>IN</td>
<td>Ø</td>
<td>PP</td>
<td>PRP$</td>
<td>NP</td>
</tr>
<tr>
<td>PRP$</td>
<td>NP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- S ▸ NP VP
- PP ▸ IN NP
- NP ▸ DET NP
- NP ▸ NP PP
- VP ▸ VBD NP
- VP ▸ VP PP
- NP ▸ PRP$ NP
I shot an elephant in my pajamas
I shot an elephant in my pajamas

<table>
<thead>
<tr>
<th>NP / PRP</th>
<th>Ø</th>
<th>Ø</th>
<th>S</th>
<th>Ø</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD</td>
<td>Ø</td>
<td>Ø</td>
<td>VP</td>
<td>Ø</td>
</tr>
<tr>
<td>DET</td>
<td>NP</td>
<td>Ø</td>
<td>Ø</td>
<td>NP1 / NP2</td>
</tr>
<tr>
<td>NP</td>
<td>Ø</td>
<td>Ø</td>
<td>NP</td>
<td></td>
</tr>
<tr>
<td>IN</td>
<td>Ø</td>
<td>PP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRP$</td>
<td>NP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
I shot an elephant in my pajamas
I shot an elephant in my pajamas

<table>
<thead>
<tr>
<th>NP / PRP</th>
<th>Ø</th>
<th>Ø</th>
<th>S</th>
<th>Ø</th>
<th>Ø</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD</td>
<td>Ø</td>
<td>VP</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
</tr>
<tr>
<td>DET</td>
<td>NP</td>
<td>Ø</td>
<td>Ø</td>
<td>NP</td>
<td>NP</td>
</tr>
<tr>
<td>NP</td>
<td>Ø</td>
<td>Ø</td>
<td>NP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IN</td>
<td>Ø</td>
<td>PP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRP$</td>
<td>NP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- S → NP VP
- PP → IN NP
- NP → DET NP
- NP → NP PP
- VP → VBD NP
- VP → VP PP
- NP → PRP$ NP
I shot an elephant in my pajamas

<table>
<thead>
<tr>
<th>NP / PRP</th>
<th>Ø</th>
<th>Ø</th>
<th>S</th>
<th>Ø</th>
<th>Ø</th>
<th>VP₁ / VP₂ / VP₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD</td>
<td>Ø</td>
<td>VP</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td></td>
</tr>
<tr>
<td>DET</td>
<td>NP</td>
<td>Ø</td>
<td>Ø</td>
<td>NP</td>
<td>NP</td>
<td>NP₁ / NP₂</td>
</tr>
<tr>
<td>NP</td>
<td>Ø</td>
<td>Ø</td>
<td></td>
<td></td>
<td></td>
<td>NP</td>
</tr>
<tr>
<td>PP</td>
<td>IN</td>
<td>Ø</td>
<td>PP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td>Ø</td>
<td>Ø</td>
<td></td>
<td></td>
<td></td>
<td>PRP$</td>
</tr>
<tr>
<td>VP</td>
<td>VBD</td>
<td>NP</td>
<td></td>
<td></td>
<td></td>
<td>NP</td>
</tr>
<tr>
<td>VP</td>
<td>VP</td>
<td>PP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td>PRP$</td>
<td>NP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- S ➔ NP VP
- PP ➔ IN NP
- NP ➔ DET NP
- NP ➔ NP PP
- VP ➔ VBD NP
- VP ➔ VP PP
- NP ➔ PRP$ NP
I shot an elephant in my pajamas!

### Syntax Tree

<table>
<thead>
<tr>
<th>NP / PRP</th>
<th>Ø</th>
<th>Ø</th>
<th>S</th>
<th>Ø</th>
<th>Ø</th>
<th>VP1 / VP2 / VP3</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD</td>
<td>Ø</td>
<td>VP</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td></td>
</tr>
<tr>
<td>DET</td>
<td>NP</td>
<td>Ø</td>
<td>Ø</td>
<td>NP</td>
<td>NP</td>
<td>NP1 / NP2</td>
</tr>
<tr>
<td>NP</td>
<td>Ø</td>
<td>Ø</td>
<td>NP</td>
<td>IN</td>
<td>Ø</td>
<td>PP</td>
</tr>
<tr>
<td>IN</td>
<td>Ø</td>
<td>Ø</td>
<td>PP</td>
<td>PRP$</td>
<td>NP</td>
<td></td>
</tr>
<tr>
<td>PRP$</td>
<td>NP</td>
<td>NP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>finally, the root!</td>
</tr>
</tbody>
</table>

- S ➔ NP VP
- PP ➔ IN NP
- NP ➔ DET NP
- NP ➔ NP PP
- VP ➔ VBD NP
- VP ➔ VP PP
- NP ➔ PRP$ NP
I shot an elephant in my pajamas

<table>
<thead>
<tr>
<th>NP / PRP</th>
<th>Ø</th>
<th>Ø</th>
<th>S</th>
<th>Ø</th>
<th>Ø</th>
<th>VP1 / VP2 / VP3</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD</td>
<td>Ø</td>
<td>VP</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td></td>
</tr>
<tr>
<td>DET</td>
<td>NP</td>
<td>Ø</td>
<td>Ø</td>
<td>NP</td>
<td>NP1 / NP2</td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td>Ø</td>
<td>Ø</td>
<td>NP</td>
<td>IN</td>
<td>Ø</td>
<td>PP</td>
</tr>
<tr>
<td>NP</td>
<td>Ø</td>
<td>Ø</td>
<td>NP</td>
<td>PRP$</td>
<td>NP</td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td>Ø</td>
<td>Ø</td>
<td>NP</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

finally, the root!

- S ➔ NP VP
- PP ➔ IN NP
- NP ➔ DET NP
- NP ➔ NP PP
- VP ➔ VBD NP
- VP ➔ VP PP
- NP ➔ PRP$ NP

S ➔ NP VP1
S ➔ NP VP2
S ➔ NP VP3
I shot an elephant in my pajamas

<table>
<thead>
<tr>
<th></th>
<th>S</th>
<th>VP</th>
<th>NP</th>
<th>PP</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1 / S2 / S3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VP1 / VP2 / VP3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Finally, the root!

- S ▸ NP VP
- PP ▸ IN NP
- NP ▸ DET NP
- NP ▸ NP PP
- VP ▸ VBD NP
- VP ▸ VP PP
- NP ▸ PRP$ NP

S > NP VP1
S > NP VP2
S > NP VP3

Three valid parses!
how do we recover the full derivation of the valid parses $S_1 / S_2 / S_3$?
CKY runtime?

three nested loops, each $O(n)$ where $n$ is # words

$O(n^3)$
how to find best parse?

- use PCFG (*probabilistic* CFG): same as CFG except each rule $A \rightarrow \beta$ in the grammar is associated with a probability $p(\beta | A)$

- can compute probability of a parse $T$ by just multiplying rule probabilities of the rules $r$ that make up $T$

$$p(T) = \prod_{r \in T} p(\beta_r | A_r)$$
• S ▶ NP VP, 0.4
• PP ▶ IN NP, 0.1
• NP ▶ DET NP, 0.3
• NP ▶ NP PP, 0.1
• VP ▶ VBD NP, 0.2
• VP ▶ VP PP, 0.3
• NP ▶ PRP$ NP, 0.5

• DET ▶ “an”, 0.9
• VBD ▶ “shot”, 0.3
• NP ▶ “pajamas”, 0.8
• NP ▶ “elephant”, 0.9
• NP ▶ “I”, 0.2
• PRP ▶ “I”, 0.6
• IN ▶ “in”, 0.9
• PRP$ ▶ “my”, 0.8
```
I shot an elephant in my pajamas
```

<table>
<thead>
<tr>
<th>NP (0.2) / PRP (0.6)</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD (0.3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DET (0.9)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP (0.8)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IN (0.9)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRP$ (0.8)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP (0.8)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

fill in first level (words) with possible derivations and probabilities
I shot an elephant in my pajamas

how do we compute this cell’s probability?
I shot an elephant in my pajamas

<table>
<thead>
<tr>
<th>NP (0.2) / PRP (0.6)</th>
<th>Ø</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD (0.3)</td>
<td>Ø</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DET (0.9)</td>
<td>NP (0.22)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP (0.8)</td>
<td>Ø</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IN (0.9)</td>
<td>Ø</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRP$ (0.8)</td>
<td>NP (0.32)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP (0.8)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

how do we compute this cell’s probability?

\[
p(\text{DET NP} | \text{NP}) * P(\text{cell}_{\text{DET}}) * P(\text{cell}_{\text{NP}}) = 0.3 * 0.9 * 0.8 = 0.22
\]
I shot an elephant in my pajamas

<table>
<thead>
<tr>
<th></th>
<th>Ø</th>
<th>Ø</th>
<th>S (-6.8)</th>
<th>Ø</th>
<th>Ø</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP (-1.6) / PRP (-0.51)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VBD (-1.2)</td>
<td>Ø</td>
<td>Ø</td>
<td>VP (-4.3)</td>
<td>Ø</td>
<td>Ø</td>
</tr>
<tr>
<td>DET (-0.11)</td>
<td>NP (-1.5)</td>
<td>Ø</td>
<td>Ø</td>
<td>NP</td>
<td>NP1 / NP2</td>
</tr>
<tr>
<td>NP (-0.22)</td>
<td></td>
<td>Ø</td>
<td>Ø</td>
<td>NP (-6.0)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IN (-0.11)</td>
<td>Ø</td>
<td>PP (-3.5)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PRP$ (-0.22)</td>
<td>NP (-1.1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NP (-0.22)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

let’s switch to log space and fill out the table some more
I shot an elephant in my pajamas

<table>
<thead>
<tr>
<th>NP (-1.6) / PRP (-0.51)</th>
<th>Ø</th>
<th>Ø</th>
<th>S (-6.8)</th>
<th>Ø</th>
<th>Ø</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD (-1.2)</td>
<td>Ø</td>
<td>Ø</td>
<td>VP (-4.3)</td>
<td>Ø</td>
<td>Ø</td>
</tr>
<tr>
<td>DET (-0.11)</td>
<td>NP (-1.5)</td>
<td>Ø</td>
<td>Ø</td>
<td>NP₁ / NP₂</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NP (-0.22)</td>
<td>Ø</td>
<td>Ø</td>
<td>NP (-6.0)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IN (-0.11)</td>
<td>Ø</td>
<td>PP (-3.5)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PRP$ (-0.22)</td>
<td>NP (-1.1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NP (-0.22)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ p(NP₁) = ? \]
\[ p(NP₂) = ? \]
do we have to store both NPs?
I shot an elephant in my pajamas

\[
p(VP_1) = ? \\
p(VP_2) = ?
\]
<table>
<thead>
<tr>
<th></th>
<th>VBD (-1.2)</th>
<th>VP (-4.3)</th>
<th>S (-6.8)</th>
<th></th>
<th>VP1 (-10.1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DET</td>
<td>NP (-1.5)</td>
<td>NP (-1.5)</td>
<td>Ø</td>
<td>Ø</td>
<td>VP2 (-9.0)</td>
</tr>
<tr>
<td>NP</td>
<td>IN (-0.11)</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>NP (-7.3)</td>
</tr>
<tr>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>NP (-6.0)</td>
</tr>
</tbody>
</table>

**do we need to store both VPs?**

I shot an elephant in my pajamas
I shot an elephant in my pajamas

<table>
<thead>
<tr>
<th></th>
<th>NP (-1.6) / PRP (-0.51)</th>
<th>Ø</th>
<th>Ø</th>
<th>S (-6.8)</th>
<th>Ø</th>
<th>Ø</th>
<th>S (-11.5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD (-1.2)</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>VP (-4.3)</td>
<td>Ø</td>
<td>Ø</td>
<td>VP (-9.0)</td>
</tr>
<tr>
<td>DET (-0.11)</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>NP (-1.5)</td>
<td>Ø</td>
<td>Ø</td>
<td>NP (-7.3)</td>
</tr>
<tr>
<td>NP (-0.22)</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>NP (-6.0)</td>
<td>Ø</td>
<td>Ø</td>
<td>PP (-3.5)</td>
</tr>
<tr>
<td>IN (-0.11)</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>PRP$ (-0.22)</td>
<td>Ø</td>
<td>Ø</td>
<td>NP (-1.1)</td>
</tr>
<tr>
<td>NP (-0.22)</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>PRP$ (-0.22)</td>
<td>Ø</td>
<td>Ø</td>
<td>NP (-0.22)</td>
</tr>
</tbody>
</table>
I shot an elephant in my pajamas

recover most probable parse by looking at back pointers
issues w/ PCFGs

- independence assumption: each rule’s probability is independent of the rest of the tree!!!
- doesn’t take into account location in the tree or what words are involved (for A>B+C)
  - John saw the man with the hat
  - John saw the moon with the telescope
add more info to PCFG!

- **How to make good attachment decisions?**
  - Enrich PCFG with
    - parent information: what’s above me?
    - lexical information via head rules
      - VP[fight]: a VP headed by “fight”
    - (or better, word/phrase embedding-based generalizations: e.g. recurrent neural network grammars (RNNGs))
Lexicalization

any issues with doing this?
where do we get the PCFG probabilities?

• given a treebank, we can just compute the MLE estimate by counting and normalizing

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

• without a treebank, we can use the inside-outside algorithm to estimate probabilities by
  1. randomly initializing probabilities
  2. computing parses
  3. computing expected counts for rules
  4. re-estimate probabilities
  5. repeat!