Grammars and Parsing

Allen’s Chapters 3,
Jurafski & Martin’s Chapters 8-9
Syntax

• Why is the structure of language (syntax) important?
• How do we represent syntax?
• What does an example grammar for English look like?
• What strategies exist to find the structure in natural language?
Syntax shows the role of words in a sentence.

John hit Sue vs Sue hit John

Here knowing the subject allows us to know what is going on.
Syntax shows how words are related in a sentence.

Visiting aunts ARE boring.

vs

Visiting aunts IS boring.

Subject verb agreement allows us to disambiguate here.
Syntax shows how words are related between sentences.

(a) Italy was beating England. Germany too.
(b) Italy was being beaten by England. Germany too.

Here missing parts of a sentence does not allow us to understand the second sentence.

But syntax allows us to see what is missing.
But syntax alone is not enough

Visiting museums can be boring

This is not ambiguous for us, as we know there is no such thing as a "visiting museum", but syntax cannot show this to a computer.

Compare with…
Visiting aunts can be boring
How do we represent syntax?

Parse Tree:

- Sentence
  - Noun phrase
    - Proper Noun: Sue
  - Verb phrase
    - Verb: hit
    - Noun phrase
      - Proper Noun: John
An example:

- Parsing sentence:
  
  "They are cooking apples."
Parse 1

Sentence

Noun phrase
  Pronoun
    They

Verb phrase
  Verb
    are
  Adjective
    cooking
  Noun
    apples
Parse 2

Sentence

Noun phrase

Pronoun

They

Aux verb phrase

Aux

are

Verb

cooking

Noun phrase

Noun

apples
How do we represent syntax?

List

Sue hit John

[ s, [np, [proper_noun, Sue] ] ,
[vp, [v, hit],
[np, [proper_noun, John] ] ]
Chomsky Hierarchy

0 Unrestricted

\[ \alpha A \beta \rightarrow \alpha \gamma \beta \]

1 Context-Sensitive

\[ \mid \text{LHS} \mid \leq \mid \text{RHS} \mid \]

2 Context-Free

\[ \mid \text{LHS} \mid = 1 \]

3 Regular

\[ \mid \text{RHS} \mid = 1 \text{ or } 2, \ A \rightarrow a \mid aB, \text{ or} \]

\[ A \rightarrow a \mid Ba \]
What Makes a Good Grammar?

- Generality
- Selectivity
- Understandability
What strategies exist for trying to find the structure in natural language?

Top Down vs. Bottom Up

Bottom - Up
John, hit, the, cat
prpn, hit, the, cat
np, hit, the, cat
np, v, the, cat
np, v, det, cat
np, v, det, n
np, v, np
np, vp
s

Top - Down
s
s -> np, vp
s -> prpn, vp
s -> John, vp
s -> John, v, np
s -> John, hit, np
s -> John, hit, det, n
s -> John, hit, the, n
s -> John, hit, the, cat
What strategies exist for trying to find the structure in natural language?

Top Down vs. Bottom Up

Bottom - Up
John, hit, the, cat
prpn, hit, the, cat
np, hit, the, cat
np, v, the, cat
np, v, det, cat
np, v, det, n
np, v, np
np, vp
s
Better if many alternative rules for a phrase
Worse if many alternative terminal symbols for each word

Top - Down
s
s -> np, vp
s -> prpn, vp
s -> John, vp
s -> John, v, np
s -> John, hit, np
s -> John, hit, det,n
s -> John, hit, the,n
s -> John, hit, the,cat
Better if many alternative terminal symbols for each word
Worse if many alternative rules for a phrase
What does an example grammar for English look like?

• Re-write rules

1. sentence -> noun phrase, verb phrase
2. noun phrase -> art, noun
3. noun phrase -> art, adj, noun
4. verb phrase -> verb
5. verb phrase -> verb, noun phrase
# Top down parsing

1. The dog cried

<table>
<thead>
<tr>
<th>Step</th>
<th>Current state</th>
<th>Backup States</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>((S) 1)</td>
<td></td>
<td>initial position</td>
</tr>
<tr>
<td>2</td>
<td>((NP VP) 1)</td>
<td></td>
<td>Rule 1</td>
</tr>
<tr>
<td>3</td>
<td>((ART N VP) 1)</td>
<td></td>
<td>Rules 2 &amp; 3</td>
</tr>
<tr>
<td></td>
<td>((ART ADJ N VP) 1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>((N VP) 2)</td>
<td></td>
<td>Match Art with the</td>
</tr>
<tr>
<td></td>
<td>((ART ADJ N VP) 1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>((VP) 3)</td>
<td></td>
<td>Match N with dog</td>
</tr>
<tr>
<td></td>
<td>((ART ADJ N VP) 1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>((V) 3)</td>
<td></td>
<td>Rules 4 &amp; 5</td>
</tr>
<tr>
<td></td>
<td>((V NP) 3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>((ART ADJ N VP) 1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td>Success</td>
</tr>
</tbody>
</table>
# An Example of Top-Down Parsing

1. The 2 old 3 man 4 cried 5

<table>
<thead>
<tr>
<th>Step</th>
<th>Current State</th>
<th>Backup States</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>((S) 1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>((NP VP) 1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>((ART N VP) 1)</td>
<td>((ART ADJ N VP) 1)</td>
<td>S rewritten to NP VP</td>
</tr>
<tr>
<td>4.</td>
<td>((N VP) 2)</td>
<td>((ART ADJ N VP) 1)</td>
<td>NP rewritten producing two new states</td>
</tr>
<tr>
<td>5.</td>
<td>((VP) 3)</td>
<td>((ART ADJ N VP) 1)</td>
<td>the backup state remains</td>
</tr>
<tr>
<td>6.</td>
<td>((V) 3)</td>
<td>((V NP) 3) ((ART ADJ N VP) 1)</td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>(((C) 4)</td>
<td>((V NP) 3) ((ART ADJ N VP) 1)</td>
<td>the first backup is chosen</td>
</tr>
<tr>
<td>8.</td>
<td>((V NP) 3)</td>
<td>((ART ADJ N VP) 1)</td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>((NP) 4)</td>
<td>((ART ADJ N VP) 1)</td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td>((ART N) 4)</td>
<td>((ART ADJ N VP) 1)</td>
<td>looking for ART at 4 fails</td>
</tr>
<tr>
<td>11.</td>
<td>((ART ADJ N) 4)</td>
<td>((ART ADJ N) 4) ((ART ADJ N VP) 1)</td>
<td>fails again</td>
</tr>
<tr>
<td>12.</td>
<td>((ART ADJ N VP) 1)</td>
<td>((ART ADJ N VP) 1)</td>
<td>now exploring backup state saved in step 3</td>
</tr>
<tr>
<td>13.</td>
<td>((ADJ N VP) 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14.</td>
<td>((N VP) 3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15.</td>
<td>((VP) 4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16.</td>
<td>((V) 4)</td>
<td>((V NP) 4)</td>
<td></td>
</tr>
<tr>
<td>17.</td>
<td>(((C) 5)</td>
<td></td>
<td>success!</td>
</tr>
</tbody>
</table>

*Figure 3.6* A top-down parse of *The 2 old 3 man 4 cried 5*
Depth First Search versus Breadth First

Figure 3.7 Search tree for two parse strategies (depth-first strategy on left; breadth-first on right)