Parsing

- Task: build the syntactic tree for a sentence
- Grammar formalism
  - phrase structure grammar
  - context-free grammar
- Parsing algorithms: CYK parsing, Early parsing
- Open problems
  - where do we get the grammar from?
  - how do we resolve ambiguities?

Penn Treebank

- Penn treebank: English sentences annotated with syntax trees
  - built at the University of Pennsylvania
  - 40,000 sentences, about a million words
  - real text from the Wall Street Journal
- Similar treebanks exist for other languages
  - German
  - French
  - Spanish
  - Arabic
  - Chinese
  - etc.

Sample Syntax Tree

```
Mr Vinken
<!>
NP-SBJ
is
chairman
NP
of
Elsevier N.V.

PPPP
NP ,
the Dutch publishing group

NP
@@
PP
XXXXX
NP-PRD
ee
VP .
XXXXX
S
```
Sample Tree with Part-of-Speech

Learning a Grammar from the Treebank

• Context-free grammar: we have rules in the form
  \[ S \rightarrow \text{NP-SBJ VP} \]

• We can collect these rules from the treebank

• We can even estimate probabilities for rules
  \[ p(S \rightarrow \text{NP-SBJ VP}|S) = \frac{\text{count}(S \rightarrow \text{NP-SBJ VP})}{\text{count}(S)} \]

  ⇒ Probabilistic context-free grammar (PCFG)

Rules Applications to Build Tree

Compute Probability of Tree

• Probability of a tree is the product of the probabilities of the rule applications:
  \[ p(\text{tree}) = \prod_i p(\text{rule}_i) \]

• We assume that all rule applications are independent of each other
  \[ p(\text{tree}) = p(S \rightarrow \text{NP-SBJ VP}|S) \times p(\text{NP-SBJ} \rightarrow \text{NNP NNP}|\text{NP-SBJ}) \times \ldots \times p(\text{NNP} \rightarrow \text{Elsevier}|\text{NNP}) \]
Prepositional Phrase Attachment Ambiguity

Rule Applications

Difference in Probability

Scope Ambiguity

• PP attachment to NP-PRD is preferred if

\[ p(VP \rightarrow VBZ \ NP-PRD|VP) \times p(NP-PRD \rightarrow NP \ PP|NP-PRD) \]

is larger than

\[ p(VP \rightarrow VBZ \ NP-PRD \ PP|VP) \times p(NP-PRD \rightarrow NP|NP-PRD) \]

• Is this too general?

However: the same rules are applied
Weakness of PCFG

- Independence assumption too strong
- Non-terminal rule applications do not use lexical information
- Not sufficiently sensitive to structural differences beyond parent/child node relationships

Head Words

- Recall dependency structure:
  
  Mr
  Vinken
  of
  Elsevier
  chairman

  - Direct relationships between words, some are the head of others
    (see also Head-Driven Phrase Structure Grammar)

Adding Head Words to Trees

- Each context-free rule has one head child that is the head of the rule
  - S → NP VP
  - VP → VBZ NP
  - NP → DT NN NN

- Parent receives head word from head child
- Head childs are not marked in the Penn treebank, but they are easy to recover using simple rules
Recovering heads

- Rule for recovering heads for NPs
  - if rule contains NN, NNS or NNP, choose rightmost NN, NNS or NNP
  - else if rule contains a NP, choose leftmost NP
  - else if rule contains a JJ, choose rightmost JJ
  - else if rule contains a CD, choose rightmost CD
  - else choose rightmost child
- Examples
  - NP \(\rightarrow\) DT NNP NN
  - NP \(\rightarrow\) NP CC NP
  - NP \(\rightarrow\) NP PP
  - NP \(\rightarrow\) DT JJ
  - NP \(\rightarrow\) DT

Using head nodes

- PP attachment to NP-PRD is preferred if

\[
p\left(\text{VP(is) } \rightarrow \text{VBZ(is) NP-PRD(chairman) VP(is)}\right) \\
\times p\left(\text{NP-PRD(chairman) } \rightarrow \text{NP(chairman) PP(Elsevier) NP-PRD(chairman)}\right)
\]

is larger than

\[
p\left(\text{VP(is) } \rightarrow \text{VBZ(is) NP-PRD(chairman) PP(Elsevier) VP(is)}\right) \\
\times p\left(\text{NP-PRD(chairman) } \rightarrow \text{NP(chairman) NP-PRD(chairman)}\right)
\]

- Scope ambiguity: combining Hoboken and Jim should have low probability

\[
p\left(\text{NP(Hoboken) } \rightarrow \text{NP(Hoboken) CC(and) NP(John)}\right) \times p\left(\text{NP(John)} \rightarrow \text{NP(Hoboken) VP(Hoboken)}\right)
\]

Sparse data concerns

- How often will we encounter

\[
\text{NP(Hoboken) } \rightarrow \text{NP(Hoboken) CC(and) NP(John)}
\]

- ... or even

\[
\text{NP(Jim) } \rightarrow \text{NP(Jim) CC(and) NP(John)}
\]

- If not seen in training, probability will be zero

Sparse data: Dependency relations

- Instead of using a complex rule

\[
\text{NP(Jim) } \rightarrow \text{NP(Jim) CC(and) NP(John)}
\]

- ... we collect statistics over dependency relations

<table>
<thead>
<tr>
<th>head word</th>
<th>head tag</th>
<th>child node</th>
<th>child tag</th>
<th>direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jim</td>
<td>NP</td>
<td>and</td>
<td>CC</td>
<td>left</td>
</tr>
<tr>
<td>Jim</td>
<td>NP</td>
<td>John</td>
<td>NP</td>
<td>left</td>
</tr>
</tbody>
</table>

- first generate child tag: \(p(\text{CC|NP,Jim,left})\)
- then generate child word: \(p(\text{and|NP,Jim,left,CC})\)
Sparse Data: Interpolation

- Use of interpolation with back-off statistics (recall: language modeling)
- Generate child tag
  \[ p(\text{NP}, \text{Jim}, \text{left}) = \lambda_1 \frac{\text{count}(\text{CC, NP, Jim, left})}{\text{count}(\text{NP, Jim, left})} + \lambda_2 \frac{\text{count}(\text{CC, NP, left})}{\text{count}(\text{NP, left})} \]
  With \( 0 \leq \lambda_1 \leq 1, \ 0 \leq \lambda_2 \leq 1, \ \lambda_1 + \lambda_2 = 1 \)

What also helps

- Adding a count for distance from head word
- Part-of-speech of the head word and the child word also useful
- Improving tags
  - instead of general VB, distinguish between intransitive verb phrases Vi, and transitive verb phrases Vt
  - distinguish between complements (required attachments, e.g. object of a transitive verb) and adjuncts (optional attachments, e.g. yesterday)
- Not only use parent tag, but also grand-parent tag
- Create n-best list of best parse trees, re-score

Parsing algorithm

- Efficient parsing algorithm is tricky
- Algorithm is similar to chart parsing, as presented
- Impossible to search entire space of possible parse trees
  \[ \rightarrow \text{rest cost estimation, pruning} \]
Performance

- Performance typically measured in recall/precision of dependency relations
  - PCFG: 74.8%/70.6%
  - using lexical dependencies: 85.7%/85.3%
  - latest models (Collins): 89.0%/88.7%

- Core sentence structure (complements, NP chunks) recovered with over 90% accuracy

- Attachment ambiguities involving adjuncts are resolved with much lower accuracy (~80% for PP attachment, ~50-60% for coordination)

Note: numbers quoted from lecture 4 Parsing and Syntax II of MIT class 6.891 Natural Language Processing by Michael Collins (2005)