Natural Language Processing

Parsing
Parsing is the automatic analysis of a sentence with respect to its syntactic structure. Given a CFG, this means deriving a phrase structure tree assigned to the sentence by the grammar. With ambiguous grammars, each sentence may have many valid parse trees:

• Should we retrieve all of them or just one?
• If the latter, how do we know which one?
I booked a flight from LA.

• This sentence is ambiguous. In what way?

• What should happen if we parse the sentence?
I booked a flight from LA.
I booked a flight from LA.
Combinatorial explosion

Ambiguity

- linear
- cubic
- exponential
Phrase structure trees

```
S
  NP  VP
    Pro Verb NP
      I prefer Nom
        Det a Noun
          Nom flight
            Noun morning
```
Basic concepts of parsing

• Two problems for grammar $G$ and string $w$:
  • **Recognition**: determine if $G$ accepts $w$
  • **Parsing**: retrieve (all or some) parse trees assigned to $w$ by $G$

• Two basic search strategies:
  • **Top-down**: start at the root of the tree
  • **Bottom-up**: start at the leaves
Top-down parsing

• Basic idea
  • Start at the root node, expand tree by matching the left-hand side of rules
  • Derive a tree whose leaves match the input

• Potential problems:
  • Uses rules that could never match the input
  • May loop on recursive rules: VP → VP PP
Bottom-up parsing

- Basic idea:
  - Start with the leaves, build tree by matching the right-hand side of rules
  - Build a tree with S at the root

- Potential problems
  - Builds structures that could never be in a tree
  - May loop on epsilon productions: \( \text{NP} \rightarrow \varepsilon \)
Dealing with ambiguity

- The number of possible parse trees grows exponentially with sentence length
- A naive backtracking approach is too inefficient
- Key observation:
  - Alternative parse trees share substructures
  - We can use dynamic programming (again)
Probabilistic context-free grammar

• The number of possible parse trees grows rapidly with the length of the input.
• But not all parse trees are equally useful.

  *Example*: I booked a flight from Los Angeles.

• In many applications, we want the ‘best’ parse tree, or the first few best trees.
• Special case: ‘best’ = ‘most probable’
A probabilistic context-free grammar (PCFG) is a context-free grammar where

- each rule $r$ has been assigned a probability $p(r)$ between 0 and 1
- the probabilities of rules with the same left-hand side sum up to 1
### Probabilistic context-free grammar

#### A sample PCFG

<table>
<thead>
<tr>
<th>Rule</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S \rightarrow NP\ VP$</td>
<td>1</td>
</tr>
<tr>
<td>$NP \rightarrow$ Pronoun</td>
<td>$1/3$</td>
</tr>
<tr>
<td>$NP \rightarrow$ Proper-Noun</td>
<td>$1/3$</td>
</tr>
<tr>
<td>$NP \rightarrow$ Det Nominal</td>
<td>$1/3$</td>
</tr>
<tr>
<td>$Nominal \rightarrow$ Nominal PP</td>
<td>$1/3$</td>
</tr>
<tr>
<td>$Nominal \rightarrow$ Noun</td>
<td>$2/3$</td>
</tr>
<tr>
<td>$VP \rightarrow$ Verb NP</td>
<td>$8/9$</td>
</tr>
<tr>
<td>$VP \rightarrow$ Verb NP PP</td>
<td>$1/9$</td>
</tr>
<tr>
<td>$PP \rightarrow$ Preposition NP</td>
<td>1</td>
</tr>
</tbody>
</table>
The probability of a parse tree is defined as the product of the probabilities of the rules that have been used to build the parse tree.
The probability of a parse tree

```
S 1/1
 /  
NP 1/3  VP 8/9
   /  
  Pro  Verb  NP 1/3
      /  
     I  booked  Det
          /  
         a  Nom 1/3
            /  
           Noun  PP
                 /  
                  from LA
```

Probability: $\frac{16}{729}$
Probabilistic context-free grammar

The probability of a parse tree

Probability: 6/729
Independence assumption

How can we make sense of this in terms of probability theory?

• The probability of a rule expansion is dependent only on the left-hand side symbol

• Is this a reasonable independence assumption?