Treebanks

- Treebanks are corpora in which each sentence has been paired with a parse tree (presumably the right one).
- These are generally created
  - By first parsing the collection with an automatic parser
  - And then having human annotators correct each parse as necessary.
- This generally requires detailed annotation guidelines that provide a POS tagset, a grammar and instructions for how to deal with particular grammatical constructions.

Penn Treebank

- Penn TreeBank is a widely used treebank.
- Most well known is the Wall Street Journal section of the Penn TreeBank.
Treebank Grammars

- Treebanks implicitly define a grammar for the language covered in the treebank.
- Simply take the local rules that make up the sub-trees in all the trees in the collection and you have a grammar.
- Not complete, but if you have decent size corpus, you’ll have a grammar with decent coverage.

Treebank Grammars

- Such grammars tend to be very flat due to the fact that they tend to avoid recursion.
  - To ease the annotators burden
- For example, the Penn Treebank has 4500 different rules for VPs. Among them...
  - \[ VP \rightarrow VBD \ PP \]
  - \[ VP \rightarrow VBD \ PP \ PP \]
  - \[ VP \rightarrow VBD \ PP \ PP \ PP \]
  - \[ VP \rightarrow VBD \ PP \ PP \ PP \ PP \]
Heads in Trees

- Finding heads in treebank trees is a task that arises frequently in many applications.
  - Particularly important in statistical parsing
- We can visualize this task by annotating the nodes of a parse tree with the heads of each corresponding node.

Lexically Decorated Tree

```
S(dumped)
   NP(workers)  VP(dumped)
     NNS(workers)  VBD(dumped)  NP(sacks)  PP(into)
        NNS(sacks)  P(into)  NP(bin)
                  DT(a)  NN(bin)
```
Head Finding

- The standard way to do head finding is to use a simple set of tree traversal rules specific to each non-terminal in the grammar.

Noun Phrases
Treebank Uses

- Treebanks (and headfinding) are particularly critical to the development of statistical parsers
  - More later

Dependency Grammars

- In CFG-style phrase-structure grammars the main focus is on constituents.
- But it turns out you can get a lot done with just binary relations among the words in an utterance.
- In a dependency grammar framework, a parse is a tree where
  - the nodes stand for the words in an utterance
  - The links between the words represent dependency relations between pairs of words.
    - Relations may be typed (labeled), or not.
Summary

- Context-free grammars can be used to model various facts about the syntax of a language.
- When paired with parsers, such grammars constitute a critical component in many applications.
- Constituency is a key phenomena easily captured with CFG rules.
  - But agreement and subcategorization do pose significant problems
- Treebanks pair sentences in corpus with their corresponding trees.