Treebanks, Treebanking and Evaluation
Phrase Structure Tree

- Example:

```
((DaimlerChrysler’s shares)_{NP} (rose (three eights)_{NUMP} (to 22)_{PP-NUM})_{VP})_{S}
```

DaimlerChrysler’s shares rose three eights to 22
Dependency Tree

- Example:

\[
\text{rose}_{\text{Pred}}(\text{shares}_{\text{Sb}}(\text{DaimlerChrysler'sAtr}), \text{eights}_{\text{Adv}}(\text{threeAtr}), \text{to}_{\text{AuxP}}(22_{\text{Adv}}))
\]
Parser Development

• Use training data for learning phase
  – segment as needed (e.g., for heldout)
  – use all for
    • manually written rules (seldom today)
    • automatically learned rules/statistics

• Occasionally, test progress on Development Test Set
  – (simulates real-world data)

• When done, test on Evaluation Test Set

• Unbreakable Rule #1: Never look at Evaluation Test Data (not even indirectly, e.g. performance numbers)
Evaluation

- Evaluation of parsers (regardless of whether manual-rule-based or automatically learned)
- Repeat: Test against Evaluation Test Data
- Measures:
  - Dependency trees:
    - Dependency Accuracy, Precision, Recall
  - Parse trees:
    - Crossing brackets
    - Labeled precision, recall [F-measure]
Dependency Parser Evaluation

- **Dependency Recall:**
  - \[ R_D = \frac{\text{Correct}(D)}{|S|} \]
  - \( \text{Correct}(D) \): number of correct dependencies
    - correct: word attached to its true head
    - Tree root is correct if marked as root
  - \(|S|\) - size of test data in words (since \(|\text{dependencies}| = |\text{words}|\))

- **Dependency precision (if output not a tree, partial):**
  - \[ P_D = \frac{\text{Correct}(D)}{\text{Generated}(D)} \]
  - \( \text{Generated}(D) \) is the number of dependencies output
    - some words without a link to their head
    - some words with several links to (several different) heads
Phrase Structure (Parse Tree) Evaluation

• Crossing Brackets measure
  – Example “truth” (evaluation test set):
    • ((the ((New York) - based company)) (announced (yesterday)))
  – Parser output - 0 crossing brackets:
    • ((the New York - based company) (announced yesterday))
  – Parser output - 2 crossing brackets:
    • (((the New York) - based) (company (announced (yesterday))))

• Labeled Precision/Recall:
  – Usual computation using bracket labels (phrase markers)
    T: ((Computers)_{NP} (are down)_{VP})_{S} ↔ P: ((Computers)_{NP} (are (down)_{NP})_{VP})_{S}
    • Recall = 100%, Precision = 75%