HW #4

Probabilistic Parsing

- Goals:
  - Learn about PCFGs
  - Implement PCKY
  - Analyze parsing evaluation
  - Assess improvements to PCFG parsing

Tasks

- Train a PCFG
  - Estimate rule probabilities from treebank
    - Treebank is already in CNF
    - More ATIS data from Penn Treebank

- Build PCKY parser
  - Modify (your) existing CKY implementation

Tasks

- Evaluation:
  - Evaluate your parser using standard metric
    - Parseval implemented as ‘evalb’
  - Provided set of ATIS test sentences

- Improvement:
  - Improve your parser in some way:
    - Coverage, accuracy, speed
  - Evaluate your new parser

Improvement Possibilities

- Coverage:
  - Some test sentences won't parse as is!
    - Lexical gaps (aka out-of-vocabulary tokens)
      - Remember to model the probabilities, too
- Better context modeling:
  - E.g. parent annotation
- Better efficiency:
  - E.g. heuristic filtering, beam search

Treebank Format

- Adapted from Penn Treebank Format

  - Rules simplified:
    - Removed traces and other null elements
    - Removed complex tags
    - Reformatted POS tags as non-terminals

Reading the Parses

- POS unary collapse:
  - (NP_NNP Ontario)
    - was
  - (NP (NNP Ontario))

- Binarization:
  - VP → VP_PRIME PP; VP_PRIME → VB PP
    - Was
  - VP → VB PP PP

Notes

- You may use any programming language
    - As long as it runs on the cluster
- You may work in teams on this assignment
    - If you do so, indicate in write-up, describe who did what
- You may use NLTK classes to manipulate trees, rules
    - E.g. nltk.Tree
    - Own code for probability computation, PCKY
- Unparseable sentences
    - Please make sure your parser doesn't crash
        - It's fine to return zero parses for a sentence, though