HW #3

Goals

Through this assignment you will:

- Explore issues in parser design for natural language processing.
- Employ key programming concepts such as dynamic programming to create (relatively) efficient parsing algorithms.
- Improve your understanding of the CKY algorithm through implementation.

NOTE: *You may work in teams of two (2) on this assignment. If you do so:*

- Please include a brief discussion of each teammate's contribution in the readme.

Background

Please review the class slides and readings in the textbook on the Cocke-Kasami-Younger (CKY) algorithm (J&M **13.4.1**).

Implementing a CKY Parser

Based on the material in the lectures and text, develop an implementation of the CKY algorithm that will parse input sentences using a CNF grammar. You may use existing implementations of the ***data structures*** to represent the grammar in NLTK or other NLP toolkits, but you must implement the parsing algorithm yourself.

Your algorithm must return all parses derived for the input sentences given the grammar.

Parsing with your CKY Parser

The program you submit should do the following:

1. Load the CNF grammar.
2. Read in the example sentences.
3. For each example sentence, output to a file:
    - The sentence itself
    - the simple bracketed structure parse(s) based on your implementation of the CKY algorithm, and
    - the number of parses for that sentence.

Programming

Create a program named hw3_parser to perform CKY parsing as described above, invoked as:

hw3_parser.sh <grammar_filename> <test_sentence_filename> <output_filename>

where:

- **<grammar_filename>**
  - is the name of the file holding grammar rules in the NLTK .cfg format in Chomsky Normal Form.
- **<test_sentence_filename>**
  - is the name of the file containing test sentences to parse with your algorithm.
- **<output_filename>**
  - is the name of the file where your system will write the parses and their counts over the test sentences.

Files

Please adhere to the naming conventions.

Test and Example Files

All test and example files will be located in **/dropbox/20-21/571W/hw3/** on the cluster.

- **grammar_cnf.cfg**: Grammar in NLTK format, already in Chomsky Normal Form, to be used by your algorithm to parse the sentences.
- **sentences.txt**: Test sentences to be parsed using your parsing program (hw3_parser.sh).
- **toy.cfg**: Simple grammar in CNF for development.
- **toy_sentences.txt**: Simple set of practice sentences.
- **toy_output.txt**: Example output file.

Submission Files

- **hw.tar.gz**: tarball containing:
  - **hw3_parser.sh**: Shell script to invoke program with necessary extension
  - binaries/scripts called by hw3_parser.sh
  - Source code, if executable required compilation
  - **hw3_output.txt**: Results of running your parser on the test sentences with the corresponding grammar **grammar_cnf.cfg**

    - **hw3.cmd**: Condor file which drives your parsing program (**hw3_parser.sh**) with the relevant grammar, test sentences, and output file.
        - Your CKY parsing program must run on patas using:

          ```
          $ condor_submit hw3.cmd
          ```

          Please see the CLMS wiki pages on the basics of using the condor (Links to an external site.) cluster.

          All files created by the condor run should appear in the top level of the directory.

- **readme.{txt|pdf}**
    - This file should describe and discuss your work on this assignment. Include problems you came across and how (or if) you were able to solve them, any insights, special features, and what you learned. Give examples if possible. If you were not able to complete parts of the project, discuss what you tried and/or what did not work.

If working in teams:

- Please include a brief discussion of each teammate's contribution.
- For the individual group members:
    - One teammate should submit the tarball and readme.
    - The other teammate should submit a .txt file naming the student with whom they worked.