Natural Language Processing - Project 3
CSE 325/425
Prof. Sihong Xie
Due Date: 11:55pm, May 5, 2021

1 Problem Statement

Given a PCFG in CNF, generate and parse a random sentence.

A PCFG Grammar The grammar is given in the file *data/grammar.gr* and you can use any text editor to view its contents. In the file, there are empty lines, comments (starting with #) and productions. Lines 20 - 23 with one comment line, one empty line and two productions look like

```
# Rules for creating full sentences.

5 ROOT S Period
1 ROOT S Excl
```

Each production (e.g., *1 ROOT S Period*) has 3 fields: a number indicating its frequency ("1" in this example. It is NOT the production probability), a left-hand-side (LHS, "*ROOT*" in this case), a **list** of symbols on the right-hand-side (RHS, "*S Period*" in this case). There are three sets of symbols: 22 terminals (all in lower cases, plus the period and exclamation), 7 POS-tags (*Det, Adj, Verb, Prep, Noun, Period, Excl*, only the first characters are in Upper case) and 5 non-terminals (*PP, NP, S, ROOT, VP*, all characters are in Upper case).

Generation Starting from the symbol ROOT in a list, expand any non-terminals and pre-terminals in the list, using one of the proper productions (picked proportionally to its probability), until there is no more non-terminals and pre-terminals remained in the list. This can be done using Depth-First Search (DFS) to recursively generate the child(ren).

Parsing The CYK algorithm can parse the sentence you generated, using the same PCFG. The algorithm is quite simple and the textbook shall give you enough ideas. The data structures Entry, Cell, and the CYK matrix are defined for you already and you need to follow such data structures. Pay attention instead to highly efficient and clear data structures.

2 Exercises

- How can you modify the production frequencies so that longer sentences can be generated? Explain how and why.
- Is every sentence generated by the grammar can be parsed by the CYK algorithm using the same grammar? Give your intuition.

3 Experiment

Download and unzip the zip file (project3.zip) from Piazza.

The list of Python files in the project:

- **problem1.py** It is provided to you to read the PCFG grammar from *data/grammar.gr*.
- **problem2.py** This file defines a generator of sentences from the given PCFG.

  ```
  (ROOT (S (NP (Det the) (Noun president)) (VP (Verb understood) (NP (Det every) (Noun pickle)))) !)
  ```

  This output is redirected to the file *data/sentences.txt*

  ```
  python problem2.py  > ../data/sentences.txt
  ```

  The file is then read and piped into a perl program (*src/prettyprint.pl*) by typing on the command line

  ```
  cat data/sentences.txt | src/prettyprint.pl
  ```

  to print the parse tree(s).

- **problem3.py** This file defines a CYK parser for the given PCFG. The parser will read the file *data/sentences.txt* and parse the sentences there. The output is redirected to the file *data/parses.txt*

  ```
  python problem3.py  > ../data/parses.txt
  ```

  You can verify the parses by the prettyprint.pl script as in problem2.

  Note that you may (or may not) get more than one tree for each sentence. One and only tree needs to be output for each sentence.

4 Deliverables

Implement the functions that has the section saying "INSERT YOUR CODE HERE". After you implemented and tested your codes, zip the folder "project_3" to "project_3_<YOUR_LIN>.zip". Submit the zip file to Coursesite under "Proejct 3".

5 Grading

The points for each function is printed when you run the unit tests using nosetests. The total is 100 points.