Task #3 part 1
Producing a Context-Free Grammar for NLTK

Thuong-Hai Pham

March 17, 2017

1 Task 1

For task 1, we parsed the sentences with Stanford parser [1] [1], then converted these parsed structures to Context-free grammar (CFG) rules in **grammar.py** using **nltk.tree** and **nltk.grammar.CFG**[2]. The output file is **grammar.cfg**.

The Chomsky normal form (CNF) was also achieved in grammar.py paralleling with non-CNF CFG. The functions **tree.chomsky_normal_form()** and **tree.collapse_unary()** allow us to convert non-CNF tree to CNF tree. This can be double checked with **is_chomsky_normal_form()** function of the CFG object. Our CNF CFG is stored in **grammar_cnf.cfg**. The converting process is quite trivial by:

- eliminating chain of unary productions until reaching binary one or terminal

```
NP -> EX -> 'There'
# merging the unary chain to one non-terminal
=> NP+EX -> 'There'
```

- replacing group of nodes with an intermediate non-terminal if the production is tertiary or more

```
NP -> JJ NN NNS
# introducing new non-terminal NP|<NN-NNS>
=> NP -> JJ NP|<NN-NNS>; NP|<NN-NNS> -> NN NNS
```

[1] http://nlp.stanford.edu:8080/parser/index.jsp

It is important to set collapsePOS=True in tree.collapse_unary() to keep reducing until reaching terminals. This is set to false as in some cases, the direct parent of a terminal is POS tag, which should be kept intact. In the same function, parameter collapseRoot must be kept False to prevent collapsing ROOT node and producing grammar with multiple root nodes.

2 Task 2

Parsing raw sentences (task 2) with grammar.cfg was implemented in **parse.py**, in which we initiated **nltk.parse.EarleyChartParser** with loaded grammar and parsed the sentences after being tokenised by **nltk.tokenize.WordPunctTokenizer**. It is important to note that we have to specify the grammar start non-terminal

```
grammar._start = Nonterminal('ROOT')
```

or else the grammar object will be loaded with starting non-terminal assigned by the first rule.

References

[1] D. Klein and C. D. Manning, "Accurate unlexicalized parsing," in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pp. 423–430, Association for Computational Linguistics, 2003.

[2] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.