Producing a Context-Free Grammar for NLTK

LIN2571 Computational Morphology and Syntax
Project Task 3

Claudia Borg

February 28, 2017

1 Background

The aim of this task is to familiarise yourself with grammar construction for English. You will also use a parser from `NLTK` to check the grammar. In general, the use of existing libraries and functions for this assignment is encouraged.

2 Starting Point

You have a set of 20 plain sentences, available through the VLE. These are your starting point.

3 Tasks

Task 1 Obtain the syntactic categories for the sentences, either through the examples in class, or through an online tagger/parser[1]. You are free to use any resource as long as you provide an adequate citation to it. Using these categories, write a CFG that covers the 20 sentences. Save your grammar in a text file called `grammar.cfg`. Below in Section 5 is a sample of what your grammar should look like.

Task 2 Create a python script that uses the `nltk.parse.EarleyChartParser` and does the following:

- load your grammar using `nltk.data.load()`
- Initialise an instance of this parser with your grammar
- Read each sentence and parse it

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

- Print out the simple bracketed structure for each parsed sentence, followed by the number of parses for that sentence. Section 6 contains a sample output. The bracketed trees can be outputted in a single line.
- Print out the average number of parses per input sentence — this will give you an indication of how ambiguous the sentences are with respect to the grammar.

Preparation for Part 2 Covert your grammar into CNF. You will be using it in the next part when we come to implement the CKY algorithm.

Extra Credit Familiarise yourself with the `nltk.grammar` module. How do you access production rules? How can you query if something exist in the grammar? Reading Chapter 8 (up to section 4) of the NLTK book will help, as well as `http://www.nltk.org/howto/parse.html` and `http://www.nltk.org/howto/grammar.html`. In part 2, we will be using the grammar produced here, and we will also use `nltk.grammar` to access the production rules and query the grammar. In your documentation, show which functions from this module you would use. Since examples of these functions will be given in Part 2, in order to gain extra credit you must send me a document detailing the functions that you would use by email by the 7th March.

4 Deliverables and practicalities

As a deliverable, you should submit:

1. The python script
2. The output of your script
3. Your CFG grammar
4. Your CNF grammar
5. A one-pager explaining your approach, sources used, problems encountered and what you did to solve those problems.

Due date: 22nd March - submission via `VLE`. Via email is optional if you want to confirm 100% that you submitted your task on time, especially in the case of technical problems with the `VLE`
Late submission: A 10% deduction from the grade of this task per every late day.
Submission type: ZIP file containing all the above, please name the zip file as: NameSurname.zip
Clarifications: Questions are to be posted on the `VLE` forum and will be answered there.

5 Sample Grammar

```
#a comment
S -> NP VP
S -> VP

NP -> NN
NNS -> ’dogs’ | ’cats’

VP -> VBP
VBP -> ’bark’
```

6 Sample Output

```
(TOP
  (S
    (NP (EX There))
    (VP (VBZ is)
      (NP
        (NP (DT an) (NN egg))
        (PP (IN in)
          (NP (PRP$ my) (NN soup)))))
    (PUNC .)))

1
-------------------------------------
(TOP
  (S
    (NP (EX There))
    (VP (VBZ is)
      (NP
        (NP (DT a) (VB fly))
        (PP (IN in)
          (NP (PRP$ my) (NN soup)))))
    (PUNC .)))

(TOP
  (S
    (NP (EX There))
    (VP (VBZ is))
      (NP
       (NP (DT a) (NN fly))
       (PP (IN in)
         (NP (PRP$ my) (NN soup)))))
    (PUNC .))

2
------------------------------------
There are on average 1.5 parses per sentence.
```