Due date: October 26, 2020

Goal: Develop a context free grammar

Overview: A parse tree is a representation of syntactic structure, on which semantic interpretation can be based. You are to write a context free grammar that covers as much different text, as you can manage, but it must cover the validation data provided below. You may draw inspiration from anywhere, starting from the different CFG grammars in NLTK.

Description:

1. run your previously developed preprocessing pipeline plus a named entity module (NLTK is ok)

2. develop a context free grammar to cover the validation data (additional coverage is always welcome)

3. run the Earley parser in NLTK using your grammar

4. proofread the results, correct if necessary. Comment on strong and weak points in your report

5. on October 25, 2020 at night I will post a short challenge text, that you have to run through your system and submit on October 26 2002

Validation Data   Run your prepipeline successively over the following sentences:

1. *John ate an apple.*

2. *John ate the apple at the table.*

3. *On Monday, John ate the apple in the fridge.*

4. *On Monday, John ate the apple in his office.*

5. *On Monday, John ate refrigerator apple in his office.*

6. *Last week, on Monday, John finally took the apple from the fridge to his office.*

7. *Last Monday, John promised that he will put an apple in the fridge. He will eat it on Tuesday at his desk. It will be crunchy.*

8. *On Monday, September 17, 2018, John O'Malley promised his colleague Mary that he would put a replacement apple in the office fridge. O'Malley intended to share it with her on Tuesday at his desk and anticipated that the crunchy treat would delight them both. But she was sick that day.*

9. *Sue said that on Monday, September 17, 2018, John O'Malley promised his colleague Mary that he would put a replacement apple in the office fridge and that O'Malley intended to share it with her on Tuesday at his desk.*

Interface   Make sure that you have a way to input a new text that is then processed by the entire pipeline and displays all the annotations obtained on the screen without touching the code. This can be as simple as a command line call and pretty print output. Provide an option to save/print. This is helpful for your own development, but essential for testing during your demo.

Deliverables and marking scheme: to be submitted in Moodle before October 26, 2020

- (4pts) the CFG you developed (Attrib 1, 12)

- (1pts) named entity processing (Attrib. 5)

- (1pt) Earley parse (Attrib. 1)

- (1pt) useful output (Attrib. 5)

- (3pts) challenge run (Attrib. 6)

- (2pts) report (Attrib. 6)