CS 585 – Fall 2023 – Homework 5 (100 total points)

Due Monday November 20, 11:59pm

GOALS

- Apply a natural language Context-Free Grammar (CFG) grammar
- Gain hands-on experience with an NLP constituency parser
- Examine variation in syntax structure among English language texts

DATA

For problems 3-6, you will use data from the One Stop English Corpus, available on github: LINK

This is a corpus of aligned texts written at different reading levels: Elementary, Intermediate, and Advanced. It was developed to study how language varies by reading level, to enable both readability assessment and text simplification tasks.

In this assignment, you will only use the Elementary and Advanced texts, and will take a small sample from the folder Texts-Together-OneCSVperFile

TOOLS

In this assignment you will use the Stanza python package created by the Stanford NLP Group. Stanza provides a collection of NLP tools for multiple languages, including a constituency parser for English based on the Penn Treebank.

You may use other open-source Python packages to complete this assignment. You must complete this homework in Python; other languages will not be accepted.

WHAT TO SUBMIT

Please upload or submit the following in Blackboard:

- For Problems 2-4, upload to Blackboard:
  - One Jupyter notebook (.ipynb file) with cell output. Name this file "HW5_[CWID]_[LastName].ipynb"
  - A PDF copy of the exact same notebook (same code and same output)
  - Please do not zip or compress files before uploading files to Blackboard.
- For Problem 1, 5 and 6: Enter your written answers in Blackboard with your HW submission.

PROBLEM 1 – Applying a Context free Grammar (15pts – Answer in Blackboard)

Section 9.2.3 in Eisenstein-NLP (pp213-218) describes a fragment of the Penn Treebank grammar. Note that this grammar fragment uses Penn Treebank Part-of-Speech (POS) tags in production rules. For example, the symbols NN, NNS, NNP and PRP in rule "NP→NN |NNS |NNP |PRP" are POS tags.

- Using section 9.2.3 as a reference, produce a **syntactic parse** of these sentences:
    a) "Lucy plays with friends"
    b) "This movie is careless and unfocused" [This sentence is from the Stanford Sentiment Treebank]
    c) **Two possible parses** for the sentence "She buys a gift with gold"
- You do not need to include punctuation, which is not covered in section 9.2.3.
- Write your output using **labeled nested parenthesis format** of "(<label> <child-1> <child-2> ...). So, for example, the sentences "Pete cooks with onions" would be represented as:

```
(S (NP (NNP Pete)) (VP (VBZ cooks) (PP (IN with) (NP (NNS onions)))))
```

Or with line breaks and indention as:

```
(S (NP (NNP Pete))

   (VP (VBZ cooks)

       (PP (IN with) (NP (NNS onions)))))
```

Note that in this example, NNP, VBZ, IN and NNS are POS tags in the Penn Treebank tagset.

- To complete this problem, you **only** need to submit four parses in parenthesis format. You do not need to submit drawings of parse trees or explanations of your answers. Submit your typed answers on Blackboard.

PROBLEM 2 – Constituency parsing (15pts)

In this problem, you will use the Stanza Constituency parser to check your work from Problem 1. Note that by default, Stanza uses the Penn Treebank model for the English language LINK.

- Check your work for Problem 1 by applying the Stanza constituency parser to the three sentences of Problem 1.
- Show your output in your notebook. Your output should show a sentence parse with constituent labels for each of these three sentences.
- You may go back to Problem 1 and revise your answers if you made mistakes, but you should first try the problems by hand, to become familiar with CFG production rules as used in NLP.
- Note that the Stanza parser outputs one parse for 1(c), "She buys a gift with gold". You will need to identify the other possible parse for this ambiguous sentence in Problem 1.

PROBLEMS 3-6 are based on the One Stop English Corpus; see section DATA above.

PROBLEM 3 – Reading the data (5 pts)

- Read in data the following three files in the folder Texts-Together-OneCSVperFile: (Since parsing is computationally expensive, we will use a very small dataset for this problem)
    - climate change.csv
    - Gangs.csv
    - Thatcher.csv
- Remove rows that do not have an "Elementary" parse, and then merge all 3 datasets in a single combined dataset.
- To show that you have loaded the data correctly, print the number of rows in your combined dataset. Show this number in your notebook. You should see **35 rows,** after removing rows with no Elementary text.
- For the first row in your dataset, print the Elementary and Advanced texts. Show the output in your notebook. Does the Advanced text seem to use more complex language than the Elementary text? (You do not have to answer this question in writing)

PROBLEM 4 – Analyzing the data (40 points)

In this problem, you will compare Elementary and Advanced texts that you read in the previous problem, to consider how texts that express the same ideas can vary syntactically by reading level.

- Write a function that takes a list of texts as input, applies the Stanza constituency parser to each multi-sentence text, and then uses the output to create a **data summary** of these texts. Your output should include these attributes:
    - The average number of sentences in each text
    - The average number of **prepositional phrases** in each text [You can compute this by scanning the tree recursively, or by searching the output of stanza's "pretty_print()" function for "PP", the Penn Treebank symbol for prepositional phrases]
    - **One other attribute of your choice** that is based on output of the stanza pipeline
- Apply your function twice, to the data you created in Problem 3:
    - The set of 35 Elementary texts
    - The set of 35 Advanced texts (after dropping rows with no analogous Elementary text)
- Show your results in your notebook. Check that you are showing all 3 attributes on both the Elementary and Advanced datasets.
- PROGRAMMING TIP: You may want to write helper functions to break this problem into steps. For example, you might write a function that takes a list of unprocessed texts as input and returns them as parsed Document objects. Note you are also asked to write one larger function to run the full analysis, so that the grader can see that you applied the **same steps** to both Elementary and Advanced texts.

PROBLEM 5 – Evaluating Complexity Level (10 points - Answer in Blackboard)

- Describe the attribute that you added in Problem 4. Answer briefly, in 1-3 sentences.
- Does your new attribute appear to be a useful indicator of reading level? Explain your answer briefly. Your response should refer to the results in your notebook. (It's okay if your attribute does not appear useful, or your results are inconclusive. You do not need to redo your work if you do not see a "positive" result.)

PROBLEM 6 – Additional Methods - (10 pts – Answer in Blackboard)

Imagine you have been asked to build a tool that can **classify texts by reading complexity level,** so that students of English can be presented with material that matches their current level.

- What tool or method have we learned this semester that you would apply? Name at least one topic or method and briefly describe how you would apply it to this specific task. (You may choose any method from this semester, and your answer does not need to apply syntax or parsing.)

CODE ORGANIZATION AND READABILITY (5 pts)

The receive full credit, please ensure that:

- You have included **both** notebook and PDF files, as described in "What to Submit". You will lose points if you do not submit both of these above files.
- You have named your files "HW5_CWID_LastName.ipynb" and "HW5_CWID_LastName.pdf"
- Your notebook includes cell output, and does NOT contain error messages
- It is easy to match a problem with its code solution in your notebook via markdown or comments (e.g., "Problem 2")
- You re-ran your notebook before submitting, and cells are numbered sequentially starting at [1]
- Your answers to written questions are clear and concise. You will not receive more points for long answers that discuss topics beyond what is requested.

REFERENCES

Sowmya Vajjala and Ivana Lučić. OneStopEnglish corpus: A new corpus for automatic readability assessment and text simplification. 2018. Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 297–304. Association for Computational Linguistics. [pdf]

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton and Christopher D. Manning. 2020. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In Association for Computational Linguistics (ACL) System Demonstrations. 2020. [pdf]