Juliana McCausland
Ling 473
Project 1

Results:

Sentence	4670
Noun Phrase	7920
Verb Phrase	13221
Ditransitive Verb Phrase	33
Intransitive Verb Phrase	123

Project Description:

I took a bit of a scattered approach to this project, primarily because I had trouble working with the files in the corpus. To find the S, NP, and VP counts, the program calls a single function while iterating through each file (after cleaning the contents of each file to ease the process) and counts the occurrences of the patterns '(S', '((S', '(NP', '(VP' with the built-in count() function. The totals are added to a dictionary that contains specific key-value pairs for each constituent. The dictionary is used as an output table.

Counting the ditransitive and intransitive verb phrases was more of a challenge, as it required a framework that understood the tree structures of the files. After trying and failing with various brute force methods of keeping an index of parentheses counts, I turned to NLTK. I used the ParentedTree function to more easily navigate and count individual nodes. To make my trees functional, I needed to use the SExprTokenizer (s-expression tokenizer) – a tool equipped to handle parenthesized expressions. The program cycles through each file, creates substrings with the SExprTokenizer and turns those into parented trees. The program then calls two separate functions (one to count ditransitive verb phrases, and the other to count intransitive verb phrases) while iterating through each parented tree. The functions use tgrep (which immensely simplifies tree navigation) to find specific structures in each parented tree. It cycles through each matching tgrep and add all values that contain a specified number of nodes to a final list. The length of each list (which is equivalent to the count of each constituent) is added to the dictionary.

The totals are printed from the dictionary.

Notes:

I kept two output files in my tarball: ‘output’ and ‘output.out’. The version ‘output’ did not allow me to view the results once I moved it to my computer (though it contained all the correct information when displayed within terminal), so I worried that it would do the same for others. ‘output.out’ opened without problems and displayed the final output as expected. I chose to keep ‘output’ in my tarball because the check_project1.sh returned an error whenever the ‘output’ file was not in the tarball. I don’t expect this to be an issue, but I just wanted to clarify why there are two output files.