**ECE 3822: Engineering Computation II**

# Homework No. 11: Python Scripting

**Goal:** The goal of this homework is to repeat our filename-processing example using Python.

Write a SINGLE Python script that:

1. Takes “-help” as an argument and displays an appropriate help message.
2. Takes arguments “-ngram 3” and “-pattern spike”. In this case the 3-gram histogram for all files containing the word “spike” would be produced. Similarly, “-ngram 1 –pattern sharp” would produce a 1-gram histogram of all files containing the word “sharp”. You should be able to specify these options anywhere on the command line and in any order.
3. Takes multiple lists of files as arguments (e.g., “foo.py myfile1.list myfile2.list mylists\*.list”) and loops over all of the filenames in these lists.
4. Counts the number of files that have the target word specified by “-pattern” in them and produces the following message: “A total of %d files contained the word %s”. “%d” and “%s” are replaced by the actual count and the actual pattern requested.

Generate a histogram of the ngrams using the following format:

|  |  |  |  |
| --- | --- | --- | --- |
| Trigram | Frequency | | |
| No. | Percentage | Cumulative |
| see jane run | 3 | 37.5000% | 37.5000% |
| jane run see | 2 | 25.0000% | 62.5000% |
| run see john | 1 | 12.5000% | 75.0000% |
| see john run | 1 | 12.5000% | 87.5000% |
| run see jane | 1 | 12.5000% | 100.0000% |

You must count ngrams that cross line boundaries and punctuation (e.g., end of sentence).

1. Run your script on the test data provided in class for the cases:

-ngram 1 –pattern “spike”

-ngram 2 –pattern “spike”

-ngram 3 –pattern “seizure”

Time your script and comment on the processing time compared to writing this in C/C++ code.

Submit your solution in this directory: /data/courses/ece\_3822/current/homework/hw\_11/<lastname>. The driver program should have a name “myprog.py”. Put your findings about CPU time in the comments at the head of the file containing your main program.