Program 2: Lists

New Attempt

Due Mar 19 by 11:59pm **Points** 20 **Submitting** a file upload
Available after Feb 17 at 12am

Overview:

In this assignment, students shall build a program that extracts tokens from an ASCII encoded input file. Using the words found within, the Python application shall extract every sequence of three consecutive words appearing in the file, and it shall then use these trigram objects to construct a random poem.

During this program, students will gain experience:

- Locating files on the file system with python (module: os, sys)
- Extracting tokens from data in an input file
- Working with a file as a list of strings
- Using tuples to combine related fields into a single structure
- Exploring some of the computational complexity limitations that emerge when using lists (i.e. searching and finding).
- Formatting output for human readability.

Requirements:

Upon launching the script, the program shall open an ASCII encoded input file. The program shall establish which file to open based upon the presence of any command-line-arguments. If the user launched the program without any (e.g. 'python3 prog2.py'), the program shall prompt the user to enter the path to a file to use. In either case, the script shall then verify the file exists. If the program is unable to open or locate a valid input file, it shall prompt the user to reenter the file name to use. It shall continue to ask the user at least three (3) times and no more than (5) times before abandoning all hope and exiting the program.

Using the user-identified input file, the program shall then begin constructing tuple objects of every three words. For example, given the following text: "We are the crystal gems"

We extract the trigrams:

```
We are the
are the crystal
the crystal gems
```

Students should accomplish this task through a function that accepts the file as an input parameter (either its name or already opened object) and returns the trigrams it finds there as a List of Tuples. During the extraction process, the program shall filter out and clean up the input tokens. The program shall ignore terminating period characters. Additionally, it shall store every token it identifies as either its upper or lower case version. For

example, this will make appearances of 'And' and 'and' identical. Upon extracting the trigrams, the program shall report the number of trigrams it extracted from the input file. At this point, the program should not need to open the input file again.

Using the list of tuples extracted from the input file, the program shall then randomly assemble a poem. All poems constructed by this algorithm use a 5/7/5 format where the first line contains five (5) words, the second line contains seven (7) words, and the final line contains five (5) words. The script shall build a poem by randomly selecting a starting trigram. Using this tuple, the program shall begin the poem's first line. To reach the five word total, the poem requires two additional words, so it shall use the last word of the initial trigram to search and find a tuple contained in the list that begins with that word. If the program selected "I'll do the" as the first trigram, it shall then search through its list of trigrams and find every one that begins with the word 'the'. From this smaller list, it shall randomly select a second trigram and splice them together. Thus, if the second trigram is: "the bee's knees", our first line becomes:

```
I'll do the bee's knees
```

If you continue this process with the remaining seven word and five word lines, and we have a machine generated poem. The script shall display this poem to the user in cleanly formatted output. The program shall generate and display at least five (5) dynamic poems. Moreover, the program shall also create a file called "output.txt" containing these exact results.

Deliverable items:

Students shall include the python script written for this assignment as well as one of the example output.txt files created after running the script. Students need not deliver the input text file used to create the poems, but they may do so if inclined.

Grading:

Reads command-line-arguments (CLA) (2)

Prompts for input file if no CLA (2)

Verifies file exists and prompts if it cannot open (1)

Cleanly formatted screen output (3)

Cleanly formatted output file (2)

Function extracts trigrams into list of tuples (4)

Tokens cleaned (e.g., lower, strip, no trailing period) (1)

Poem randomly built using trigrams with correct number of lines and words (5)

The graders may award additional points for solutions that pay extra attention to output formatting, word conditioning, and additional artistic intelligence introduced in the algorithm. Additionally, students may expand the trigram model to 4-gram objects and build a poem using these. This modification is worth substantial points

(up to 5), and it should display examples of poems built with tri-grams vs. those with 4-grams. That is, you must provide 4-gram functionality as an additional feature, and not one that replaces the 3-gram requirement.

Additional Notes:

Trigrams (https://en.wikipedia.org/wiki/Trigram) are used by your auto-correct algorithm, so you experience them daily. As you start typing into google, the list of recommended searches uses a similar approach. The first few words determine the likely remaining words.

Command-line-arguments are accessible through the sys module. The variable sys.argv is a list of arguments present when the user started the script. The item at index zero (0) will be the name of the script itself, and the remaining, space-separated arguments appear from index one (1) and beyond.

To determine if a file exists in python, use the os module. It includes 'path' objects and functions which you can use to make certain the users entered a valid location and not simply garbage.

Saving a new output file in python is easy using the 'with' and 'open' keywords/functions. This may help: **https://docs.python.org/3/library/functions.html#open (https://docs.python.org/3/library/functions.html#open)**

Ethics Related:

If you have free time to explore some CS ethical issues, here are a few related links to this assignment. These do not appear on our material in this class, so reading/viewing them is fully optional, but I think they do illustrate a modern problem worthy of our consideration.

Musicians use AI to generate every possible melody: **https://www.techdirt.com/articles/20200220/09561543951/attempt-to-put-every-musical-melody-into-public-domain-demonstrates-craziness-modern-copyright.shtml (https://www.techdirt.com/articles/20200220/09561543951/attempt-to-put-every-musical-melody-into-public-domain-demonstrates-craziness-modern-copyright.shtml)**

Katy Perry: **https://www.techdirt.com/articles/20200319/00094444129/surprise-judge-throws-out-jurys-awful-copyright-infringement-decision-over-katy-perry-song.shtml (https://www.techdirt.com/articles/20200319/00094444129/surprise-judge-throws-out-jurys-awful-copyright-infringement-decision-over-katy-perry-song.shtml)**

The original issue (which was thrown out): **https://www.youtube.com/watch?v=0ytoUuO-qvg (https://www.youtube.com/watch?v=0ytoUuO-qvg)**