Python Assignment: Text Generator

In this assignment you will read in a text file and automatically generate new text which stylistically mimics the original.

In order to capture the style of the text, we will look at the way words follow one another. For example, if I look at the sentences:

> “I am what I am and that’s all that I am. I am what I am.”

We can see that the words “I am” are followed by “what” twice, “and” once, and “I” once. The words “what I” are followed by “am” twice. To mimic this style, we would want to use these patterns in our new text.

1. Your program should begin by prompting the user for the name of the text file.
2. You should then read in the text file and create a dictionary, where the keys are every two-word combination, and the values are a list of the words that follow.

For the above sentence, my dictionary would look like:
I am : [what, and, I]
am what: [I]
what I: [am]
am and: [that’s]
Etc…

3. Generate and print up to 500 new words of text. Start with the first 2 words of the original text. Then use your dictionary to generate the next word. If there is no appropriate entry in the dictionary at any time, the text can just stop. To generate the next word from the dictionary, you may do one of the following:
    a. (B+ level) Use a random word from the list.
    b. (A- level) Use the most frequent word from the list. In the above example, I would always choose “what” after “I am” because it occurred the most. If there are ties in frequency, use the alphabetically first word. (See the note on sorting below.)
    c. (A+ level) Use the most frequent word from the list, but then remove it so the next time that pair of words is encountered a new word will be generated. This is to remove cycles in the text, (“I am what I am what I am what…”). Only remove entries if there is another word to replace it. For example, in the dictionary above I can remove from the “I am” list, but not from the “am what” list, because it only has one entry.

A Note About Sorting

Being able to sort a list may be helpful in this assignment. If the list contains items that are natively sortable (like strings or numbers), you can just call .sort() on your list variable. However, you may want to sort things that are not lists, or things that are not obviously sortable (like tuples.)

To do this, we generally use the sorted method with a key. The key tells what to sort based on. Suppose I have a list of tuples like this:

```
tuples = [ (“I”, 1), (“and”,1), (“what”, 2) ].
```

I want to sort first by number in descending order, then by the word. The line to do this would be

```
tuples = sorted(tuples,key=lambda t: (-1*t[1],t[0].lower()))
```

This says the key to sort on is -t[1], or the negative value of the second thing in the tuple. That’s so I get the 2’s before the 1’s. Then the second key is used as a tie-breaker: t[0].lower(), or the lower-case version of the word. (In python, every upper-case word is alphabetically before every lower-case word, so it’s best to consider just the lower case version of the word.)

After this line, I have:

```
tuples = [(“what”, 2), (“and”,1), (“I”, 1)].
```

Text Files and Examples

Here are some text files to run for fun, as well as the beginning of my generated output.

Bible.txt

Genesis 1:1 In the day of the LORD thy God hath given me a father to the house of the LORD, and the LORD hath spoken it. Ezekiel 11:8 Ye have heard of the children of Israel and the children are come to pass, when the LORD of hosts, the God of Israel, and say unto you, and ye shall be a statute for ever and ever. Amen.

TreasureIsland.txt

From the side of the boats. Instantly the figure reappeared, and, making a wide circuit, began to believe that I had heard of cannibals. I was a civil, pious boy, and could rattle off my catechism that fast as you ask me there are some of Flint's hands aboard; worse luck for the fear of Benjamin Gunn.

Grail.txt (Script of Monty Python and the Holy Grail)

_______________________________________ | | | Scene 1 | |____________| [King Arthur music] [clop clop clop] ARTHUR: Halt! [horn] Hallo! [pause] Hallo! FRENCH GUARD: (Fetchez la vache!) [mooo] ARTHUR: If you do not agree to my commands, then I shall-- [twong] [mooooooo] Jesus Christ! [thud] Ah! Ohh! ARTHUR: Right! Charge! KNIGHTS: Charge! [mayhem] FRENCH GUARD: And now, remain gone, illegitimate-faced bugger-folk! And, if you do doubt your courage or your strength, come no further, for death awaits you all with nasty, big, pointy teeth. ARTHUR: What is your name? ARTHUR: It is a witch? VILLAGER #2: A bit.