CSE 231 Fall 2016

COMPUTER PROJECT 9

(Nov. 15: Added a small input file for testing. See item 5 under Notes.)

Assignment Overview

This program focuses on the use of dictionaries, sets, files and text manipulation..

This assignment is worth 55 points (5.5% of course grade), and must be submitted before 11:59 PM on Monday, November 21.

Background

We take word completion for granted. Our phones, text editors, and word processing programs all provide suggestions for how to complete words as we type based on the letters typed so far. These hints help speed up user input and eliminate common typographical mistakes (but can also be frustrating when the tool insists on completing a word that you don't want completed).

Overview

You will implement two functions that such tools might use to provide command completion. The first function, `fill_completions`, will construct a dictionary designed to permit easy calculation of possible word completions. A problem for any such function is what *vocabulary*, or set of words, to allow completion on. Because the vocabulary you want may depend on the domain a tool is used in, you will provide `fill_completions` with a representative sample of documents from which it will build the completions dictionary (we provide the file `ap_docs.txt`). The second function, `find_completions`, will return the set of possible completions for a prefix of any word in the vocabulary (or the empty set if there are none). In addition to these two functions, you need a function to open a file, ane will implement a simple driver program to use for testing your functions.

Program specifications

- `open_file()` prompts the user to enter a file name. The program will try to open the data file. Appropriate error message should be shown if the data file cannot be opened. This function will loop until it receives proper input and successfully opens the file. Use of try-except is required. It returns a file pointer. Use this function from previous projects.
- `fill_completions(fd)` returns a dictionary whose keys are tuples and the values are **sets**. This function takes as an opened file pointer. It returns the completion dictionary as described below.
    - The keys are tuples of the form `(n, l)` for a non-negative integer `n` and a lower-case letter `l`.
    - The value associated with key `(n, l)` is the **set** of words in the file that contain the letter `l` at position `n`. For simplicity, all words are converted to lower case. For example, if the file contains the word `"Python"` then the sets returned by `c_dict[0,"p"], c_dict[1,"y"], c_dict[2,"t"], c_dict[3,"h"], c_dict[4,"o"],` and `c_dict[5,"n"]` all contain the word `"python"` (as well as other words).

    - Words are stripped of punctuation.
    - "Words" containing non-alphabetic characters are ignored, as are words of length 1 (since there is no reason to complete the latter).
- `find_completions(prefix, c_dict)` returns a **set** of strings. This function takes a prefix of a word (possibly empty) and a completions dictionary of the form described above. It returns the **set** of words in the completions dictionary, if any, that complete the prefix. It the prefix cannot be completed to any vocabulary words, the function returns the empty **set**.
- Main part of your program:
    - Calls `open_file()` to get a file pointer (in this case we want the file `"ap_docs.txt"`, but we are not testing for that filename). This file contains a collection of old newswire articles.
    - Calls `fill_completions` to fill out a completions dictionary using this file.
    - Repeatedly prompts the user for a prefix to complete or for an `'#'` to quit.
    - Prints the set of words that can complete each prefix or states that the prefix has no completions.

Assignment Deliverable:

The deliverable for this assignment is the following file:

`proj09.py` -- your source code solution

Be sure to use the specified file name and to submit it for grading via the **handin system** before the project deadline

Assignment Notes:

0. Items 1-9 of the Coding Standard will be enforced for this project.
1. You will find `enumerate` very useful, e.g.
   ```
   for i,ch in enumerate(someString):
   ```
2. Be smart!!! Implement the functions and test them thoroughly on inputs for which you know what the answer should be. That means you will want to use a much smaller input file initially.
3. The design of the completions dictionary makes retrieval of completions a simple matter using intersection of sets. Consider, for example, possible completions of `"pyt"`. You should be able to convince yourself that the set of possible completions is the **intersection** of the sets `c_dict[0,"p"]`, `c_dict[1,"y"]`, and `c_dict[2,"t"]`.
4. Python provides a binary operation for finding intersection of sets, denoted `&`. However, you need to form the intersection of an arbitrary number of sets (depending on the length of the prefix). How will you do this? In principle, this is no different than finding the sum of the numbers in a list `L` of arbitrary size using the binary + operator. To sum the list you initialize a working variable, say `result`, making it 0. Then, you add each subsequent element of `L` to `result` possibly using `"+="`, You can do essentially the same thing for the intersection problem. For the example above, you can initialize `result` to be `c_dict[0,"p"]`. Then, using **set intersection** on `result` and `c_dict[1,"y"]` to

get the next value for `result`. Finally, intersect `result` and `c_dict[2,"t"]` to get the intersection of the three sets.

5. I created a small test input file named `test.txt` and a second file that has the dictionary created from that file named `test_dictionary.txt`. You might find them helpful.

Sample Output

```
Input a file name: ap_docs.txt

Enter the prefix to complete (or '#' to quit): promot
Completions of promot: promoted promoters promoting promotional
promote promotions promotion


Enter the prefix to complete: promotion
Completions of promotion: promotional promotions promotion


Enter the prefix to complete: col
Completions of col: colleges columbia coleman colter colorado
colorblind colo colleague collett colleagues collateral column
columnist collapsed colony colombo colman colossus college
collapse cold colombia collecting colleages coloradoan colonial
col colombian collision collected


Enter the prefix to complete: weig
Completions of weig: weigh weighed weighs weight


Enter the prefix to complete: #
~
```

Questions for you to consider (not hand in):

A problem with your `find_completions` function is that, for many short word prefixes, it returns too many possible completions to be useful. An editing tool that invokes your function will need to select some subset of the possible completions to display to a user. To permit this, it would be useful if your `find_completions` function returned a ranked list of completions, in order by decreasing frequency of use. If you assume that the input file to your `fill_completions` function is representative for the domain of the tool you are building, this function could collect the information needed to determine how to rank the possible completions for each prefix. How would you redesign the completions dictionary to record this information? How would you modify your two functions? Finally, is it better for `find_completions` to return a ranked list of all possible completions for a prefix or just the five or six top-ranked words? Why?

==========

Educational Research

When you have completed the project insert the 5-line comment specified below.

For each of the following statements, please respond with how much they apply to your experience completing the programming project, on the following scale:

1 = Strongly disagree / Not true of me at all
2
3
4 = Neither agree nor disagree / Somewhat true of me
5
6
7 = Strongly agree / Extremely true of me

*********Please note that your responses to these questions will not affect your project grade, so please answer as honestly as possible*.*****

Q1: Upon completing the project, I felt proud/accomplished

Q2: While working on the project, I often felt frustrated/annoyed

Q3: While working on the project, I felt inadequate/stupid

Q4: Considering the difficulty of this course, the teacher, and my skills, I think I will do well in this course.

Please insert your answers into the bottom of your project program as a comment, formatted exactly as follows (so we can write a program to extract them).

```
# Questions
# Q1: 5
# Q2: 3
# Q3: 4
# Q4: 6
```