HW05: The Complete Works of William Shakespeare

CSE2050 Fall 2017
Instructors: Jeffrey Meunier & Wei Wei
TAs: Jenny Blessing, Param Bidja, Yamuna Rajan & Zigeng Wang

1. Introduction

In this assignment, we are going to read the complete works of William Shakespeare and write an essay about how Shakespeare chooses his words.

Just kidding :-) But this is not far from the truth. In fact, we are going to write Python code to read the complete works of William Shakespeare and then report the word frequencies of the words used in his works.

2. Objectives

The purpose of this assignment is to give you experience in:

- Understanding when to use dictionary and how dictionary is implemented.
- Implementing and using the basic operations of a mapping (dictionary).
- Learn how to handle a file and practice some string operations.
- Learn what to do when we need to sort a dictionary according to the values.

***Note**: Before you start, if you are not familiar with content of mappings and hash tables, it is recommended you read the class notes first and do the code for thought #13, #14, and #15.*

3. Background

3.1. The Complete Works of William Shakespeare

The text file that has the complete works of William Shakespeare is provided with the skeleton code. This text file has a special format. That is, every word or symbol is followed by a space. This makes the job of splitting words easier. This text file has all the works of Shakespeare, and it is very long. The size of the text file is about 4.5 Mbytes. The total number of lines is 129,107, the total number of words and symbols is 980,637, and the total number of characters is 4,538,523.

Here is the beginning of the text file:

A MIDSUMMER-NIGHT'S DREAM

Now , fair Hippolyta , our nuptial hour

Draws on apace : four happy days bring in

Another moon ; but O ! methinks how slow

This old moon wanes ; she lingers my desires ,

Like to a step dame , or a dowager

Long withering out a young man's revenue .

Four days will quickly steep themselves in night ;

Four nights will quickly dream away the time ;

And then the moon , like to a silver bow

New-bent in heaven , shall behold the night

Of our solemnities .

Go , Philostrate ,

Stir up the Athenian youth to merriments ;

Awake the pert and nimble spirit of mirth ;

Turn melancholy forth to funerals ;

The pale companion is not for our pomp .

Hippolyta , I woo'd thee with my sword ,

And won thy love doing thee injuries ;

But I will wed thee in another key ,

With pomp , with triumph , and with revelling .

We will read this text file and count the number of times each of the words and symbols appear in the text file.

4. Assignment

In the skeleton zip file, you will find a skeleton for your .py file named *shakespeare.py*. All you need to do is to complete the skeleton code based on the instructions and submit it to the Mimir system.

4.1. Solving the problem using dictionary

We first solve the problem using dictionary. The symbol code is included in the skeleton code.

```
import operator

import time


f = open('shakespeare.txt', 'r')

data = f.read()

f.close()


data = data.split()

start = time.time()


d = {}

for i in range(len(data)):

    key = data[i].lower()

    if key in d:

        d[key] += 1

    else:

        d[key] = 0
```

```
sorted_d = sorted(d.items(), key=operator.itemgetter(1), reverse = True)


k = 0

count = 0

while count < 20:

   if sorted_d[k][0] not in {"'", ":", "!", ".", ",", ";", "?"}:

      print(count + 1, sorted_d[k])

      count += 1

   k += 1


end = time.time()

print(end - start)
```

In the above code, we first read the text file into a string and split the string into symbols and words. And then we loop through the list of symbols and words and use a dictionary to count the number of times these symbols and words appeared in the list. After that, we sorted function to sort the dictionary according to the values and save the result into a list. Last, we print out the top 20 frequently used words and their corresponding frequencies.

4.1. Solving the problem using HashMapping

Your job now is to use the code from code for thought #14 and code for thought #15 to implement class HashMapping and use this class to replace the dictionary above and repeat the above operations and obtain the same results. Note you need to implement class Entry, Mapping, ListMapping and HashMapping.

Below is a screenshot of my results. Using dictionary, we get the job done in 2 seconds. Using HashMapping, it is done around 9 seconds.

```
1 ('the', 26804)
2 ('and', 24037)
3 ('i', 20041)
4 ('to', 18532)
5 ('of', 16006)
6 ('you', 13833)
7 ('a', 13678)
8 ('my', 12256)
9 ('that', 10718)
10 ('in', 10524)
11 ('is', 9138)
12 ('not', 8450)
13 ('me', 7757)
14 ('it', 7736)
15 ('for', 7538)
16 ('with', 7141)
17 ('be', 6840)
18 ('your', 6744)
19 ('this', 6584)
20 ('his', 6528)
1.9993631839752197
------------------------------------
1 ('the', 26804)
2 ('and', 24037)
3 ('i', 20041)
4 ('to', 18532)
5 ('of', 16006)
6 ('you', 13833)
7 ('a', 13678)
8 ('my', 12256)
9 ('that', 10718)
10 ('in', 10524)
11 ('is', 9138)
12 ('not', 8450)
13 ('me', 7757)
14 ('it', 7736)
15 ('for', 7538)
16 ('with', 7141)
17 ('be', 6840)
18 ('your', 6744)
19 ('this', 6584)
20 ('his', 6528)
```

5. Submit your work to Mimir

Submit your code to Mimir after you complete your code. Mimir will automatically check the output of your code and grade your submission. You can submit your code to Mimir up to **30 times** to refresh your existing score before the submission deadline. Before you submit your code, comment out all the code outside of the class Entry, Mapping, ListMapping and HashMapping.

6. Due date

This lab assignment is worth **4 points** in the final grade. It will be due by **11:59pm on November 13th, 2017**. A penalty of **10% per day** will be deducted from your grade, starting at 12:00am.

7. Getting help

Start your project early, because you will probably not be able to get timely help in the last few hours before the assignment is due.

- Go to the office hours of instructors and TAs.
    - Prof. Wei Wei: Mon. 2:30 - 3:15pm, Wed. 2:30 - 3:15pm, Thurs. 2:30 - 3:15pm, Fri. 2:30 - 3:15pm @ITE258
    - Jenny Blessing: Fri. 12pm - 2pm @ITE140
    - Param Bidja: Tues. 2pm - 3pm @ITE140
    - Yamuna Rajan: Tues. 11am - 12pm, Wed. 9:30am – 10:30am @ITE140
    - Zigeng Wang: Mon. 3pm - 4pm @ITE140
- Post your questions on Piazza. TAs and many of your classmates may answer your questions.
- Search the answer of your unknown questions on the Internet. Many questions asked by you might have been asked and answered many times online already.