CS 101
Fall 2016
Program Assignment # 7 - Word Frequency
Algorithm Due : Sunday, Nov 20, 2016
Program Due : Wednesday, Nov 30, 2016

Word Frequency

Google's ngram viewer allows you to see the relative frequency words are used in literature throughout the years. You can test their n-gram viewer at https://books.google.com/ngrams

You'll have 2 files to work with; all_words.csv and total_counts.csv. Allwords will contain all the words that the user can compare with their counts in literature for each year. total _counts has the counts of all words by year, so that we can find the relative frequency.

all_words.csv

```
Art,1505,5,1
Art,1524,3,1
Art,1564,12,1
Art,1568,1,1
Art,1574,11,1
Art,1575,1,1
Art,1579,1,1
Art,1581,13,3
```

Each line in the csv contains the word, followed by the year and how many times that word was used during the year. The last number is how many books it came out of. We won't use the last value.

total_counts.csv

```
1505,32059,231,1
1507,49586,477,1
1515,289011,2197,1
1520,51783,223,1
```

Each line for total counts have the year followed by how many total words were used in literature for that year. To find the relative frequency %, simply take the amount of times the word was used in a year and divide it by the total words in the year. For instance, Art was used 5 times in year 1505 with 32059 total words in 1505. So the relative frequency % is 5/32059*100 = 0.01559 %

Your program will get 2 words from the user, a start date, and end date and then display in a table the relative frequencies by year.

Requirements

- Your programs should make good use of functional decomposition. Make sure you break your program down into functions.
- The program will be menu driven, and will ask the user if they want to see an ngram view of 2 words or quit.
- If the word the user gives is not in available, the program should prompt the user again. You will need to ask for 2 words for comparison.
- The user will provide a start date, between 1505 and 2008. If the user hits enter, the default value will be 1900. If the user doesn't provide a valid integer they will be shown an error and asked to reprompt. If the year isn't between 1505 and 2008 inclusive, then they will be warned and prompted.
- The user will also be asked to provide an end date. The valid range for the end date should be between the start date and 2008 inclusive. You will be required to validate the value in the same way that start date was. The default value for the end date will be 2008.

Example

```
>>> ================================ RESTART
================================
>>> 
         Ngram Viewer

        1. Get Ngram Table
        Q. Quit

==> e
You must choose a valid item from 1,Q
         Ngram Viewer

        1. Get Ngram Table
        Q. Quit

==> 1
```

```
Enter a word to get the frequencies of ==> twitter
twitter was not found, choose another word

Enter a word to get the frequencies of ==> internet

Enter a word to get the frequencies of ==> book

Enter the start date. default[1900] bad answer
You did not enter an integer.  Try again

Enter the start date. default[1900] 300
You must enter an integer between 1500 and 2008

Enter the start date. default[1900] 1980

Enter the end date date. default[2008] 1979
You must enter an integer between 1980 and 2008

Enter the end date date. default[2008] bad answer
You did not enter an integer.  Try again

Enter the end date date. default[2008]

           Ngram Results
  Year       internet           book
====================================
  1980       0.000032       0.019580
  1981       0.000011       0.019132
  1982       0.000010       0.019570
  1983       0.000021       0.019550
  1984       0.000027       0.020017
  1985       0.000023       0.019676
  1986       0.000037       0.019464
  1987       0.000050       0.019643
  1988       0.000066       0.020115
  1989       0.000059       0.020251
  1990       0.000132       0.020551
  1991       0.000145       0.020956
  1992       0.000286       0.020906
  1993       0.000659       0.021530
  1994       0.001648       0.021814
  1995       0.003184       0.022348
  1996       0.004109       0.022943
  1997        0.005963       0.022944
  1998        0.006428       0.023294
  1999        0.007001       0.023210
  2000        0.007890       0.023426
  2001        0.008955       0.023373
  2002        0.009359       0.024020
  2003        0.008262       0.024156
  2004        0.007073       0.024001
  2005        0.006594       0.024144
  2006        0.005965       0.024327
  2007        0.005736       0.024029
  2008        0.004539       0.023347


         Ngram Viewer

        1. Get Ngram Table
        Q. Quit

==> q
>>>
```

Tips

- The csv module helps in reading/writing csv files. When you use it to read through a csv file, it will read each line as a list of strings with the comma removed. Instead of iterating over the opened file handle, you iterate over the csv variable.

```
import csv
file=open("all_words.csv")
csv_file = csv.reader(file)

for line in csv_file:
        print(line)               # Line is a list of strings for each line.
file.close()
```

Extra Credit (5 Points)

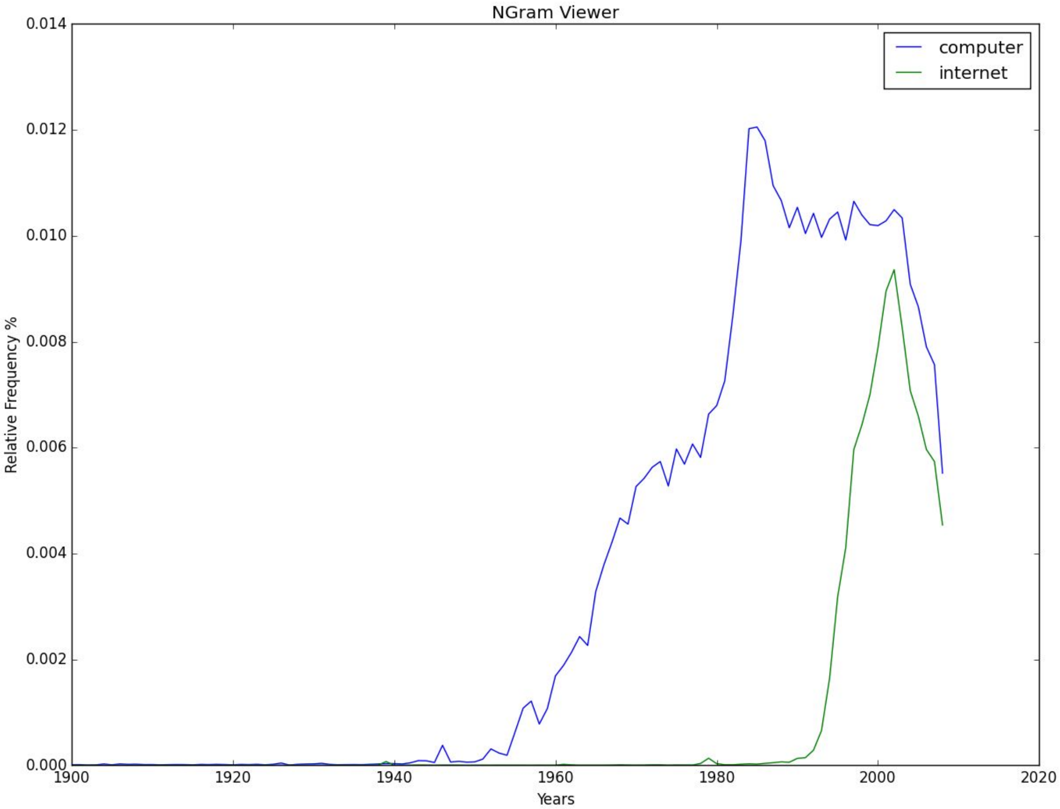
The python module pylab easily lets you make graphs. It is not an installed library. The UMKC labs in Flarsheim do have it installed though. If you want to install it yourself you'll need to follow some internet tutorials on how to install it for python 3.4 for whatever operating system you use.

You'll want to change the program to allow the user to get just a table of output, a table and a graph or just a graph of the results.

```
          Ngram Viewer

         1. Show Ngram Table
         2. Show NGram Graph and Table
         3. Show Only Ngram Graph
         Q. Quit

==>
```



```
>>> import pylab
>>> x_values = [1, 2, 3, 4, 5]
>>> y1 = [10, 14, 15, 16, 17]
>>> y2 = [20, 19, 18, 17, 16]
>>> pylab.plot(x_values, y1)
[<matplotlib.lines.Line2D object at 0x05987530>]
```

```
>>> [<matplotlib.lines.Line2D object at 0x059CAF30>]
SyntaxError: invalid syntax
>>> pylab.plot(x_values, y2)
[<matplotlib.lines.Line2D object at 0x05987A90>]
>>> pylab.legend(["y1 values", "y2 values"])
<matplotlib.legend.Legend object at 0x05987ED0>
>>> pylab.title("Sample Graph")
<matplotlib.text.Text object at 0x059763D0>
>>> pylab.xlabel("X Values")
<matplotlib.text.Text object at 0x059A23F0>
>>> pylab.ylabel("Y Value Scale")
<matplotlib.text.Text object at 0x0075DBD0>
>>> pylab.show()
>>>
```

References

- https://docs.python.org/3.6/library/csv.html