Homework No. 1
CISC 6210 Natural Language Processing **Due Date: 9/9/2020**
Fall 2020 **Points: 10**

Goal

- Practice raw text data processing with Python.
- Get familiar with regular expression
- Practice text tokenization techniques
- Prepare data for further text analysis

Data Set

2235 love poems are collected from www.poetryfoundation.org. You could access them from the folder - http://storm.cis.fordham.edu/~yli/data/LoveOutput. Each poem is in its own .txt file.

Example of a poem file:

1979 By Roddy Lumsden

Living; Life Choices; Time & Brevity; Love; Desire; Couplet; Free Verse

They arrived at the desk of the Hotel Duncan<br x..>and Smithed in, twitchy as flea-drummed squirrels.<br><br>Her coat was squared and cream, his patent shoes<br>were little boats you…
……..
<br>our shadows' footsteps clatter, they match our dread. <br><br><br>

Original link: https://www.poetryfoundation.org/poetrymagazine/poems/54541/1979

- First paragraph is Title and Author
- Second paragraph is topic tags marked by the website.
- Third paragraph is the body of the poem. <br> marks line break and <br><br> marks paragraph break.
- The last part is the original link of the poem. You could visit it to check the original poem.

Tasks

1. **Remotely** read all poem files from the storm server into a table, which has five columns - Author, Title, Tags, Body, and Link. Each poem is a row of the table.
   1) Author: Poem's author name.
   2) Title: Poem's Title
   3) Tags: All tags given to the poem, separated by ';'
   4) Body: The body of the poem, with line breaks replaced by [L] and paragraph breaks replaced by [P]. All other html tags are removed. The original text is untouched.
   5) Link: Only the web address is saved.

   If some part of the poem is missing (Link is optional), do not save it into the table.

   Once you processed all poems, show information of the table on screen, such as total number of poems stored, total number of authors, sort authors by the amount of their poems collected in the table, and show top 20 authors.

- Save the table into a Microsoft Excel Worksheet named **CleanOutputLoveOutput.xlsx**.

2. Read data from **CleanOutputLoveOutput.xlsx**, collect information about words, sentences, paragraphs, and punctuations of poems in this data set.
   1) Create a new table, which has columns - PoemID, Author, LengthOne, LengthTwo, NumLine, NumPara, NumSent, NumComma
      a. PoemID: the id number of this poem in the table created in step no.1.
      b. Author: Poem's author name.
      c. LengthOne: the total number of tokens in this poem, punctuation marks are **not** included.
      d. LenthTwo: the total number of tokens in this poem, punctuation marks are included.
      e. NumLine: the total number of line breaks plus paragraph breaks.
      f. NumPara: the total number of paragraph breaks.
      g. NumSent: the total number of complete sentences (use nltk)
      h. NumComma: the total number of commas.

      Show the statistics information about this data set, such as average, max. and min.
   2) Create three new tables, each has five columns – PoemID, Author, Body, Length, and UniCount.
      a. PoemID: the id number of this poem in the table created in step no.1
      b. Author: Poem's author name.
      c. Body, Length, and UniCount: tokenized words in the poem, its length, and its vocabulary size.

      The first table is for tokenized words from original poems without any processing (use nltk tokenizer). The second table is for tokenized words after stop words removal. (use nltk English stopwords list). The third table is for tokenized words after stop words removal and stemming. (use one stemmer from nltk)

- Write all the above four tables into Excel Worksheet file named **ProcessedLoveOutput.xlsx**, with four sheets, one sheet for each table.

Submission:

Please name your python program as **YourName_HWOne.ipynb**, upload to your Google drive and share it with me at **yli@fordham.edu**. Please test your code before you submit it.

Grading Policy:

1. Good documentation.
    a. You should have the header block to show author, date, file description and so on.
    b. For each task completed, describe it.
    c. For each function you create, write appropriate comments.
    d. For major steps in your code, write appropriate comments.
2. Completeness of coding
    a. All tasks listed in the assignment should be finished.
3. Correctness of coding
    a. The output of each task should be correct.