Assignment 1

Submission date: End of day September 1, 2024

Use this partial data set (sourced from Kaggle) which contains research articles related to COVID-19. This corpus has around 56000+ files. In this assignment, you will perform a set of tasks as given below.

Tasks

1. Extract the text content from the JSON-encoded data set and create a text corpus. You may use any JSON library to extract the text (2 marks).
2. Develop your pre-processing steps (case-folding, removal of numbers, etc.) and order of steps (5 marks)
3. Find the weighted term frequency for every word in the corpus and order it according to its rank using Zipf's Law. Find the value of the α using the data obtained from the previous step (8 marks).
4. Print the number of tokens and the vocabulary (5 marks)
5. Plot Tokens Vs Vocabulary graph using Heaps' empirical law. Find Vocabulary count for every 10000 tokens. You may use a log scale for plotting (5 marks)

Note

1. To start with, try your program with a smaller corpus. You may proceed to extract all the content and perform the tasks to complete the assignment once your trial run is successful
2. Use Colab to submit your assignments. Learn all the tricks of Colab from the web, especially how to read remote files.
3. Share the final version of your assignments with the following email IDs:
   (a) ramaseshan.nlp@gmail.com
   (b) ambaye.om.cmi@gmail.com
   (c) vergil167867@gmail.com
   (d) rohitatcmi@gmail.com

4. Naming Conventions for the python notebooks (pynb):
   (a) The first part of the filename should be your First name.
   (b) The second part of the filename should be your roll number.
   (c) The third part of your assignment should be Assignment0X, where X is the assignment number.

   Example: `SriramBMC202204_Assignment01.ipynb`

5. We will not evaluate files with an arbitrary file name.

6. Write your official name :) at the beginning of the Python notebook.

7. Python notebooks should be available on the Colab platform (Google).

8. DO NOT send the Python notebook as an attachment to the shared email IDs.

9. DO NOT modify the code/results after the deadline.

10. Make sure that all the results are available when you share the assignments. Incomplete Python notebooks will not be evaluated.

11. We will NOT run/change your Python notebook.

12. Follow the pep-8 coding style.

13. Use functional-style of coding.

14. Use the multiprocessing library of Python wherever necessary.

15. Write at least 1-3 lines of comments for every function.

16. You may use NLTK or SpaCy library to pre-process the text. Do not use lemmetize or stemming functions to remove inflections.

17. You may use regex libraries to remove unwanted words/patterns from the corpus.

18. Keep the processed corpus safe. You may create a large single file or multiple text files. It/they will be useful for future your assignments.

19. Optional: You may use GitHub to store your versions of the assignment. Advantages - You will never lose your code if you check-in the code into the GitHub repository.

Sample Code

Extracting context from JSON formatted text

```
import json
def json2text(filename):
    file = open(filename)
    paper_content = json.load(file)
    body_text = ""
    abstract = ""
    title = ""
    #get the paper_id
    paper_id = paper_content['paper_id']
    if 'title' in paper_content:
        title = paper_content['title']
    #get the abstract
    if 'abstract' in paper_content:
        for abs in paper_content['abstract']:
            abstract = abstract + abs['text']
    # get the paper
    if 'body_text' in paper_content:
        for bt in paper_content['body_text']:
            body_text = body_text + bt['text']
    return (f'{title}  {abstract}  {body_text}').lower()
```

Parallel Write

```
import os
import multiprocessing as mp
from multiprocessing import Pool
def write_file(filename):
    with open(filename, 'r') as fd:
        json2text(filename)

def par_write(files):
    '''Read chank of files and let the cores of you machine
       do the job of format conversion in parallel'''
    #parameter: files - list of files from a folder

    cpu_count = os.cpu_count()

    p = Pool(processes=cpu_count)
    p.map(write_file,files, chunksize=16)
    p.close()
```