Assignment 4
Web Crawling and Extracting Information-Part1

Computing Lab-II
9th Feb 2022

This assignment is on crawling web pages and extracting the required information from them by creating suitable grammar rules.

Task 1 (Crawling worldometers website→ https://www.worldometers.info/coronavirus/)

1. Worldometer is a website where you will find all the coronavirus-related statistics world/continent/country-wise, like total cases, active cases, total death, new death, total recovered, serious/critical cases, total tests are done, etc.
2. You will be provided with a file named “worldometers_countrylist.txt,” which contains the continent-wise country's name.
3. Write a python code that reads the main URL & saves the page along with all the pages of countries present in the given file in HTML format.

Task 2 (Creating grammar and parsing the files)

1. Create grammar that can be used to extract the following fields for any country/continent/world -
    a. Total cases
    b. Active cases
    c. Total deaths
    d. Total recovered
    e. Total tests
    f. Death/million
    g. Tests/million
    h. New case
    i. New death
    j. New recovered

    Use yesterday’s data to answer the above queries. Also, provide the percent of total world cases for each of the queries.
2. You can ignore other fields except the above.
3. Now for a given country, you need to answer the queries below-[given time range]
    1. Change in active cases in %
    2. Change in daily death in %
    3. Change in new recovered in %
    4. Change in new cases in %
    5. Closest country similar to any query between 1-4

    Ask the user for the start and end date.

4. So basically you need to design a menu-driven program resolving user queries. We leave the design of the menu up to you, but keep in mind that your menu should be easy to use and address all the queries. The user should also be able to go back to the previous menu.

5. Write (python code using PLY) or (C code using Lex,Yacc) to extract the above fields. Your program should show all the possible query fields a user can ask for (from the

above list items).

6. Your program should also save the result in a log file as per the following format. <Word/Continent/Country> <Field_requested> <Field_value>

7. You have to think correctly about what kind of errors can come in the process and try to handle them. Note that you cannot use the “Beautiful Soup” python package for this assignment. Use the PLY package in python / Or you can code in C using lex and Yacc.

PLY ref: https://www.dabeaz.com/ply/

8. You can write a readme file to provide any particular instructions related to program execution steps, input format, or anything that you might think is useful for the evaluator while evaluating the assignment.

Deliverables:

1. Codes for task1 and task2, readme file if any.
2. Save this in a folder named in the format: <Roll No.>_CL2_A4. Compress this folder to zip format, creating a compressed file <Roll No.>_CL2_A4.zip. Upload this compressed file to moodle. Example: If your roll no. is 21CS60R05, the folder should be 20CS60R05_CL2_A4, and the compressed file should be 21CS60R05_CL2_A4.zip.
3. Not adhering to these instructions can incur a penalty.

Evaluation Scheme

Task1: 10 marks
Task2: 80 marks (all the fields grammar + correct output)
Error handling: 5 marks
Coding Style: 5 marks

Important Instructions

1. Plagiarism Rule: If your code matches (more than 50%) with another student’s code, all those students whose codes match will be awarded zero marks without any evaluation. Therefore, it is your responsibility to ensure that neither you copy anyone’s code nor anyone can copy yours.
2. Code error: If your code doesn’t run or gives an error while running, you will be awarded zero marks.