CS 210
Assignment 2 - Web scraping

April 2023

Overview

In this assignment, you will use web scraping tools to collect data about the diplomatic activities of a certain country. Each country usually has a website for their ministry of foreign affairs on which they share their diplomatic activities. Using the URL assigned to you, you are expected to scrape all the diplomatic activity news that that country has shared on their website. The assignment is divided into four tasks:

1. Collecting links to all individual news pages
2. Downloading the HTML pages of the collected links
3. Parsing the downloaded HTML files and extracting the required information
4. Analyzing the scraped data

Please enter your student ID in the ID variable in `config.py`.

1 Collecting links

The URL you are assigned will lead you to the web page of a certain ministry of foreign affairs, where a list of news about their diplomatic activities is available. Each item in the list has a link that leads to another web page, where more details are provided. In this task, you have to collect all of these links **from all pages** and save them in a .txt file. For this task, you will use the script in `download_links.py`. This file already has a structure to save the collected links in a specific file. What you need to do is fill the part responsible for collecting the links from the website **from all pages**. The part you need to fill is marked by the comment `# WRITE YOUR CODE HERE` and explanatory comments are there for guidance. The code is written in a way that collected URLs are saved one at a time. In addition, if an error occurs during scraping and the code stops running, it will start running from the last page you have reached to avoid

collecting links twice. This is a good practice when scraping data because there is a high risk of getting an error due to connection problems.
If necessary, you can modify the code inside the function, **but without changing the format of the saved `link_list.txt` file and without changing the main function name `download_links_from_index()`**. An example of the `link_list.txt` file is provided in the assignment material for guidance.

2 Downloading HTML pages

In this task, you have to download the HTML content of the links that you have collected in the first task. Downloading web page HTMLs and then parsing them locally is also a good practice in web scraping, because you may realise later that some information was wrongly extracted or that you need more information from the page that you have not extracted earlier. This task is relatively easy as most of the code is already prepared. All you need to do is modify the function `get_page_content()` in the file `save_html_pages.py` as described in the comments. If an error occurs while downloading the web page or while writing it, an empty file will be saved and the code will keep on running. Therefore, make sure that you check the sizes of the downloaded HTML files. In task 4 ”Analyzing scraped data”, you will plot the distribution of the file sizes like the one shown in Fig. 1. Notice that there are about 13 files having size 0 bytes in the example in Fig. 1, indicating that some files have not been downloaded correctly.

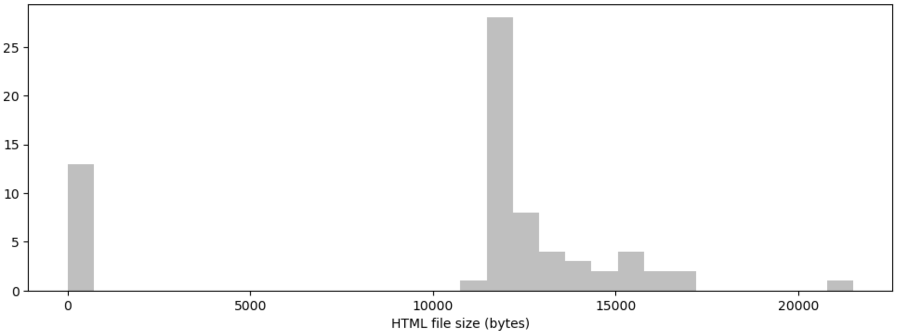

Figure 1: File size distribution

3 Parsing HTML pages

In this task, you have to read the downloaded HTML files, extract the required information from them, and save them in a JSON file. From each HTML page, you have to extract the date that the diplomatic activity occurred, the title of the news article, and the text content of the news article. In this task, you will modify the function `extract_content_from_page()` in the file `parse_html.py` as described in the comments. This code will keep running if an error occurred while parsing one of the HTML files. Therefore, make sure you check that the saved JSON file includes data for all HTML files.

4 Analyzing scraped data

In this task, you will do some analyses that will serve as a sanity check for the quality of the collected data to make sure that the scraping was done correctly. For this task, you will use the Jupyter notebook `DataAnalysis.ipynb` and perform the tasks defined in it.

Submission guidelines

- Submit one compressed file with the following files and the same structure:
  - `config.py`
  - `download_links.py`
  - `save_html_pages.py`
  - `parse_html.py`
  - `DataAnalysis.ipynb`
  - **data/**
    - `link_list.txt`
    - `parsed_data.jsons`
    - **raw_html/**: A folder containing all downloaded HTML pages.
- Name your submission (compressed file) yourSUid-hw2

Grading

In this assignment, we follow the motto “A job half done is as good as none.” We evaluate the code and the data collected for this assignment. Your grade will be computed by taking data completion (80%) and analysis conducted on `DataAnalysis.ipynb` (20%).