Indian Institute of Technology Bombay

Department of Computer Science

Project Report: Python Web Crawling

Thomas Biju Cheeramvelil
June 14, 2023

Contents

1 Introduction **2**

2 Project Scope **2**

3 Implementation Details **2**

4 Logic **2**

5 Functionality Overview **2**

6 Customization **3**

7 Conclusion **3**

8 Source Code **3**

9 References **3**

1 Introduction

Web crawling is a technique used to extract information from websites by systematically navigating through web pages and collecting data. It is an essential tool for various applications such as data mining, content indexing, search engine optimization, and website analysis. This project aims to develop a web crawling script that retrieves file sizes and provides insights into the file distribution and website structure.

2 Project Scope

The web crawling script focuses on the following key functionalities:

- Crawling a given URL up to a specified recursion level
- Retrieving file sizes of web pages and files
- Finding files and webpages of specifed extensions and domains
- Providing options for filtering URLs based on criteria like file extension and domain

3 Implementation Details

The web crawling script is developed using the Python programming language and utilizes various libraries and modules to achieve the desired functionalities. The following libraries are used:

- `argparse` for handling command-line arguments
- `requests` for making HTTP requests to fetch web pages
- `urllib.parse` for parsing and manipulating URLs
- `BeautifulSoup` for parsing HTML and extracting tags
- `functools` for functional programming operations

4 Logic

The web crawler implemented follows the below logical scheme for printing the webpage found for both finite and infinite level recursion:

- If at the ith recursion level, If at the ith recursion level, a webpage is being referred by multiple webpages then that webpage will be printed only once at that particular recursion level. Also, if a webpage is found at ith recursion level and then even if it is found again at jth recursion level where $j > i$ it will not be crawled again. However, the webpage will be printed again so as to indicate that it was also obtained at that recursion level.

5 Functionality Overview

The web crawling script consists of the following key functions:

- `unique_list(list1)`: Returns a list with unique elements by reducing duplicates.
- `get_links(url)`: Retrieves tags with `href` or `src` attributes from a given URL.
- `is_internal_link(domain, url)`: Checks if a given URL has the specified domain.
- `get_file_extension(url)`: Retrieves the file extension from a given URL.
- `crawl(url, threshold, output_file, domain2)`: Performs the web crawling process.
- `main()`: Handles command-line arguments and initiates the crawling process.

6 Customization

- Displaying files of specified domains:
  Only files of specified domains are displayed. Specify the domains after -d to use this function.
- Displaying files of specified extensions:
  Only files of specified extensions are displayed. Specify the domains after -d to use this function.
- Sorting Files with respect to Domain, Extension, or File Size:
  Sorting list of displayed files with respect to file size. This will not display the file size but only sort with respect to size. Specify -s along with -x in order to get file size as well.
- Displaying File Size:
  File or webpage size is displayed along with the url of each file or webpage. Specify -f to use this function.

Options	**Description**
-h, –help	show this help message and exit
-u URL, –url URL	URL to crawl
-t THRESHOLD	provide output file(s) for storing the result
-d [DOMAIN ...], –domain [DOMAIN ...]	URL domain needed
-e [EXTENSION ...], –extension [EXTENSION ...]	Extension needed
-s, –sort	Sorts on given basis
-f, –file_size	Displays file size

7 Conclusion

In conclusion, the web crawling project provides a valuable tool for extracting data and insights from websites. The implemented script offers functionalities for retrieving file sizes, analyzing file distribution, and exploring the structure of a website. With future enhancements, it has the potential to become a versatile and powerful web crawling solution for various applications.

8 Source Code

The source code of the web crawler can be found in this http://www.overleaf.comGitHub Repo .

9 References

- Python Software Foundation. *argparse – Parser for command-line options, arguments, and sub-commands.* Retrieved from `https://docs.python.org/3/library/argparse.html`
- Python Software Foundation. *urllib.parse – Parse URLs into components.* Retrieved from `https://docs.python.org/3`
- Requests: HTTP for Humans. *https://docs.python-requests.org/en/latest/*
- Beautiful Soup Documentation. Retrieved from `https://www.crummy.com/software/BeautifulSoup/bs4/doc/`
- functools — Higher-order functions and operations on callable objects. Retrieved from `https://docs.python.org/3/li`