International Journal of Scientific Research in Science and Technology
Print ISSN: 2395-6011 | Online ISSN: 2395-602X (www.ijsrst.com)
doi :https://doi.org/10.32628/IJSRST

Web Scraping For Book Recommendation System

Rasha Shaikh
Department of Computer Engineering, Savitribai Phule University, Pune, Maharashtra, India

Article Info
Volume9, Issue 2
Page Number: 552-556

Publication Issue
March-April-2022

Article History
Accepted :03April2022
Published :20April2022

ABSTRACT

The purpose of a book recommendation system is to predict buyer's interest and recommend books to them accordingly. Personal recommendation systems have been emerged to conduct effective search which mine related books based on user rating and interest. This paper proposed an effective system for recommending books for online users by providing the data which not only counts the ratings but also the users vote for the best books of 2022 along with their genre by using web scraping. Web scraping, also known as web extraction or harvesting, is a technique to extract data from the World Wide Web (WWW) and save it to a file system or database for later retrieval or analysis. Rather than using big data, smart data would work much better. The proposed system used Beautiful Soup designed and selenium web drivers for scraping HTML documents. Convenient Pythonic functions for navigating, searching, and modifying a parse tree; a toolkit for decomposing an HTML file and extracting desired information via html parser. The required data was successfully scraped or extracted and saved in csv file. Further a book recommendation model needs to be build using this dataset.

Keywords:
Web scraping, Beautiful Soup, Selenium, Web Drivers, HTML Parser, Data Extraction.

I. INTRODUCTION

Web scraping is a method used to get great amounts of data from websites and then data can be used for any kind of data manipulation and operation on it.

For this technique, we use web browsers. You usually do not have the built-in option to get that data you want. That is why we use Web Scraping to automate the process of getting that data and not having to do it manually. Web Scraping is the technique of automating this process so that instead of manually copying the data from websites.

This is accomplished either manually by a user or automatically by a bot or web crawler. Due to the fact that an enormous amount of heterogeneous data is constantly generated on the WWW, web scraping is widely acknowledged as an efficient and powerful technique for collecting bigdata. To adapt to a variety of scenarios, current web scraping techniques have become customized from smaller ad hoc, human-aided procedures to the utilization of fully automated

systems that are able to convert entire websites into well-organized data set.The purpose of this study is to scrape the best books of 2022 data from the Goodreads website for book recommendation system and convert it into a structured data which can be further used for analysis and building recommendation system.

II. METHODS AND MATERIALS

There are two essential modules of a web scraping program – a module for composing an HTTP request, such as Urllib2 or selenium and another one for parsing and extracting information from raw HTML code, such as Beautiful Soup or Pyquery. Here, the Urllib2 module defines a set of functions to dealing with HTTP requests, such as authentication, redirections, cookies, and so on, while Selenium is a web browser wrapper that builds up a web browser, such as Google Chrome or Internet Explorer, and enables users to automate the process of browsing a website by programming. Regarding data extraction, Beautiful Soup is designed for scraping HTML and other XML documents. It provides convenient Pythonic functions for navigating, searching, and modifying a parse tree; a toolkit for decomposing an HTML file and extracting desired information via lxml or html5lib.In the proposed study, I have used both the methods for retrieving data.

Web data was scrapped utilizing Hypertext Transfer Protocol (HTTP) and through a web browser. The process of scraping data from the Internet can be divided into two sequential steps; acquiring web resources and then extracting desired information from the acquired data. Goodreads website was used to scrape best books of 2022 data along with it's Book Title, Author name, Ratings and Genre.

Important libraries like requests, beautifulsoup and selenium were imported. The program was started by composing a HTTP request from goodreads website. This request was formatted in either a URL containing a GET query. Once the request was successfully received and processed by the goodreads website, beautiful soup was used to parse the text retrieved from the website.Book Title ,Author Name and Ratings were retrieved using find_all function.

A. CHALLENGES ENCOUNTERED

In the pursuit of finding the genre of a book, I stumbled upon a hurdle. My program was flaky while locating the genre element. It passed for some books and for some others it failed. We may classify this problem under the category of 'False Negative'.Although the genre of the book was present, the program result displayed it not to be present, thereby failing.My locator strategy being correct, the problem baffled me at the beginning.I decided to dig down to find the Root Cause.Since the strategy of 'requests' was headless, I could not visualize the real problem.At this juncture, Selenium WebDriver came to the rescue.

B. LITTLE ABOUT SELENIUM WEBDRIVER

Selenium WebDriver provides implementations that can help us visualize the proceedings of the program like I would do manually, also it is more powerful when it comes to parsing the DOM and applying waiting mechanisms.Changing the implementation for the extracting the genre part from 'requests and Beautiful soup' to 'Selenium WebDriver'.

C. BUILDING BLOCKS

I used CHROME as a browser and XPATH as our locator strategy. I also exploited the powers of Fluent Wait which would help us wait for the 'Genre' element to be located for a certain time.

D. BOOK GENRE EXTRACTION PROGRAM FLOW / MODULES

i. **Storing Book URLs:** Using Requests, hit the URL of the Good Reads Page. Parse the response HTML using Beautiful Soup 'html parser'. Find the number books present in the list using locators of beautiful soup's findAll method.

Iterate using a For Loop over the number of books to extract the 'anchor tag' thereby pulling the 'href' link of each book. Storing the book URLs in an array. Code Snapshot captured below in Fig. no.1:

Fig no.1

```
import requests
from bs4 import BeautifulSoup
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.desired_capabilities import DesiredCapabilities

bestbooks_url='https://www.goodreads.com/list/show/171064.Best_books_of_2022'
response = requests.get(bestbooks_url)
page_contents = response.text
soup = BeautifulSoup(page_contents, 'html.parser')
allrows = soup.findAll('tr', itemtype='http://schema.org/Book')
listhref=[]

for i in allrows:
    a_tag =i.a
    link=a_tag['href']
    listhref.append(link)
len(listhref)
```

ii. **Setting up Prerequisites for Selenium:** Downloading the required Chrome Driver.Defining the Desired Capabilities and Options for Chrome Driver: Page Strategy as normal. Maximized State of the Chrome Window. Incognito mode. Not loading images. Most important, Headless Mode of Operation. Code Snapshot captured below in Fig. No.2:

Fig no.2

```
caps = DesiredCapabilities().CHROME
caps["pageLoadStrategy"] = "normal"

options = webdriver.ChromeOptions()
prefs = {"profile.managed_default_content_settings.images": 2}
options.add_experimental_option("prefs", prefs)
options.add_argument('--headless')
options.add_argument('--start-maximized')
options.add_argument("--incognito")
```

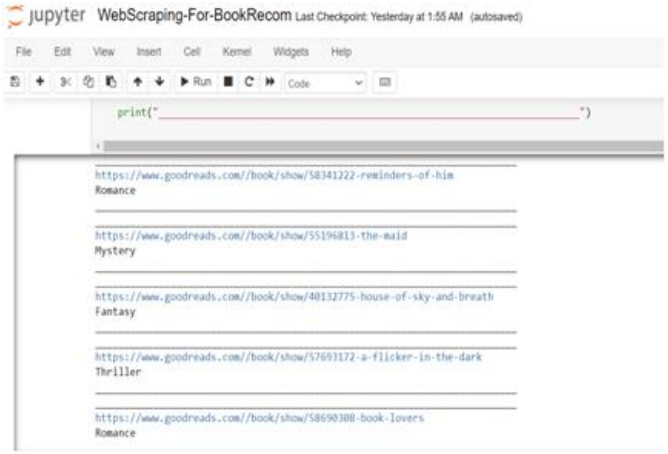
iii. **Go to Book URLs and extract Genres:** Hitting the Book URLs using looping mechanism.Looking whether the Genre Element is Present.If the Genre Element is not present, then find the 'sign-in' pop up.If the pop up is present, then refresh the page to bypass the pop up.Post refreshing, looking for the Genre Element with the new locator.Conditioning and waiting throughout, either looking out for the presence of pop up, or the two locators of genre elements. Indicating the user if no Genre is present for the book. Capturing the Genre Text corresponding to the Book URL. Repeating the above steps until the Genre of all books are captured. Code Snapshot Depicted Below in Fig.No.3 and Output in Fig.no.4:

Fig no.3

```
for i in listhref:
    driver = webdriver.Chrome(desired_capabilities=caps, options=options,
                              executable_path='C:\\Users\\Rahil\\PycharmProjects\\pythonProject\\chromed
    driver.get("https://www.goodreads.com/"+i)
    print("_____")
    print("https://www.goodreads.com/"+i)
    listgenre = driver.find_elements(By.XPATH,"+//a[@class='actionLinkLite bookPageGenreLink']")
    listpopup = driver.find_elements(By.XPATH,"/html/body/div[3]/div/div[1]/div/div/button")
    listrefreshgenre = driver.find_elements(By.XPATH, "//+[@id=\"__next\"]/div[1]/main/div[1]/div[2]/di
    if(len(listgenre)==0 and len(listpopup) == 1 and len(listrefreshgenre)==0):
        driver.refresh()
        try:
            WebDriverWait(driver, 100, poll_frequency=5).until(
            EC.presence_of_element_located((By.XPATH, "//+[@id=\"__next\"]/div[1]/main/div[1]/div[2]/di
            "Khatam")
            genretext = driver.find_element(By.XPATH, "//+[@id=\"__next\"]/div[1]/main/div[1]/div[2]/di
            print(genretext)
        except:
            print(("Genre is not mentioned for this book"))

    elif(len(listgenre)>0 and len(listrefreshgenre)==0):
        try:
            WebDriverWait(driver, 100, poll_frequency=5).until(EC.presence_of_element_located((By.XPATH,
                                                    'Khatam')
            genretext = driver.find_element(By.XPATH, "+//a[@class='actionLinkLite bookPageGenreLink']")
            print(genretext)
        except:
            print(("Genre is not mentioned for this book"))
```

Fig no.4



III. RESULTS AND DISCUSSION

A dataframe was created with attributes as Book Title, Author Name, Ratings and Genre using pandas libraries. This dataframe was then converted into csv file with hundred book details or hundred rows.Csv file snapshot is shown in Fig.no.4:

Fig no.4

jupyter Books.csv 2 minutes ago

File Edit View Language

```
Book Title,Author Name,Ratings,Genre
Reminders of Him,Colleen Hoover," 4.56 avg rating – 254,742 ratings",Romance
The Maid,Nita Prose," 3.89 avg rating – 116,121 ratings",Mystery
"House of Sky and Breath (Crescent City, #2)",Sarah J. Maas," 4.55 avg rating – 102,286 ratings",Fantasy
A Flicker in the Dark,Stacy Willingham," 4.03 avg rating – 58,659 ratings",Thriller
Book Lovers,Emily Henry," 4.48 avg rating – 71,355 ratings",Mystery
The Paris Apartment,Lucy Foley," 3.71 avg rating – 56,901 ratings",Romance
Olga Dies Dreaming,Xóchitl González," 4.03 avg rating – 14,935 ratings",Fiction
Sea of Tranquility,Emily St. John Mandel," 4.25 avg rating – 24,260 ratings",Fiction
The Golden Couple,Greer Hendricks," 4.04 avg rating – 50,819 ratings",Thriller
How High We Go in the Dark,Sequoia Nagamatsu," 3.94 avg rating – 10,653 ratings",Science Fiction
The Last House on the Street,Diane Chamberlain," 4.28 avg rating – 18,732 ratings",Historical
The Final Flaw,Michael R. Sullivan, 4.37 avg rating – 43 ratings,Science Fiction
Bloomsbury Girls,Natalie Jenner, 4.13 avg rating – 946 ratings,Historical
One Italian Summer,Rebecca Serle," 3.73 avg rating – 31,335 ratings",Fiction
The Diamond Eye,Kate Quinn," 4.35 avg rating – 17,328 ratings",Historical
The Overnight Guest,Heather Gudenkauf," 4.06 avg rating – 21,752 ratings",Thriller
"The Hourglass Throne (The Tarot Sequence, #3)",K.O. Edwards, 4.73 avg rating – 635 ratings,Fantasy
Lessons in Chemistry,Bonnie Garmus," 4.46 avg rating – 22,062 ratings",Fiction
"Where the Drowned Girls Go (Wayward Children, #7)",Seanan McGuire," 4.12 avg rating – 7,973 ratings",Fantasy
In Search of a Prince,Toni Shiloh, 4.26 avg rating – 517 ratings,Romance
Her Last Goodbye,Rick Mofina, 4.01 avg rating – 591 ratings,Mystery
The Book of Cold Cases,Simone St. James," 3.91 avg rating – 35,101 ratings",Mystery
The Violin Conspiracy,Brendan Slocumb," 4.20 avg rating – 8,154 ratings",Mystery
The Christie Affair,Nina de Gramont," 3.76 avg rating – 23,701 ratings",Historical
Reckless Girls,Rachel Hawkins," 3.63 avg rating – 49,309 ratings",Thriller
The World Cannot Give,Tara Isabella Burton, 3.53 avg rating – 650 ratings,Fiction
My Government Means to Kill Me: A Novel,Rasheed Newson, 4.26 avg rating – 31 ratings,Historical Fiction
Admissions: A Memoir of Surviving Boarding School,Kendra  James, 3.72 avg rating – 999 ratings,Nonfiction
Sister Mother Warrior,Vanessa Riley, 4.35 avg rating – 17 ratings,Historical
Tell Me the Truth,Kiersten Modglin," 3.91 avg rating – 2,592 ratings",Audiobook
```

IV. CONCLUSION

While this project may not be as sophisticated as web scrapers made by large corporations, there is enough scope in this application to make a decent impact in the world of book recommendation. By first scraping and then utilizing a set of information like genre and ratings, users may be recommended books based on collaborative andcontentbased recommendation techniques, that would help both the users and the business. Users' search time for the right book may be significantly reduced, thereby the saved time may be invested in reading the recommended book.

V. REFERENCES

[1]. Saurkar, Anand V., Kedar G. Pathare and Shweta A. Gode, An Overview On Web ScrapingTechniques And Tools, International Journal on Future Revolution in Computer Science&Communication Engineering, pages 363-367, 2018.

[2]. Liu B., Sentiment Analysis and Subjectivity, Handbook of Natural Language Processing, pages 627- 666, 2010.

[3]. PratikshaAshiwal, S.R. Tandan, Priyanka Tripathi and Rohit Miri, Web Information Retrieval UsingPython and BeautifulSoup, International Journal for Research in Applied Science &EngineeringTechnology (IJRASET), pages 335-339, 2016.

[4]. Rahul Dhawani, Marudav Shukla, Priyanka Puvar, Bhagirath Prajapati, A Novel Approach toWebScraping Technology, International Journal of Advanced Research in Computer Science and SoftwareEngineering, Volume 5, Issue 5, 2015.

TABLE NO.1. Chronological Order: Discovery of problem navigating through them successfully

PROBLEM	IMPLEMENTED SOLUTION
The Good Reads page throws the registration pop-up, once the user has accessed around 10 books.	Using Selenium, I checked for the presence of a sign-in pop-up on the page. If a pop up was encountered, page refresh function of selenium was invoked, resulting in the page being displayed in its normal state, without the sign-in pop up bothering us
Selector of the "Genre Element" changing on page refresh	When the Good Reads page was refreshed, to deal with the 'sign-in' pop-up, the program could no longer identify that 'Genre Element' which it was seamlessly able to find, pre-refresh. On Printing the HTML code in the 'except' snippet, I observed that the locator of the 'Genre Element' has changed. I handled this using programming conditional statements on 'locators' - pre and post refresh.
Some Good Read books were in languages other than English, and did not have a Genre associated with it.	This is a special and rare occurrence on the Good Reads page for a book not to have a Genre. This was handled, using conditions statements.
Program was trying to find the Genre Element before the entire page loads.	Selenium's Desired Capabilities allows to set the 'page-strategy' to 'normal' which will allow the program to run only after the page has fully loaded.
Program execution was slow	To improve the speed of the program, Selenium helped us to have some options with Chrome Browser, in which I chose not to load images of the page, because Image takes more time to load than text. Secondly, Selenium also gives us the options to run the Chrome Browser in Headless mode. Both these helped us to fasten the execution by 50% with regards to execution time.