OPEN ACCESS

Manuscript ID:
ASH-2021-08033588

Volume: 8

Issue: 3

Month: January

Year: 2021

P-ISSN: 2321-788X

E-ISSN: 2582-0397

Received: 26.10.2020

Accepted: 30.11.2020

Published: 01.01.2021

Citation:
Midhu Bala, G., and K. Chitra. "Data Extraction and Scratching Information Using R." *Shanlax International Journal of Arts, Science and Humanities*, vol. 8, no. 3, 2021, pp. 140-144.

DOI:
https://doi.org/10.34293/sijash.v8i3.3588

Data Extraction and Scratching Information Using R

G. Midhu Bala
Assistant Professor, Department of Computer Science
Mangayarkarasi College of Arts and Science for Women, Madurai, Tamil Nadu, India
https://orcid.org/0000-0001-9751-2739

K. Chitra
Assistant Professor, Department of Computer Science
Government Arts College, Melur, Madurai, Tamil Nadu, India

Abstract
Web scraping is automatic process of extracting multiple Web pages from the World Wide Web. It is a field with active developments that shares a common goal with text processing, the semantic web vision, semantic understanding, machine learning, artificial intelligence and human- computer interactions. Current web scraping solutions range from requiring human effort, the ad-hoc, and to fully automated systems that are able to extract the required unstructured information and convert into structured information, with restrictions. A method for budding a web scraper using R programming which locates files on a website, then extracts the filtered data and stores it is explained in this paper. The modules, algorithm for automating the navigation of a website through links are mentioned in this paper. Further it can be used for data analytics.
Keywords: Web scraping, Web mining, Locating files in websites, R programming, R vest, Web Crawling

Introduction

Data are universal on the Internet, Searching the web for useful data and information for analysis has become a routine job. The data on the websites are found in unstructured format such as tables, articles, comments, nested in different HTML tags, etc. Gathering a vast amount of data from the web is not simple task, but it is a fine way to gather information which can be used for future analysis. In this paper a new methodology is introduced to deal with the process of web scraping data from dissimilar locations on the Internet and to store it in database.

Web scraping play a vital role in growing businesses; harvesting big data is considered a necessary requirement for staying in the market. The web is like an endless ocean with lot of unstructured data, and this data comes unexplored possibilities.

Definition

Web Scraping, Web Data Extraction, Web Harvesting is a technique employed to extract large amount of unstructured data from websites, saved to a local file or to a database in structured format.

Essential of Web Scraping

- **Price Comparison**: Collect data from online shopping websites and use it to compare the prices of products.
- **Email address gathering**: Companies collect email ID and send bulk emails for marketing.

- **Social Media Scraping**: To know the new trending, for sentiment analysis data is collected from Social Media websites such as Twitter, facebook., etc.,.
- **Research and Development:** To analyse data, carry out Surveys and for R&D large set of data is collected through web scraping from websites.
- Job listings: Details regarding job openings, interviews are collected from different websites and listed in one place to access easily by the user.
- **Movie Review**: Scraping data to compare different movies, medicines etc.
- Image Classification: Scraping an image from different websites to train the image
- **E-commerce:** Scrapping user reviews and feedbacks from Flipkart and Amazon etc, to improve the sale.

Types of Web Scrapping

Human Copy-Paste: Well-organized and slow way of scraping data from the web involves human by copy-paste the data from different websites.

Text pattern matching and grep: Simple, powerful approach to extract information from the web using regular expression matching facilities from open source like UNIX, LINUX grep command of programming languages.

API Interface: Many websites like Facebook, Twitter, Linked In, etc. Provides public, private APIs which can be called using the standard code for retrieving the data in the prescribed format.

DOM Parsing: By using the web browsers, programs can retrieve the dynamic content generated by client-side scripts.

HTTP Programming: Static and dynamic web pages can be retrieved by posting HTTP requests from the remote web server using socket programming.

Proposed Methodology

This section specifies the steps in making a web scraper and searches the website that contains the data required for analytics. The application begins with a URL that contains the data we want to search. It obtains the contents of the webpage pointed to by the URL and extracts all the information from it. The data is in unstructured format. Using the selector gadget data required is selected and the following steps explain the implementation details to accomplish the above mentioned tasks:

Step 1: Generate the URL of a website from which you want to extract data.

Step 2: Install the package rvest. Store the URL generated in the previous step in a variable. The read the contents of the web page and store it.

Step 3: Find the data to extract and select it using selector gadget and copy the path and store it.

Step 4: Read the required data from all the pages and store it. Set the working directory to store the csv file.

Step 5: Write the data in the csv file. Read the data from the csv file for analysis.

Step 6: Analyze the data and plot the histogram.

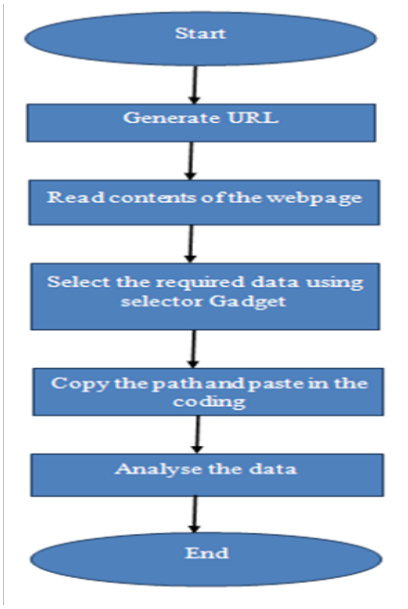


Figure 1: Flow Diagram

Data Set & Tools for Web Scraping

Data Set

In this paper search result of "mobile" in flipkart.com is used for analysis. Name of the mobile, price and ratings are extracted using r code.

Tool: R programming

Function: rvest

The rvest package is the workhorse toolkit used to extract the unstructured data from the web. Read

the content of the webpage using the function read_html(). This function will download the webpage and store it to for it so that rvest can navigate it.

Select the elements you want using the function html_nodes(). This function will take an HTML object (from read_html) along with a CSS or Xpath selector (e.g., p or span) and save all the elements that match the selector. This is where SelectorGadget can be helpful.

Functions like html_tag(), html_text(), html_attr()and html_attrs() used to extract data selected from the nodes.

Results and Discussion

Figure 1: The entire methodology is explained in the flow diagram for easy understanding.

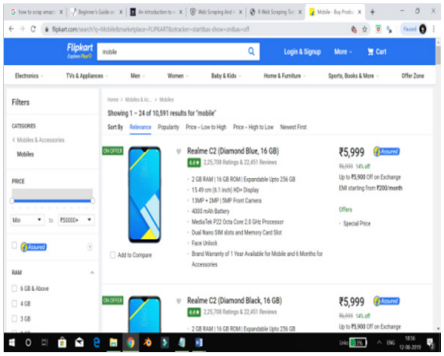


Figure 2: Flipkart Website

Figure 2 shows the URL in which the data to be extracted.

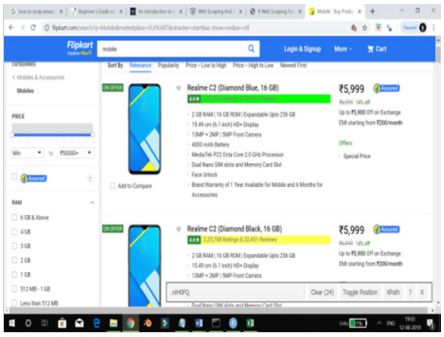
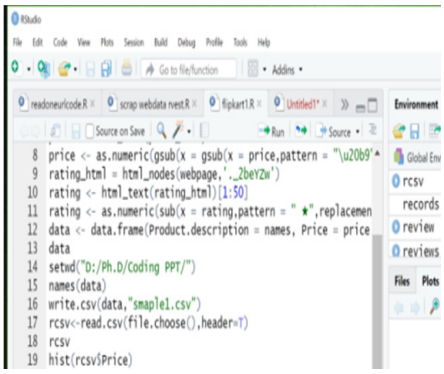


Figure 3: Data for analysis

Figure 3, shows the data which is to be used for analysis are highlighted.



```
price <- as.numeric(gsub(x = gsub(x = price,pattern = "\u20b9"
rating_html = html_nodes(webpage,'._2beYZw')
rating <- html_text(rating_html)[1:50]
rating <- as.numeric(sub(x = rating,pattern = " ★",replacemen
data <- data.frame(Product.description = names, Price = price
data
setwd("D:/Ph.D/Coding PPT/")
names(data)
write.csv(data,"smaple1.csv")
rcsv<-read.csv(file.choose(),header=T)
rcsv
hist(rcsv$Price)
```

Figure 4: R Code

Figure 4 shows the sample coding is displayed in this figure.

		Product.description	Price	Rating
1		Redmi Note 5 Pro (Black, 64 GB)	14999	4.5
2		Asus Zenfone Max Pro M1 (Black, 32 GB)	10999	4.2
3		Redmi Note 5 Pro (Gold, 64 GB)	14999	4.5
4		Samsung Galaxy J6 (Black, 32 GB)	12990	4.4
5		Asus Zenfone Max Pro M1 (Black, 64 GB)	12999	4.3
6		Infinix HOT 6 Pro (Sandstone Black, 32 GB)	7999	4.3
7		Honor 7A (Black, 32 GB)	8999	4.2
8		Honor 7A (Blue, 32 GB)	8999	4.2
9		Honor 7A (Gold, 32 GB)	8999	4.2
10		Redmi Y1 (Grey, 32 GB)	8999	4.2
11		Redmi 5A (Rose Gold, 16 GB)	5999	4.4
12		Redmi 5A (Grey, 16 GB)	5999	4.4
13		Redmi 5A (Gold, 16 GB)	5999	4.4
14		Samsung Galaxy On6 (Blue, 64 GB)	14490	4.3
15		Redmi 5A (Gold, 32 GB)	6999	4.4
16		Redmi 5A (Blue, 16 GB)	5999	4.4
17		Redmi 5A (Rose Gold, 32 GB)	6999	4.4
18		Redmi 5A (Blue, 32 GB)	6999	4.4
19		Redmi 5A (Grey, 32 GB)	6999	4.4
20		Samsung Galaxy J8 (Blue, 64 GB)	18990	4.4
21		Asus Zenfone Max Pro M1 (Grey, 64 GB)	14999	4.5
22		Redmi Note 5 (Gold, 32 GB)	9999	4.4
23		Asus Zenfone Max Pro M1 (Grey, 32 GB)	10999	4.2
24		Samsung Galaxy On6 (Black, 64 GB)	14490	4.3

Figure 5: Extracted Data

Figure 5 shows the output of rvest is displayed in this figure.

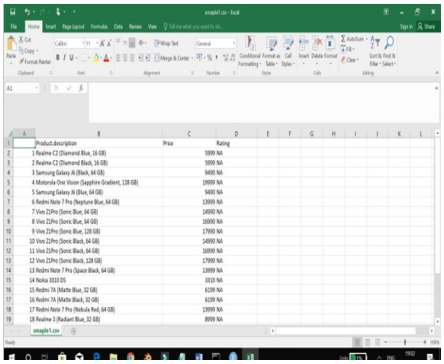


Figure 6: csv file

Figure 6 shows the extracted data is converted into csv file and some is given in this figure.

Conclusion and Future Work

Information retrieval from web is one of the challenging task for researchers because it is dynamic. In Today's era increasing use internet,

social media services are turning towards analysis of big data. Web information is mostly unstructured format, the developed method is useful to retrieve the unstructured data and make it useful for the analysis.

In future this method can be used to filter data such as year wise and can be enhanced to handles infinite loops while using links to traverse websites. Additional techniques to handle pagination in web pages can being corporate and it can be stored in database.

References

Acar, Gunes, et al. "Fpdetective: Dusting the Web for Fingerprinters." *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, 2013, pp. 1129-40.

Ashiwal, Pratiksha, et al. "Web Information Retrieval Using Python and BeautifulSoup." *International Journal for Research in Applied Science & Engineering Technology,* vol. 4, no. 4, 2016, pp. 335-339.

Beno, Miloslav, et al. "AgentMat: Framework for Data Scraping and Semantization." *International Conference on Research Challenges in Information Science,* 2009.

Case, Karl E., et al. "Comparing Wealth Effects: The Stock Market versus the Housing Market." *The BE Journal of Macroeconomics*, vol. 5, no. 1, 2005.

Castrillo-Fernández, Osmar. *Web Scraping: Applications and Tools*, European Public Sector Information Platform, 2015,

Doran, Derek, and Swapna S. Gokhale. "Web Robot Detection Techniques: Overview and Limitations." *Data Mining and Knowledge Discovery*, vol. 22, 2011, pp. 183-210.

Fisher, Danyel, et al. "Terms of Service, Ethics, and Bias: Tapping the Social Web for CSCW Research." *CSCW*, 2010, pp. 603-606.

Grasso, Giovanni, et al. "Effective Web Scraping with OXPath." *Proceedings of the 22nd International Conference on World Wide Web,* 2013, pp. 23-26.

Gu, Chengjian, and Lucheng Huang. "Web Mining in Technology Management." *International Seminar on Business and Information Management*, 2008.

Kolari, Pranam, and Anupam Joshi. "Web Mining: Research and Practice." *Web Engineering*, 2004, pp. 49-53.

Kosala, Raymond, and Hendrik Blockeel. "Web Mining Research: A Survey." *ACM SIGKDD Explorations*, vol. 2, no. 1, 2000, pp. 1-15.

Liu, John Chung-En, and Bo Zhao. "Who Speaks for Climate Change in China? Evidence from Weibo." *Climatic Change*, vol. 140, 2016, pp. 413-422.

Mahto, Deepak Kumar, and Lisha Singh. "A Dive into Web Scraper World." *International Conference on Computing for Sustainable Global Development*, 2016.

Malik, Sanjay Kumar, and Sam Rizvi. "Information Extraction using Web Usage Mining, Web Scrapping and Semantic Annotation." *2011 International Conference on Computational Intelligence and Communication Systems,* 2011, pp. 465-469.

O'Reilly, Sean. "Nominative Fair Use and Internet Aggregators: Copyright and Trademark Challenges Posed by Bots, Web Crawlers and Screen-Scraping Technologies." *Loyola Consumer Law Review*, vol. 19, no. 3, 2007, pp. 273-288.

Penman, Richard Baron, et al. *Web Scraping Made Simple with SiteScraper.*

Sacramento, Clara, and Ana C.R. Paiva. "Web Application Model Generation through Reverse Engineering and UI Pattern Inferring." *International Conference on the Quality of Information and Communications Technology,* 2014.

Schrenk, Michael. *Webbots, Spiders, and Screen Scrapers: A Guide to Developing Internet Agents with PHP/CURL*, No Starch Press, 2007.

Sirisuriya, S.C.M. "A Comparative Study on Web Scraping." *Proceedings of 8th International Research Conference*, 2015, pp. 135-140.

Vargiu, Eloisa, and Mirko Urru. "Exploiting Web Scraping in a Collaborative Filtering-based Approach to Web Advertising." *Artificial Intelligence Research*, vol. 2, no. 1, 2013, pp. 44-54.

Vasani Krunal, A. "Content Evocation Using Web Scraping and Semantic Illustration." *IOSR Journal of Computer Engineering*, vol. 16, no. 3, 2014, pp. 54-60.

"Web Scraping." *Wikipedia*, https://en.wikipedia.org/wiki/Web_scraping

"Web Scraping Software." *Web Data Scraping*, http://webdata-scraping.com/web-scraping-software/

Yi, J., et al. "Sentiment Analyzer: Extracting Sentiments about a Given Topic using Natural Language Processing Techniques." *Third IEEE International Conference on Data Mining*, 2003.

Author Details

G. Midhu Bala, *Assistant Professor, Department of Computer Science, Mangayarkarasi College of Arts and Science for Women, Madurai, Tamil Nadu, India,* ***Email ID****: midhug.research@gmail.com.*

Dr. K. Chitra, *Assistant Professor, Department of Computer Science, Government Arts College, Melur, Madurai, Tamil Nadu, India.*