Web Scraping Techniques and Applications - Literature Review

Written by: Chaimaa Lotfi and Swetha Srinivasan

Supervised by: Professors Myriam Ertz and Imen Latrous

June 2021

Table of contents :

1 Introduction

Data has a vital role in business, marketing, engineering, social sciences, and other disciplines of study since it may be used as a basic reference in all activities that include the use of information and knowledge. Data collection is the first stage of research, followed by the systematic measurement of information about important factors, allowing one to answer inquiries, formulate research questions, test hypotheses, and assess outcomes.

Data collection methods differ depending on the discipline or field of study, the nature of the information sought, and the user's aims or objectives. The method's application methodology can also change, depending on the goals and conditions, without jeopardizing data integrity, correctness, or reliability[1]. On the Internet, here are numerous data sources that can be employed in the research process. The technique of extracting data from websites is often known as web scraping, web extraction, web harvesting, web crawler.

The basic design of a web scraper is as follows :

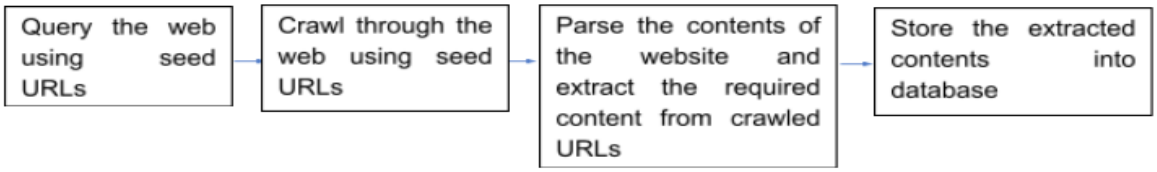

FIGURE 1 – The basic design of a web scraper

Web scraping is widely used for a variety of purposes, including online price comparison, weather data monitoring, website change detection, research, integrating data from multiple sources, extracting offers and discounts, scraping job postings information from job portals, brand monitoring, and market analysis[26].

It is also used as a means of data collection in a fast and efficient manner. Web scraping has myriad applications in various domains. It acts as a prerequisite to big data analytics. Discussed below are few of the several domains where web scraping is used.

1.1 In healthcare :

Healthcare is no longer a sector relying solely on person to person interaction. It has gone digital in its own way. In this data-centric setup, web scraping in healthcare can improve lots of lives by

providing rational decisions.
Healthcare workers typically regard data collecting involving a large number of patients as a laborious and time-consuming process. Even while clinical data is now needed more than ever, the current patient load makes gathering it nearly impossible. To that end, the author proposes deploying a system that automatically and autonomously retrieves clinical data from SARS-CoV2 patients who come to the hospital and collects it for future research[32].

Another application of web data extraction techniques in the healthcare domain is a research conducted by Dascaru et al.[31], where crawlers are utilized to extract drug leaflets.

1.2 In social media :

Extracting data from social media proves to be a great help in improving the marketing campaigns for companies. In this fast-paced world, the companies can quickly analyze the sentiment of the customers towards their products, improve public relations and audience engagement.

For this purpose they used a web scraping method to create a web-based Instagram account data download application that can be used by many parties. Researchers chose the web scraping method because it eliminates the need to use Instagram's Application Programming Interface (API), which has access restrictions in retrieving data on Instagram. The web scraping method was successfully utilized to create an Instagram account data downloader application. In this study, application testing was conducted on 15 Instagram Accounts with a range of publications ranging from 100 to 11000. According to the results of the data analysis, the web scraping implementation was able to successfully download Instagram account data for a total of 2412 accounts. Users can download Instagram account data to a data collection and manage it in this app, including deleting and exporting data collections in CSV, Excel, or JSON formats[32].

1.3 In finance :

The author proposed a first approach to develop web-based innovation indicators that could address some of the drawbacks of existing indicators. In particular, they created a strategy for identifying product innovator enterprises on a wide scale at minimal cost. Then, trained an artificial neural network classification model using labelled (product innovator/no product innovator) online texts of surveyed firms using traditional firm level variables from a questionnaire-based innovation survey (German Community Innovation Survey). Following that, they used their categorization model to forecast whether or not hundreds of thousands of German companies are product innovators by analyzing their online texts. Next ,they compared their predictions to firm-level patent statistics, benchmark data from survey extrapolation, and regional innovation indicators.The findings suggest

that this method generates solid forecasts and has the potential to be a useful and cost-effective addition to the existing set of innovation indicators, particularly given its breadth and geographic granularity[15].

The research conducted by Tharanya et al.[17], uses technical analysis of news articles scraped from the Internet. The news is extracted from a reliable website and the contents of the website is summarized to perform analysis and event modelling.

1.4 In marketing :

Boegershausen et al. [38] in the report talk about vast amount of customer data in the form of digital footprint available to analyze customer behaviour and to answer customer research questions.

Saranya et al. [39] in their paper, propose to predict customer purchase intention during online purchases using machine learning models. The data is collected using web scraping since the information on web is in an unstructured format. The data is further analyzed to predict the purchase intent.

Nguyen et al. [40] analyze social media engagement of Australian SMEs using web scraping. They collect the data from Instagram using Instagram API and use the data to further find that tagging instead of hashtags garners more engagement as it is more trustworthy.

1.5 Others :

Deng in his paper has used web data extraction techniques to extract information on mineral intelligence in China [11]. Kotouza et al [16]. have taken advantage of web data extraction techniques to design a system which acts as an assistant to a fashion designer to provide information about newest fashion trends improving customization. In [23], the authors have used the information available on the Internet to extract forestry information features. Based on the reviews published in the web, the authors Yaroslav et al.[20], performed the task of analysing traffic safety in Northwestern Federal District using Python libraries Scrapy to scrape the reviews from the Internet.

1.6 In research :

Authors Suganya et. al., [2] in their paper use web scraping for web citation analysis which helps researchers in finding related papers for further analysis. They study and compare three methods namely Particle Swarm Optimization, Hidden Markov Model algorithm and Firefly Optimization algorithm based web scraping for extracting web citation information based on the given user query. Based on their experiments it is found that Firefly Optimization Algorithm based web scraping (FOAWS) performs better than the rest of the techniques.

Similarly, authors Rahmatulloh et al. [13] in their paper employ HTML DOM based web scraping to make recapitulations of scientific article publications from Google Scholar to aid in research

studies. The recapitulations are further programmed to be presented as a report either in the form of PDF or Excel file.

The authors Kolli et al. in [7] present a customized news Internet searcher that centers around constructing a storehouse of reporting stories by relating proficient mining of content data from a network information sheet from shifted e-information entrances.

In [36], the author proposes to reduce the time required to identify the research gap using web scraping and natural language processing. This approach is tested by reviewing three distinct areas namely safety awareness, housing price, sentiment and AI. The titles of the publications are scraped from Google Scholar and using tokenization, the titles are parsed. By ranking the collocations from highest to lowest frequency, the set of keywords that are not used in the paper title is obtained and the research void is determined.

In [37], the paper presents a scholarly production data-set focused on COVID-19 to provide an overview of scientific research activities, making it easy to identify countries, scientists and research groups most active in this task force to combat the coronavirus disease. A dataset containing 40,212 records of articles' metadata are extracted from Scopus, PubMed, arXiv and bioRxiv databases from January 2019 to July 2020 using the techniques of Python Web Scraping and pre-processed with Pandas Data Wrangling using a pipeline versioned with the Data Version Control tool (DVC) making it easy to reproduce and audit. To extract data from PubMed and Scopus, API were used and Scrapy was used for scraping data from arXiv and bioRxiv databases.

2 Web Scraping methods

Web scraping is the process of automatically mining data or collecting information from the Internet and other common databases. In multiple researches, different web scraping methods have been developed, including :

2.1 Traditional copy and paste :

The copy-pasting method is simple to use : access the page in your browser, then manually copy and paste it onto other media. This method is easy and straightforward, however it cannot be used if the website employs a barrier program[1], which requires human selection of objects or sentences that are somewhat long. While other methods are more difficult to utilize and necessitate the usage of an extra program.

2.2 HTML parsing :

Large collections of pages are generated dynamically from an underlying structured source, such as a database, on many websites. A common script or template is used to encode data from the same

category into similar pages. A wrapper is a program in data mining that recognizes such templates in a given information source, extracts its content, and converts it into a relational form[10]. Wrapper generation techniques presume that a wrapper induction system's input pages follow a common pattern and can be easily identified using a common URL scheme. Furthermore, semi-structured data query languages like XQuery and HTQL can be used to analyse HTML websites and retrieve and change their content[27][28].

2.3 DOM parsing :

Programs can obtain dynamic material generated by client-side scripts by embedding a full-fledged web browser, such as Internet Explorer or the Mozilla browser control. These browser controls also parse web pages into a Document Object Model (DOM) tree, from which applications can extract sections of the pages [10][28]. Also, a tree structure Document Object Model can be used to represent a web page. It translates and saves a specified website address page into a DOM tree from a search engine.
This method provides a lot of flexibility and agility, if it's on the page, it can be tracked without having to wait for the web development team to expose it through the data layer.

2.4 HTML DOM :

The HTML DOM (Hyper Text Markup Language Document Object Model) is a standard for obtaining, altering, adding, or deleting HTML elements [29]. By defining objects and properties for all HTML components, as well as ways to access them, DOM efficiency can be improved. JavaScript can access all elements in an HTML document using the DOM. To access objects, the HTML DOM employs computer languages, most often JavaScript[29].
Every HTML element is considered as an object. Each object's method and property make up the programming interface[1][29].

2.5 Regular Expression (Regex) :

Regex is a formula that explains a group of words that spans numerous alphabets and follows a precise pattern. It can be used to match specific character patterns across several strings. Ordinary characters and metacharacters are the two sorts of regular expressions[1]. Some of these patterns look pretty strange because they contain both the material to match and special characters that modify how the pattern is perceived. Regular expressions are a must-know tool for parsing string data and should be learned at the very least at a basic level[35].

2.6 XPath :

The main component of the XSLT standard is XPath (Stylesheet Language Transformation). In eXtensible Markup Language (XML) documents, XPath can be used to explore elements and attributes [27]. XPath is a node selection language for XML documents that may also be used with HTML. The location path is the most useful XPath expression. To identify a set of nodes in a document, a path location uses at least one step location. A location path that selects the document root node is the simplest. This road is merely a slash "/" in the middle. The symbol is the root of a Unix system file as well as a document's root node.

2.7 Vertical aggregation platform :

Several companies have created vertical-specific harvesting platforms. With no "man in the loop" (direct human interaction) and no effort tied to a single target site, these systems build and monitor a slew of bots for specific verticals. The preparatory phase entails creating a knowledge base for the entire vertical, after which the platform builds the bots on its own. The resilience of the platform is determined by the quality of the data it retrieves (typically the amount of fields) and its scalability (how quick it can scale up to hundreds or thousands of sites). This scalability is mostly utilized to target the Long Tail of sites that are too difficult or time-consuming for traditional aggregators to extract content from.

2.8 Semantic annotation recognizing :

Metadata, semantic markups, and annotations may be included on the scraped pages, which can be utilized to discover specific data snippets. This technique can be considered as a specific case of DOM parsing if the annotations are incorporated in the pages, like Microformat does. In another case, the annotations are saved and handled independently from the web pages, arranged into a semantic layer, so scrapers can acquire the schema and instructions from this layer before scraping the pages.

2.9 Computer vision web-page analyzer :

Machine learning and computer vision are being used in an attempt to recognize and extract information from web pages by visually analyzing them as a human would. A computer vision-based system is used to analyze the semantic structure of web pages based purely on an image of the rendered page, and a rich representation of the page is produced as a tree of regions labelled according to their semantic role.

2.10 Comparison between web scraping methods presented above :

The comparison is carried out by putting each approach to the test when retrieving data from the target website, then measuring and comparing the results. The experiment's measuring parameters are process time, memory usage, and data consumption. In comparison to the HTML DOM approach and Xpath, the results of the experiment reveal that web scraping with the Regex method uses the least amount of RAM. When compared to Regular Expression and Xpath techniques, HTML DOM takes the least amount of time and consumes the least amount of data[1].

3 Web Scraping technology

3.1 Web crawlers :

Web crawler is a bot that visits websites and extracts data from them. A web crawler, according to Mahto and Singh (2016), works by loading a tiny list of links. The program then looks for more links on those pages and adds them to a new list called crawl frontier for further exploration. The crawler must determine whether a URL is absolute or relative. In the case of relative URLs, the crawler must first determine the URL's base [10]. In order to extract and store data efficiently, a decent crawler must be able to recognize circular references and minor modifications of the same page.
There are several types of web crawlers namely :

1. Focused crawler :
   Focused web crawler selectively searches for web pages relevant to specific user fields or topics. It attempts to obtain more relevant pages with a higher level of accuracy. It only downloads relevant pages and avoids pages not related to the subject. This is achieved by prioritizing websites.
2. Incremental crawler :
   Web crawlers known as incremental crawlers are designed to visit and access updated web pages. Incremental crawlers update the content of websites by visiting them frequently and storing the updated version of pages.
3. Distributed crawler :
   Distributed crawlers assign crawling to other crawlers. A central server in remote areas communicates and syncs with the nodes.
4. Parallel crawler :
   Multiple crawler processes are combined to make a parallel crawler where each process performs the process of filtering and retrieving the URLs and the URLs are collected from each process.

5. Hidden crawler :
   The content which is behind websites which are not accessible to general users is known as hidden web. The crawler which collects this data is known as hidden crawler[25].

Table1 : Comparison between web crawler types [25]

Parameters	Hidden crawler	Distributed crawler	Incremental crawler	Parallel crawler	Focused crawler
Freshness	No	No	Yes	No	No
Search technique	DFS[1]	BFS[2]	BFS	BFS	DFS
Network load reduction	-	No	No	Yes	-
Scalability	Yes	No	No	Yes	Yes
Extensibility	-	No	Yes	No	-
Overlapping	-	No	No	Yes	-
Selection of pages	Form analyzer	From seed URLs	From priority queue	From seed URLs	Related to specific topic

[1] DFS : Depth First Search [2] BFS : Breadth First Search

3.2 Web Scraping parsers :

Web scrapers must use parsers in order to extract useful information from scraped data. Programmers use them to format and extract certain details from data, such as a CV parser that can extract a person's name and contact information from an email's text. Simple HTML parser functionality is included in most Web Scraping libraries. Parsers for particular data such as PDF, CSV, QR code, or JSON are also available. Parsers are built into real web browsers like Firefox and Chrome. Web scraping done with a genuine web browser can also take advantage of the browser's built-in parser.

3.3 Web Scraping policies :

Selection, re-visit, politeness, and parallelization, according to Mahto and Singh (2016), are the four fundamental policies that a crawler must follow in order to act efficiently[10]. The crawler can eliminate most of the useless links and considerably reduce its search space by focusing on vital links first. When pages are dynamic, the crawler must check for updates on a regular basis.

4 Development of web scraping tools

The web data extraction tool can be tailor made for each specific application. The following section discusses how the web data extraction tool has been built using different techniques.

In [2], the authors collected information from 12,250 web pages from a Google Scholar web citation database through web scraping and crawling. Author information and paper details are extracted and stored as a .csv file. The user query to Google scholar is considered as the Seed URL for the web crawler. The web crawler then crawls the HTML pages and download the user required content. The citation information and the links are extracted from the URL and stored in the database. The web scraper selects the citation content using a selector gadget and scrapes citation information from a given URL. After parsing the information, it is extracted from the web document. The collected or scraped content is then filtered by keywords or by matching a specific pattern and stored into a structured format of a .csv file. The proposed algorithm is the combination of web scraping and firefly optimization algorithm. The web scraping technique scraps/extracts the citation information from the web but firefly algorithm assigns random values then updates the light intensity and checks relevancy of title of the paper and user query at every time. Hence, this proposed algorithm extracts the information with higher accuracy.

4.1 Web scraping using PHP :

In [3], a daemon has been designed by the authors in PHP programming language connected to a MySQL MariaDB database that continuously searches for new patients consulting at hospital. Medical records of various types have been collected applying web scraping that uses HTTP protocol to their hospital web interface. The collected data was further analysed for medical observations using machine learning was performed.

The paper titled "Web Scraping with HTML DOM Method for Data Collection of Scientific Articles from Google Scholar" attempts to summarize the scientific articles in Google Scholar. To extract the articles from Google Scholar, HTML DOM parser using PHP programming language is used. Data related to the paper namely the title, authors, links, citation and the year it was published are extracted. The scraped data is then stored in MySQL server database for further analysis[13].

4.2 Web scraping using BeautifulSoup :

Beautiful Soup is a Python package for parsing HTML and XML documents. It creates a parse tree for parsed pages that can be used to extract data from HTML, which is useful for web scraping. Research carried out by Lunn et al.[4] comprises of data extraction from Indeed.com, a job searching website with keywords and locations specified using two libraries namely Beautifulsoup4 for extraction of data from HTML and XML files and lxml to process XML and HTML information in Python. Kasereka[8] in his paper suggested the use of BeautifulSoup to retrieve particular content from a web page, remove HTML tags, and save the information. Clement et al.[12] in their paper use BeautifulSoup to extract digital notices from government portals regarding smart city strategy.

4.3 Web crawling using Java libraries :

Crawler4j is an open-source Java crawler designed for crawling webpages with a simple interface. One can set up a multi-threaded crawler using this library in a few minutes. Dascalu et al.[5], implemented two crawlers to extract relevant drug information from Biofarm and HelpNet websites. The extracted content was then parsed using the jsoup library to extract needed information and stored in Elasticsearch. On the other hand, Kolli et al.[7], used Crawler4j library to extract data from online news websites and APIs provided by JTidy were used to clean the extracted data for further analysis. The authors also used DOM hierarchy for parsing the contents and filtered to provide the user required content.

Authors Hassanien et al. [6] used a web scraper tool named WebScraper, an extension in Google Chrome to extract information from Google scholar.

Ahmed et al. [15] in their paper propose a framework by modifying the behaviour of focused crawler using a domain distiller using Optimized Naïve Bayes (ONB) Classifier. By using a domain distiller, the performance of focused crawlers is improved.

In [9], depth first technique is combined with the technique of web scraping. It implements a keywords-based data searching approach. The user is allowed to give an input based on which the scraper uses depth first search technique to fetch the required data comprising dates, headlines, links to pictures, news, links, categories based on which group is done. This study begins with the process of loading the URL in online news intended for the keyword "education". After that, the depth first search starts by taking the start date, and the expiration date of the news, the URL news, and a category, and will be repeated until news that matches the search is found. Search result URL continues to scratch and crawl data in accordance with keywords. After scraping and crawling process data, news data is exported to an Excel file format (.csv), and stored in a NoSQL database.

4.4 Web scraping using Selenium :

Selenium is an open-source web-based automation tool which is quite efficient for web scraping. The web driver in Selenium provides numerous features that enable to navigate through the desired web pages and fetch various contents of the page depending on necessities. Thus, lots of data from various web pages concerning the user's query can be extracted from multiple web pages and grouped[24].

In [24], extractive text summarization of web pages is performed with the help of Selenium framework and TF-IDF algorithm. The data is extracted from the webpages and the extracted content is then summarized using the TF-IDF algorithm. The extraction framework proposed comprises of the following steps :

- The user enters a query
- The user query is concatenated with the pre-defined URL and user query related URL is generated
- The data is then retrieved from the URLs and saved into a text file.

In [19], the authors Fang et al., propose to provide a web-based platform giving information about the pesticides including scientific information by integrating data from several public databases. To extract data, the authors used several techniques to crawl the web to extract information about pesticides and by evaluating the performance used a combined approach of Selenium based

crawler and footprint preservation method to crawl the websites and provide the filtered information.

4.5 Web scraping using Apache Nutch :

Apache Nutch is an open-source large scale distributed web-crawler and is developed in Java language that can be extended very easily.

In the study of Shafiq et al.[21], an attempt to build NCL Crawl, a system using Apache Nutch Crawler and Compact Language Detector (CLD2) for language specific web crawling is made.

Barman et al.[22], aim to develop a Monolingual Information Retrieval (IR) system for the Assamese language. A list of Govt. and General Assamese URLs were identified for crawling purposes. Apache Lucene along with Apache Nutch has been used by authors to index the web documents crawled by Apache Nutch.

4.6 Web Scraping using Scrapy :

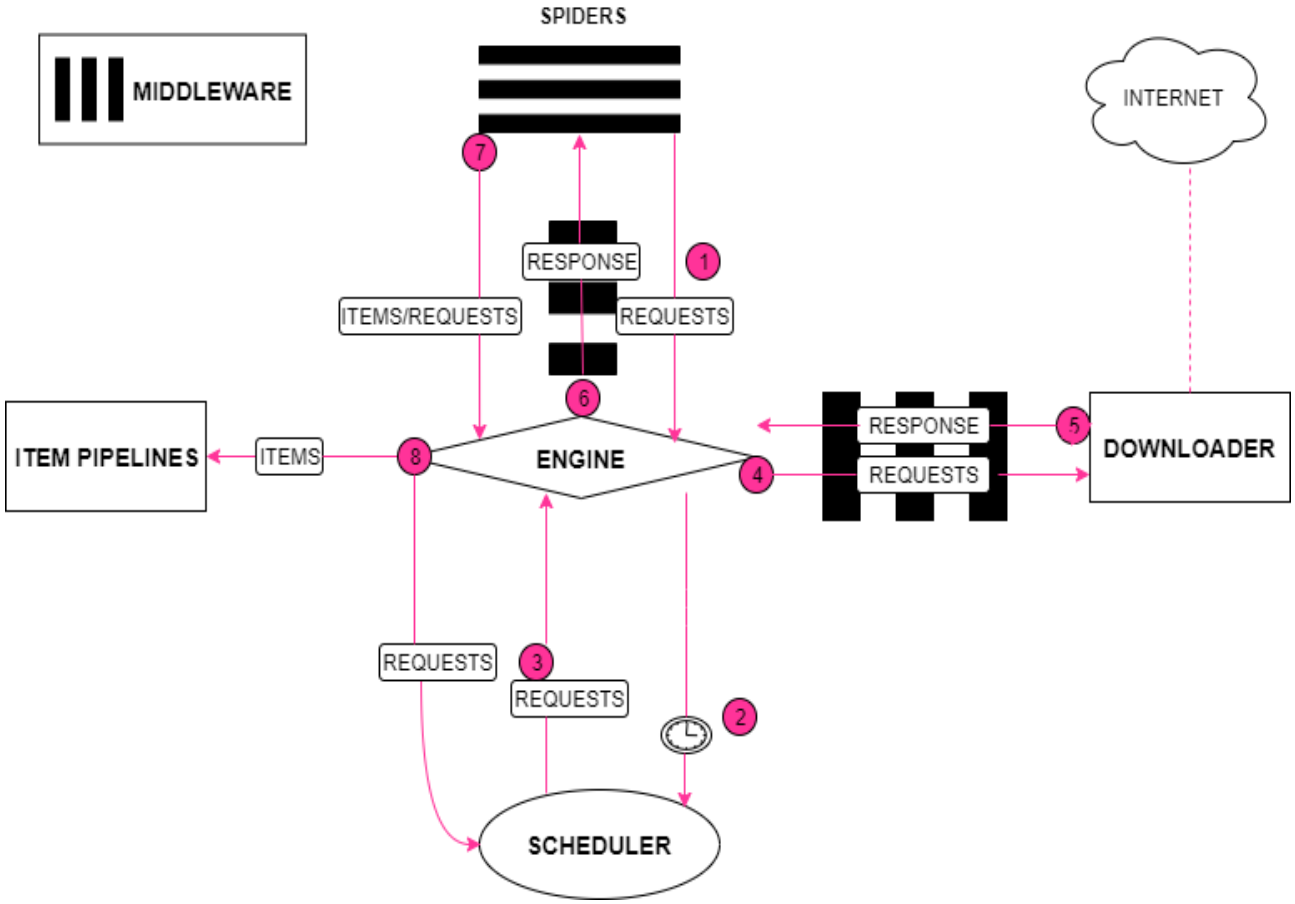

FIGURE 2 – Scrapy framework

The components of the Scrapy framework is shown in figure 2.

Engine is the centre of the Scrapy framework. It controls the flow of data between Scrapy's components. It's also in charge of listening for and generating events in reaction to events like request errors, response errors, and exceptions.

The Scheduler controls when a task should be completed and provides a direct link to task queues. It has the ability to control the amount of time each request takes.

The Downloader is where HTTP requests are made. In the normal case, where no real browser is used, it stores and returns the HTTP response data to the engine[30]. If a genuine browser is used to make requests, Downloader will be totally replaced by a middleware that can control the browser.

Spiders are developer-created classes that specify what actions the Scraper should take to obtain and interpret certain web material[30]. Custom options for Downloader and associated middleware can also be set here. Item Pipeline receives the parsed contents.

Item Pipeline parses data returned by Spiders and performs validation, custom transformations, cleaning, and data persistence to Redis, MongoDB, or Postgres.

Downloader middleware intercept requests and responses sent to and from Downloader, and add custom metadata to the request and response data[30].

Asikri et al.[10], in their paper employ Scrapy framework to scrape information from an e-commerce website called "http ://www.jumia.ma/". CSS Selectors have been used to parse and extract the required content from the website.

4.7 Web scraping using R :

RCrawler is a package developed by Khalil et al.[14], for the R language. It is used for domain-based web crawling and content scraping. The RCrawler can crawl, parse, store pages, extract contents, and produce data that can be directly employed for web content mining applications. The main features of RCrawler are multi-threaded crawling, content extraction, and duplicate content detection.

Marchi et al.[18], utilize R language to scrape data from official city websites and official tourism promotion websites of the destinations to study the sustainability communication in official destination websites for informing and motivating visitors to adopt sustainable practices and behaviours.

Table 2 : Comparison of open source web scraping libraries and frameworks.

	Type[1]	API/stand alone	Language	Extraction facilities[2]
Jsoup	CP	API	Java	H, C
HttpClient	C	API	Java	
Scrapy	F	Both	Python	R, X, C
BeautifulSoup	P	No	Python	H
Apache Nutch	F	Both	Java	R, X, H, C
Selenium	P	API	Java, Python	R, X, C

[1] Type :

- C : HTTP Client
- P : Parsing
- F : Framework

[2] Extraction facilities :

- R : Regular expressions
- H : HTML parsed tree
- X : XPath
- C : CSS selectors

Table3 : Comparison of Python web scraping libraries and frameworks.

Factors	BeautifulSoup	Scrapy	Selenium
Extensibility	Suitable for low-level complex projects	Best choice for large or complex projects	Best for projects dealing with Core JavaScript
Performance	Pretty slow compared to other libraries while performing a certain task	Rapid processing due to use of asynchronous system calls	Can handle up to some level but not as much as Scrapy
Ecosystem	It has a lot of dependencies on the ecosystem.	It has a flexible ecosystem making it easy to integrate with proxies and VPNs.	It has good ecosystem for development

5 Conclusion :

In this study, we have reviewed the recent literature relating to the applications of web scraping in various domains, web scraping techniques and tools that employ web scraping techniques. We use this study to improve our process of web scraping, and we discovered that the majority of the web scrapers are often quite generic and mostly designed to perform common, simple tasks. By comparing the performance and features of different tools and frameworks, we found that Scrapy provides better results as it is fast, extensible and powerful. Since Scrapy handles requests asynchronously, the results can be scraped at a very quick pace. Scrapy's architecture is based on a

web crawler which enables easy data extraction. Scrapy's selectors like CSS and XPath can be employed to extract the required data. For complex projects, Scrapy is the perfect tool because of its flexible and extensible capabilities making integration with VPNs and proxies easier. In addition, ScraperAPI supports browsers, proxies, and CAPTCHAs, allowing you to get raw HTML from any website with a single API call.

6 References

1. Gunawan, Rohmat & Rahmatulloh, Alam & Darmawan, Irfan & Firdaus, Firman. (2019). Comparison of Web Scraping Techniques : Regular Expression, HTML DOM and Xpath. 10.2991/icoiese-18.2019.50.
2. Suganya, E., Vijayarani, S. Firefly Optimization Algorithm Based Web Scraping for Web Citation Extraction. Wireless Pers Commun 118, 1481–1505 (2021). https://doi.org/10.1007/s11277-021-08093-z
3. Melchor, Raul & Fonseca, Marta & Rey, Beatriz & Hernández, Alberto & Puertas, Borja & Gomez, Sandra & Palomino, Danylo & Román, Luz & Peña, Andres & Mateos, Maria. (2020). CT-152: Application of Web-Scraping Techniques for Autonomous Massive Retrieval of Hematologic Patients' Information During SARS-CoV2 Pandemic. Clinical Lymphoma Myeloma and Leukemia. 20. S214. 10.1016/S2152-2650(20)30778-3.
4. Lunn, Stephanie & Zhu, Jia & Ross, Monique. (2020). Utilizing Web Scraping and Natural Language Processing to Better Inform Pedagogical Practice. 10.1109/FIE44824.2020.9274270.
5. Dascalu, Maria-Dorinela & Paraschiv, Ionut & Nicula, Bogdan & Dascalu, Mihai & Trausan-Matu, Stefan & Nuta, Alexandru. (2019). Intelligent Platform for the Analysis of Drug Leaflets Using NLP Techniques. 1-6. 10.1109/ROEDUNET.2019.8909606.
6. Hassanien, Hossam El-Din. (2019). Web Scraping Scientific Repositories for Augmented Relevant Literature Search Using CRISP-DM. Applied System Innovation. 2. 37. 10.3390/asi2040037.
7. Kolli, Srinivas & Rama, Peddarapu & Reddy, Parvathala. (2021). A Novel NLP and Machine Learning based Text Extraction Approach from online news feed.
8. Kasereka, Henrys. (2020). Importance of web scraping in e-commerce and e-marketing.. SSRN Electronic Journal. 10.6084/m9.figshare.13611395.v1.
9. Arumi, Endah Ratna, and Pristi Sukmasetya. "Exploiting Web Scraping for Education News Analysis Using Depth-First Search Algorithm." Jurnal Online Informatika 5.1 (2020): 19-26.
10. Asikri[1], M & Chaib, Hassan & Salah-ddine, Krit. (2020). Using Web Scraping In A Knowledge Environment To Build Ontologies Using Python And Scrapy. European Journal of Translational and Clinical Medicine. 7. 433-442.
11. Deng, Shiqi. (2020). Research on the Focused Crawler of Mineral Intelligence Service Based on Semantic Similarity. Journal of Physics: Conference Series. 1575. 012142. 10.1088/1742-6596/1575/1/012142.
12. Nicolas, Clément. (2020). Natural language processing-based characterization of top-down communication in smart cities for enhancing citizen alignment. Sustainable Cities and Society. 66. 10.1016/j.scs.2020.102674.

13. Rahmatulloh, Alam & Gunawan, Rohmat. (2020). Web Scraping with HTML DOM Method for Data Collection of Scientific Articles from Google Scholar. Indonesian Journal of Information Systems. 2. 16. 10.24002/ijis.v2i2.3029.

14. Khalil, Salim & fakir, Mohamed. (2017). RCrawler: An R package for parallel web crawling and scraping. SoftwareX. 6. pp. 98-106. 10.1016/j.softx.2017.04.004.

15. Saleh, Ahmed & Abulwafa, Arwa & Alrahmawy, Mohammed. (2017). A Web Page Distillation Strategy for Efficient Focused Crawling Based on Optimized Naïve Bayes (ONB) Classifier. Applied Soft Computing. 53. 10.1016/j.asoc.2016.12.028.

16. Kotouza, Maria & Tsarouchis, Sotirios - Filippos & Kyprianidis, Alexandros-Charalampos & Chrysopoulos, Antonios & Mitkas, Pericles. (2020). Towards Fashion Recommendation: An AI System for Clothing Data Retrieval and Analysis. 10.1007/978-3-030-49186-4$_3$6.

17. Tharaniya, B & Liyanapathirana, Chethana & Sampath, Kalpa & Rupasinghe, Prabath. (2018). Extracting Unstructured Data and Analysis and Prediction of Financial Event Modeling.

18. Marchi, Valentina & Apicerni, Valentina & Marasco, Alessandra. (2021). Assessing Online Sustainability Communication of Italian Cultural Destinations – A Web Content Mining Approach. 10.1007/978-3-030-65785-7$_5$.

19. Fang, Tian & Han, Tan & Zhang, Cheng & Yao, Ya. (2020). Research and Construction of the Online Pesticide Information Center and Discovery Platform Based on Web Crawler. Procedia Computer Science. 166. 9-14. 10.1016/j.procs.2020.02.004.

20. Seliverstov, Yaroslav & Seliverstov, Svyatoslav & Malygin, Igor & Korolev, Oleg. (2020). Traffic safety evaluation in Northwestern Federal District using sentiment analysis of Internet users' reviews. Transportation Research Procedia. 50. 626-635. 10.1016/j.trpro.2020.10.074.

21. Shafiq, Hafiz Muhammad, and Muhammad Amir Mehmood. "NCL-Crawl: A large scale language-specific Web crawling system." LANGUAGE & TECHNOLOGY (2020): 79.

22. Barman, Anup Kumar, Jumi Sarmah, and Shikhar Kr Sarma. "Developing Assamese Information Retrieval System Considering NLP Techniques: an attempt for a low resourced language." ADBU Journal of Engineering Technology 8.2 (2019).

23. Wang H, Song J. Fast Retrieval Method of Forestry Information Features Based on Symmetry Function in Communication Network. Symmetry. 2019; 11(3):416. https://doi.org/10.3390/sym11030416

24. K. U. Manjari, S. Rousha, D. Sumanth and J. Sirisha Devi, "Extractive Text Summarization from Web pages using Selenium and TF-IDF algorithm," 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), 2020, pp. 648-652, doi: 10.1109/ICOEI48184.2020.9142938.

25. Chaitra, P. & Deepthi, V. & Vidyashree, K. & Rajini, S.. (2020). A Study on Different Types of Web Crawlers. 10.1007/978-981-13-8618-$3_8$0.

26. S. C. M. de S Sirisuriya, "A Comparative Study on Web Scraping," Proc. 8th Int. Res. Conf. KDU, no. November, pp.135–140, 2015.

27. M. El Asikri , S. Krit , H.Chaib , M. Kabrane, H. Ouadani, K. Karimi,K.Bendaouad and H. Elbousty Polydisciplinary Faculty of Ouarzazate, Ibn Zohr University, Agadir : Mining the Web for learning ontologies : state of art and critical review, (ISCSA2017 submission 45) à Errachidiya

28. Song, Ruihua; Microsoft Research (Sep 14, 2007). "Joint Optimization of Wrapper Generation and Template Detection" (PDF). The 13th International Conference on Knowledge Discovery and Data Mining.

29. W3C, "What is the Document Object Model?," 2016.

30. Scrapy documentation.

31. "Phan, H.. "Building Application Powered by Web Scraping." (2019).

32. Spangher, Alexander & May, Jonathan. (2021). A Web Application for Consuming and Annotating Legal Discourse Learning.

33. Himawan, Arif & Priadana, Adri & Murdiyanto, Aris. (2020). Implementation of Web Scraping to Build a Web-Based Instagram Account Data Downloader Application. IJID (International Journal on Informatics for Development). 9. 59-65. 10.14421/ijid.2020.09201.

34. Kinne, Jan & Lenz, David. (2021). Predicting innovative firms using web mining and deep learning. PLOS ONE. 16. e0249071. 10.1371/journal.pone.0249071.

35. Web Scraping, Regular Expressions, and Data Visualization: Doing it all in Python

36. Li R.Y.M. (2021) Building Updated Research Agenda by Investigating Papers Indexed on Google Scholar: A Natural Language Processing Approach. In: Ahram T. (eds) Advances in Artificial Intelligence, Software and Systems Engineering. AHFE 2020. Advances in Intelligent Systems and Computing, vol 1213. Springer, Cham. https://doi.org/10.1007/978-3-030-51328-$3_4$2

37. Breno Santana Santos, Ivanovitch Silva, Marcel da Câmara Ribeiro-Dantas, Gisliany Alves, Patricia Takako Endo, Luciana Lima, COVID-19: A scholarly production dataset report for research analysis,Data in Brief, Volume 32, 2020, 106178, ISSN 2352-3409.

38. Boegershausen, Johannes, Abhishek Borah, and Andrew T. Stephen. "Fields of Gold: Web Scraping for Consumer Research."

39. Saranya, G., et al. "Prediction of Customer Purchase Intention Using Linear Support Vector Machine in Digital Marketing." Journal of Physics: Conference Series. Vol. 1712. No. 1. IOP Publishing, 2020.

40. Nguyen, Viet-Hoang, Suku Sinnappan, and Minh Huynh. "Analyzing Australian SME Instagram Engagement via Web Scraping." Pacific Asia Journal of the Association for Information Systems 13.2 (2021): 2.