Web Scraping Techniques and Applications: A Literature Review

Chaimaa Lotfi, Swetha Srinivasan, Myriam Ertz, Imen Latrous

LaboNFC, University of Quebec at Chicoutimi, 555 Boulevard de l'Université, Saguenay (QC), Canada

Corresponding author: Myriam Ertz, Email: myriam_Ertz@uqac.ca

Big data analytics gives organizations a way to analyze huge data sets and gather new information. It helps answer basic questions about business operations and business performance. It also helps discover unknown patterns in vast datasets or combinations thereof. In the current data-driven world, it becomes increasingly essential that big data techniques are applied and analyzed for organizational growth. More specifically, with the large availability of data on the Web, whether from social media, websites, online portals, or platforms, to name but a few, it is important for organizations to know how to mine that data in order to extract useful knowledge. Web scraping represents a fundamental approach in this regard. Therefore, this paper aims to provide an updated literature review about the most advanced Web Scraping techniques to better equip scholars and managers with helpful knowledge on how to mine most effectively online data. The paper starts with presenting the basic design of a web scraper and the applications of web scraping in diverse sectors and areas. Next, the different Web scraping methods and Web scraping technologies are presented. Finally, a procedure to develop Web scraping with various tools is proposed before a conclusion wraps up the paper.

Keywords: Big data, web scraping, business performance, web crawling, web mining

2021. In Raju Pal & Praveen K. Shukla (eds.), *SCRS Conference Proceedings on Intelligent Systems*, 381–394. Computing & Intelligent Systems, SCRS, India.
https://doi.org/10.52458/978-93-91842-08-6-38

1 Introduction

Data has a vital role in business, marketing, engineering, social sciences, and other disciplines of study since it may be utilized as a starting point for any operations activities that include the exploitation of information and knowledge. The initial step of research is data collecting, 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 subject or topic of study, the type of data sought, and the user's aims. Depending on the goals and conditions, the method's application methodology can also change without jeopardizing data integrity, correctness, or reliability [1]. There are numerous data sources on the Internet that might be employed in the design process. The technique of extracting data from websites is often known as web scraping, web extraction, web harvesting, web crawler.

This research will address how to build a web scraping tool to extract meaningful information from online sources and look for recent web scraping methods and techniques. The study further helped us compare the available tools and choose the most suitable one for the study.

The basic design of a web scraper is shown in Table 1. This table depicts the general schema that we will follow in our resolution study.

The table below shows the stages of the process of literature review that was followed for this study.

Table 1. Process of the Study

No.	Stage	Description
1	Determination of research questions	How to collect data from the Internet? What is Web data extraction, and how to perform it? What are the techniques to perform web scraping, and what are its applications?
2	Establishing criteria for inclusion and exclusion	The research includes documents that were published as of June 2021. Peer-reviewed publications, conference proceedings, book chapters, professional articles, industry data, and research reports were all included in the search. Furthermore, only papers written in English were considered.
3	Developing a research nomenclature	ABI/Inform, Academic Search Complete, Google Scholar, JSTOR, Scopus, PubMed, Web of Science, and IEEE Xplore were used to conduct a series of systematic retrievals in critical academic databases. The following search phrases were chosen: "web scraping," "web data extraction," "web mining," and "web crawling" are all terms used to describe the process of extracting information from the Internet. The publications were thoroughly examined, with all parts of the text scrutinized.
4	Independent Analysis and mapping of the results	All the selected publications were analyzed individually and mapped based on domains, tools, and technologies used and presented as a summary for each publication.

5	Structure Analysis	Based on the findings of Stage 4, we summarized the papers and publications under each head, namely, "Applications of Web scraping" explaining the different domains where web scraping was used, "Web Scraping Methods," describing the different ways of extracting data from the Internet, "Web Scraping Technology" elucidating the different available technologies and "Development of Web scraping tools" details how the web scraping tools used in the publications are built.

2 Applications of Web Scraping

Web scraping is widely utilized for a variety of purposes, including comparing prices online, observing changes in weather data, 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 [2].

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

2.1 In Healthcare

Healthcare is no longer a domain that relies wholly on physical contact. Instead, in its unique manner, it has gone digital. In this data-driven environment, web scraping in healthcare can save many lives by allowing sensible decisions to be made.

Healthcare workers typically regard data collecting engaging many patients as a tedious and arduous process. Even while clinical data is needed more than ever, the current patient load makes gathering it nearly impossible. To that end, the author proposes implementing a system that collects clinical data from SARS-CoV2 patients who visit the hospital automatically and autonomously for future research [3].

Another application of web data extraction techniques in the healthcare domain is research conducted by Dascalu et al. [4], where crawlers extract drug leaflets.

2.2 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, companies can quickly analyze the customers' sentiment towards their products, improve public relations and audience engagement.

For this purpose, they created a web-based Instagram account data download application that may be utilized by numerous parties for this purpose, using a web scraping technology. The web scraping method was chosen by the researchers eliminating the need to use Instagram's Application Programming Interface (API), which has several restrictions for accessing and retrieving data on the platform. The web scraping method successfully created an Instagram account data grabber application. Application testing was carried out on 15 accounts with a total number of publications ranging from 100 to 11000 in this study. The web scraping solution was able to successfully capture Instagram account data for 2412 accounts, based on the results of the analysis. This application can

help users save Instagram account data to a database manager and export data to several formats, namely Excel, JSON, or CSV [3].

2.3 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 a minimal cost. Then, utilizing traditional company-level data from a questionnaire-based innovation survey, trained an ANN classification model using labeled (product innovator/no product innovator) online texts of surveyed enterprises (German Community Innovation Survey). They then 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 against patent statistics at the firm level, benchmark data derived from survey analysis, and regional innovation indicators. Given its breadth and geographic granularity, the findings show that this method yields solid projections and has the potential to be a valuable and cost-effective addition to the existing set of innovation indicators [5].

The research conducted by Tharanya et al. [6] uses technical analysis of news articles scraped from the Internet. The news is extracted from a reliable website, and the contents of the website are summarized to perform analysis and event modeling.

2.4 In Marketing

Boegershausen et al. [7], in the report, talk about the vast amount of customer data in the form of a digital footprint available to analyze customer behavior and to answer customer research questions. In their paper, Saranya et al. [8] propose to predict customer purchase intention during online purchases using machine learning models. The data is collected using web scraping since the information on the Web is in an unstructured format. The data is further analyzed to predict the purchase intent.

Nguyen et al. [9] 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 garner more engagement as it is more trustworthy.

2.5 Others

Deng, in his paper, has used web data extraction techniques to extract information on mineral intelligence in China [10]. Kotouza et al. [11]. have taken advantage of web data extraction techniques to design a system that acts as an assistant to a fashion designer to provide information about the newest fashion trends improving customization. In [12], the authors have used the information available on the Internet to extract forestry information features. Based on the reviews published on the Web, the authors Yaroslav et al. [13] performed the task of studying traffic safety in Northwestern Federal District using Python libraries Scrapy to scrape the reviews from the Internet.

2.6 In research

Authors Suganya et al. [14], 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: Particle Swarm Optimization, Hidden Markov Model algorithm, and Firefly Optimization algorithm-based Web scraping to extract information regarding web citation based on the given 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. [15] 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 a PDF or Excel file.

The authors Kolli et al. [16] show a customized news Internet search engine that focuses on constructing a repository of reporting stories by relating adept content data mining from a network information sheet from shifted e-information entrances.

In Li [17], the author proposes employing web scraping and natural language processing to decrease the time required to detect the research gap. This strategy is tested by looking at three different areas: safety awareness, home prices, sentiment, and artificial intelligence. First, the titles of the publications are scraped from Google Scholar and using tokenization. The titles are parsed. By ranking the collocations based on descending range of frequency, the set of keywords that are not used in the paper title is obtained, and the research void is determined.

In Breno et al. [18], the paper proposes a scholarly production dataset focusing on COVID-19 to provide an overview of scientific research activities, making it easier to identify countries, scientists, and research groups most active in this corona virus disease task force. Between January 2019 and July 2020, a dataset containing 40,212 records of article metadata was extracted from various databases, namely Scopus, PubMed, arXiv, and bioRxiv using Python Web Scraping techniques and pre-processed with Pandas Data Wrangling using a pipeline versioned with the Data Version Control tool (DVC), making it easy to replicate and audit. To extract data from PubMed and Scopus, API was used, and Scrapy was used for scraping data from arXiv and bioRxiv databases.

3 Web Scraping Methods

Web scraping is the process of autonomous data mining or gathering information from the Internet and other common databases. Different Web scraping methods have been developed in multiple types of research and are presented in the following sub-sections.

3.1 Traditional Copy and Paste

The copy-pasting method is simple: access the page using your browser, then manually copy and paste it onto other media. However, even though the method is pretty easy and straightforward if the website employs a barrier program, it makes it difficult to use [1], which requires a human selection of objects or sentences that are somewhat long. At the same time, other methods are more challenging to utilize and necessitate an extra program.

3.2 HTML Parsing

Extensive collections of pages are produced programmatically from a fundamental organized source, such as a database, on many websites. A common script or template encodes data from the same category into similar pages. A wrapper is a program in data mining that recognizes templates in a given source of information, extracts its content, and converts it into a relational form [19]. Wrapper generation techniques presume that a wrapper induction system's input pages follow a common pattern and can be determined pretty easily by using a common URL format. Furthermore, semi-structured data query languages like XQuery and HTQL can be used to analyze HTML websites and extract and change their information [20,21].

3.3 DOM parsing

Programs can obtain dynamic material generated by client-side scripts by placing a developed web browser, such as Internet Explorer or the Mozilla browser control. These applications also parse web pages into a Document Object Model (DOM) tree, from which applications can extract sections of the pages [19,21]. Also, a tree structure Document Object Model can represent a web page. For example, 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. For example, if it's on the page, it can be tracked without waiting for the web development team to expose it through the data layer [21].

3.4 HTML DOM

The HTML DOM (Hyper Text Markup Language Document Object Model) is a yardstick for obtaining, altering, and editing HTML elements [22]. By defining objects and properties for all HTML components, as well as ways to access them, DOM efficiency can be improved. For example, JavaScript can access all elements in an HTML document using the DOM. To access objects, the HTML DOM employs computer languages, most often JavaScript [22].

Every HTML element is considered an object. Each object's method and property make up the programming interface [1, 22].

3.5 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 meta characters 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 [23].

3.6 XPath

The main component of the XSLT standard is XPath (Stylesheet Language Transformation). In eXtensible Markup Language (XML) documents, XPath can explore elements and attributes [20]. XPath is a node selection language for XML documents that may also be used with HTML. The most useful XPath expression is the location path. A path location employs at least one step location to determine a group of nodes in a document. The simplest is a location path that selects the document root node. This road has a slash "/" in the centre of it. The symbol is both the root of a Unix system file and a document.

3.7 Vertical aggregation platform

Various companies have created vertical-specific harvesting platforms. With no manual intervention and 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 whole 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 and its scalability. This scalability is mainly utilized to choose the Long Tail of sites that are too difficult or time-consuming for traditional aggregators to extract content from.

3.8 Semantic annotation recognizing

Metadata, semantic markups, and annotations may be included on the scraped pages, which can discover particular data pieces. For example, this technique can be considered a specific case of DOM parsing if the annotations are incorporated in the pages as Micro format 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.

3.9 Computer Vision Web Page Analyzer

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

3.10 Comparison between web scraping methods

The comparison is conducted by putting each method to the test when extracting data from the required website, then computation and comparing the results. Process time, memory utilization, and data consumption are the experiment's measurement parameters. The findings of the experiment show that web scraping with the Regex method uses the least amount of RAM when compared to the HTML DOM approach and XPath. In addition, HTML DOM takes the least amount of time and consumes the least amount of data when compared to Regular Expression and XPath approaches [1].

4 Web Scraping Technology

4.1 Web Crawlers

A web crawler is a bot that visits websites and extracts data from them. According to Mahto and Singh [24], a web crawler 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. First, 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 [19]. 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 web crawler: This type of crawler searches for web pages related to certain user fields or subjects. It makes an effort to find more relevant pages with a greater level of precision. It only downloads pages relevant to the topic and ignores ones that are not relevant, which is enabled by ranking the Web pages.
2. Incremental crawler: Incremental crawlers are web crawlers that visit and access updated web pages. These crawlers visit the web pages frequently and update website material by saving the most recent version of pages.
3. Distributed crawler: These crawlers function by assigning crawling to other crawlers. A central server manages the communication and synchronization of 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 that are not accessible to general users is known as hidden Web. The crawler which collects this data is known as a hidden crawler [25].

In the table below, we carry out a comparison between web crawler types to choose the convenient one to work with.

Table 2. Comparison between Web crawler types

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

Notes: [1] DES: Depth First Search, [2] BFS: Breadth First Search.

4.2 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 extracting 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.

4.3 Web Scraping Policies

Selection, re-visit, politeness, and parallelization, according to Asikri et al. [19] and Mahto and Singh [24], are the four fundamental policies that a crawler must follow in order to act efficiently. The crawler can eliminate most useless links and considerably reduce its search space by focusing on vital links first. When pages are dynamic, the crawler must check for updates regularly.

5 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 Suganya et al. [14], the authors used web scraping and crawling to obtain information from 12,250 web pages in a Google Scholar web citation database. First, the information about the authors and the manuscript is extracted and saved in a .csv file. The Seed URL for the web crawler is the user's query to Google Scholar. After that, the web crawler crawls the HTML pages and downloads the content that the user requires. The URL is parsed for citation information and links, and then placed in the database. The web scraper uses a selector gadget to choose the citation material and collects citation information from a particular URL. The information is taken from the web document once it has been parsed. The text is then filtered by keywords or matching a certain pattern before being saved in a structured fashion as a.csv file. Web scraping and the firefly optimization technique are combined in the suggested algorithm. The web scraping technique scrapes/extracts citation information from the Web, but the

firefly algorithm assigns random values, refreshes the light intensity, and evaluates the relevance of the paper's title and the user query at each step. As a result, the suggested approach retrieves information with more precision.

5.1 Web Scraping using PHP

In Melchor et al. [26, 27], the authors designed a daemon in PHP programming language connected to a MySQL MariaDB database that continuously searches for new patients consulting at the hospital. Medical records of various types have been collected applying web scraping that uses HTTP protocol to their hospital web interface. The collected data were further analyzed for medical observations using machine learning.

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. An HTML DOM parser using the PHP programming language is used to extract the articles from Google Scholar. 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 the MySQL server database for further analysis [15].

5.2 Web Scraping using Beautiful Soup

Beautiful Soup is a Python library that allows you to parse HTML and XML files. A parse tree is created for the pages parsed, using which data from HTML pages are extracted. Research carried out by Lunn et al. [28] comprises data extraction from Indeed.com, a job searching website with keywords and locations specified using two libraries, namely Beautifulsoup4, to extract data from HTML and XML files and lxml to process XML and HTML information in Python. Kasereka [29], in his paper, suggested the use of Beautiful Soup to retrieve particular content from a web page, remove HTML tags, and save the information. In their paper, Clement et al. [30] use Beautiful Soup to extract digital notices from government portals regarding smart city strategy.

5.3 Web Scraping 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. For example, Dascalu et al. [31] 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 Elastic search. On the other hand, Kolli et al. [16] used the 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 the DOM hierarchy for parsing the contents and filtered to provide the user required content.

Authors Hassanien et al. [32] used a web scraper tool named WebScraper, an extension in Google Chrome, to extract information from Google scholar.

In their paper, Ahmed et al. [5] propose a framework by modifying the behavior 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 Arumi and Sukmasetya [33], the depth-first technique is combined with the technique of web scraping. It implements a keywords-based data searching approach. The user can 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 following keywords. After scraping and crawling process data, news data is exported to an Excel file format (.csv) and stored in a NoSQL database.

5.4 Web Scraping using Selenium

Selenium is an open-source web-based automation tool that is quite good at scraping websites. Selenium's web driver has several features that allow users to move across web pages and retrieve different page parts depending on their needs. As a result, many data from several web pages related to the user's query can be retrieved and organized [34].

In Manjari et al. [34], extractive text summarization of Web pages is performed with the help of the Selenium framework and TF-IDF algorithm. The data is extracted from the Web pages, 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 a user query related URL is generated;
- The data is then retrieved from the URLs and saved into a text file.

In Fang et al. [35], the authors propose to provide a web-based platform giving information about 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.

5.5 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. [36], the authors attempt to build a Web crawling tool, NCL Crawl, for specific languages. NCL Crawl, using Apache Nutch Crawler and Compact Language Detector (CLD2).

Barman et al. [37] aimed to develop a Monolingual Information Retrieval (IR) system for the Assamese language. A list of Government and General Assamese URLs was compiled for crawling purposes. Authors have utilized Apache Lucene and Apache Nutch to index the web content crawled by Apache Nutch.

5.6 Web Scraping using Scrapy

Asikri et al. [19], 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. The components of the Scrapy framework are shown in Figure 1.

The 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 re- quest errors, response errors, and exceptions.

The Scheduler controls when a task should be completed and directly links to task queues. In addition, it can control the amount of time each request takes.

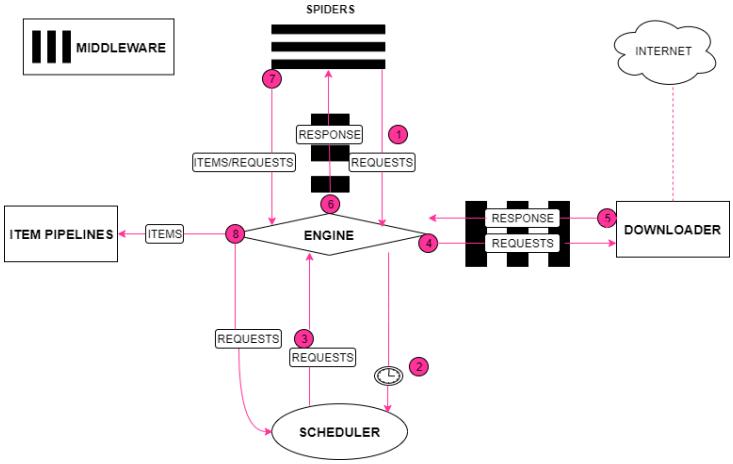

Fig. 1. The Scrapy framework; Source: Huy Phan [4]

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 [38]. However, if a genuine browser is used to make requests, the 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 [38]. Custom options for Downloader and associated middleware can also be set here. Finally, 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 intercepts requests and responses sent to and from Downloader and add custom metadata to the request and response data [38].

5.7 Web Scraping using R

RCrawler is a package developed by Khalil et al. [39] for the R language. It's used for content scraping and domain-based Web crawling. The RCrawler can crawl, parse, store, and extract material from online sites, as well as generate data that may be used directly in web content mining applications. Multi-threaded crawling, content extraction, and duplicate content detection are the core characteristics of RCrawler.

Marchi et al. [40] utilize R language to scrape data from official websites of the city and official tourism promotion websites of the destinations to study the sustainability communication in websites for informing and motivating visitors to adopt sustainable practices and behaviors.

5.8 Comparison of web scraping tools

Table 3 and Table 4 summarize and compare the different tools that can be used for web scraping purposes. Table 3 focuses more specifically on the Python Web scraping libraries and frameworks.

Table 3. Comparison between Open-Source Web scraping techniques and frameworks

Parameters	Type[1]	API/standalone	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

Notes: [1]Type: C = HTTP Client
[2] Extraction facilities:
R = Regular expressions
P = Parsing
H = HTML parsed tree
F = Framework
X = XPath
C = CSS selectors

Table 4. Comparison between 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 the 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 a good ecosystem for the development

6 Conclusion

Building on previous topical work [41,42], this study reviews 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 web scraping process, and we discovered that most of the web scrapers are often quite similar and general in nature designed to carry out generic and easy jobs. 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 rapidly. Furthermore, 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. Scrapy is the perfect tool for complex projects because of its flexible and extensible capabilities, making integration with VPNs and proxies easier. In addition, Scraper API supported browsers, proxies, and CAPTCHAs, allowing you to get raw HTML from any website with a single API call.

References

[1] Gunawan, R. et al. (2019). Comparison of web scraping techniques: regular expression, HTML DOM and Xpath. In *International Conference on Industrial Enterprise and System Engineering*, 2:283-287.

[2] Sirisuriya, D. S. (2015). A comparative study on web scraping. In the *Proc. 8th Int. Res. Conf. KDU*, 135–140.

[3] Spangher, A. and May, J. (2021). A Web Application for Consuming and Annotating Legal Discourse Learning. *arXiv preprint arXiv:2104.10263*.

[4] Phan, H. (2019). Building Application Powered by Web Scraping. *Doctoral Thesis*.

[5] Saleh, A. I. et al. (2017). A web page distillation strategy for efficient focused crawling based on optimized Naïve bayes (ONB) classifier. *Applied Soft Computing*, 53:181-204.

[6] Tharaniya, B. et al. (2018). Extracting Unstructured Data and Analysis and Prediction of Financial Event Modeling. In *Conference proceedings of the Annual Conference IET,* 6-11.

[7] Boegershausen, J. et al. (2021). Fields of Gold: Web Scraping for Consumer Research. *Marketing Science Institute Working Paper Series*, 21-101:1-58.

[8] Saranya, G. et al. (2020). Prediction of Customer Purchase Intention Using Linear Support Vector Machine in Digital Marketing. In *Journal of Physics: Conference Series, IOP Publishing*, 1712(1):012024.

[9] Nguyen, V. H., Sinnappan, S. and Huynh, M. (2021). Analyzing Australian SME Instagram Engagement via Web Scraping. *Pacific Asia Journal of the Association for Information Systems*, 13(2):11-43.

[10] Deng, S. (2020). Research on the Focused Crawler of Mineral Intelligence Service Based on Semantic Similarity. In *Journal of Physics: Conference Series, IOP Publishing*, 1575(1):012042.

[11] Kotouza, M. T. et al. (2020). Towards fashion recommendation: an AI system for clothing data retrieval and analysis. In *IFIP International Conference on Artificial Intelligence Applications and Innovations. Springer, Cham,* 433-444.

[12] Wang, H. and Song, J. (2019). Fast Retrieval Method of Forestry Information Features Based on Symmetry Function in Communication Network. *Symmetry*, 11(3):416.

[13] Seliverstov, Y. et al. (2020). Traffic safety evaluation in Northwestern Federal District using sentiment analysis of Internet users' reviews. *Transportation Research Procedia*, 50:626-635.

[14] Suganya, E. and Vijayarani, S. (2021). Firefly Optimization Algorithm Based Web Scraping for Web Citation Extraction. *Wireless Personal Communications*, 118(2):1481-1505.

[15] Rahmatulloh, A. and Gunawan, R. (2020). Web Scraping with HTML DOM Method for Data Collection of Scientific Articles from Google Scholar. *Indonesian Journal of Information Systems*, 2(2):95-104.

[16] Kolli, S., Krishna, P. R. and Reddy, P. B. (2006). A novel NLP and Machine Learning based text extraction approach from online news feed. *ARPN Journal of Engineering and Applied Sciences*, 16(6):679-685.

[17] Li, R. Y. M. (2020). Building updated research agenda by investigating papers indexed on Google scholar: A natural language processing approach. In *International Conference on Applied Human Factors and Ergonomics. Springer, Cham*, 298-305.

[18] Santos, B. S. (2020). COVID-19: A scholarly production dataset report for research analysis. *Data in Brief*, 32:106178.

[19] Asikri, M. E., Krit, S. and Chaib, H. (2020). Using Web Scraping In A Knowledge Environment To Build Ontologies Using Python and Scrapy. *European Journal of Molecular and Clinical Medicine*, 7(3):433-442.

[20] El Asikri, M. et al. (2017). Mining the Web for learning ontologies: State of art and critical review. In *International Conference on Engineering & MIS*), *IEEE,* 1-7.

[21] Zheng, S. et al. (2007). Joint optimization of wrapper generation and template detection. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 894-902.

[22] W3C (2016). What is the Document Object Model?. https://www.w3.org/TR/WD-DOM/introduction.html

[23] Mitchell, R. (2018). Web scraping with Python: Collecting more data from the modern web. *O'Reilly Media, Inc.*

[24] Mahto, D. K. and Singh, L. (2016). A dive into Web Scraper world. In *3rd International Conference on Computing for Sustainable Global Development*, *IEEE,* 689-693.

[25] Chaitra, P. G. et al.(2020). A study on different types of web crawlers. In *Intelligent communication, control and devices. Springer, Singapore*:781-789.

[26] Melchor, R. A. et al. (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.
[27] Lunn, S., Zhu, J. and Ross, M. (2020). Utilizing web scraping and natural language processing to better inform pedagogical practice. In *IEEE Frontiers in Education Conference*, 1-9.
[28] Henrys, K. Importance of web scraping in e-commerce and e-marketing, 1-10.
[29] Nicolas, C., Kim, J. and Chi, S. (2021). Natural language processing-based characterization of top-down communication in smart cities for enhancing citizen alignment. *Sustainable Cities and Society*, 66:102674.
[30] Dascalu, M. D. et al. (2019). Intelligent Platform for the Analysis of Drug Leaflets Using NLP Techniques. In *18th RoEduNet Conference: Networking in Education and Research*, *IEEE*, 1-6.
[31] Hassanien, H. E. D. (2019). Web Scraping Scientific Repositories for Augmented Relevant Literature Search Using CRISP-DM. *Applied System Innovation*, 2(4):37.
[32] Arumi, E. R. and Sukmasetya, P. (2020). Exploiting Web Scraping for Education News Analysis Using Depth-First Search Algorithm. *Jurnal Online Informatika*, 5(1):19-26.
[33] Manjari, K. U. et al. (2020). Extractive Text Summarization from Web pages using Selenium and TF-IDF algorithm. In *4th International Conference on Trends in Electronics and Informatics,* 48184:648-652.
[34] Fang, T. et al. (2020). Research and construction of the online pesticide information center and discovery platform based on web crawler. *Procedia Computer Science*, 166:9-14.
[35] Shafiq, H. M. and Mehmood, M. A. (2020). NCL-Crawl: A large scale language-specific Web crawling system. *Languaue & Technology,* 79.
[36] Barman, A. K., Sarmah, J. and Sarma, S. K. (2019). Developing Assamese Information Retrieval System Considering NLP Techniques: an attempt for a low resourced language. *ADBU Journal of Engineering Technology*, 8(2):1-12.
[37] Scrapy (2021). Scrapy 2.5 documentation. https://docs.scrapy.org/en/latest/
[38] Khalil, S., and Fakir, M. (2017). RCrawler: An R package for parallel web crawling and scraping. *SoftwareX*, 6:98-106.
[39] Marchi, V., Apicerni, V. and Marasco, A. (2021). Assessing Online Sustainability Communication of Italian Cultural Destinations–A Web Content Mining Approach. In *Information and Communication Technologies in Tourism, Springer, Cham*, 58-69.
[40] Ertz, M., Sun, S. and Latrous, I. (2021). The Impact of Big Data on Firm Performance. In *International Conference on Advances in Digital Science. Springer, Cham*, 451-462.
[41] Ertz, M. (2022). Handbook of research on the platform economy and the evolution of e-commerce. *Hershey, PA: IGI Global.*