IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 13, Issue 5, May 2024

DOI: 10.17148/IJARCCE.2024.13596

The Benchmark Analysis of Different Web Scraping Tools and Techniques

Dr.S.Sarumathi[1], Ms.M.Sharmila[2], Ms C.Saraswathy[3], Ms R.Loga priya[4]

[1]Professor, Department of Artificial Intelligence and Data Science, K.S.Rangasamy College of Technology, Tamil Nadu, India

[2]IT Lecturer, Computing Department, FPT Greenwich University, Ho Chi Minh City, Vietnam.

[3] Associate Professor, Department of Electronics and Communication Engineering, K.S.Rangasamy College of Technology, Tamil Nadu, India

[4] Assistant Professor, Department of Computer Science and Business Systems, K.S.Rangasamy College of Technology, Tamil Nadu, India

Abstract: The World Wide Web is a vast repository of knowledge from many diverse sources. Data, which is the most significant entity on its own, is particularly vital in the worlds of data science, computer vision, artificial intelligence, machine learning, and deep learning. Since data has already had a significant impact on so many organizations worldwide, it will always occupy center stage in the technology world. Web scraping was used to obtain data in the best possible form. The globe is chasing after the data supplied on the internet since it is so useful. Web scraping is a practice that has existed for some time and is still useful today. There are numerous forms and access methods for information on the internet. Web-based indexing or semantic processing of the material may therefore be laborious. The method that seeks to solve this problem is called web scraping. Using web scraping, unstructured web data can be converted into structured data that can be kept in a central database or spreadsheet for analysis. A few common web scraping techniques include traditional copy-and-paste, text wrapping and regular expression matching, HTTP programming, HTML parsing, DOM parsing, web-scraping software, vertical aggregation platforms, semantic annotation identifying, and computer vision webpage analysers. This comparative analysis of Web scraping tools and the techniques involved in it intends to increase the readers' awareness of this technology and aid in their quest for knowledge.

Keywords: Data science, computer vision, artificial intelligence, machine learning, deep learning, web scraping

I. INTRODUCTION

Websites store data and are explicitly accessible on the internet. As a result, anyone who has access to the site can search for the information. As per Sirisuriya [1,] this data can be gathered in a variety of ways, ranging from manually searching and copying the data to automated methods that analyse the page, identify, and retrieve the desired data. Web scraping technology can be used to collect data efficiently. Web scraping, also known as web harvesting or web extraction, is a technique for extracting data from the internet and storing it in a file or database [2]. technology. Data from webpages, whether structured or unstructured, can be harvested and converted into a structured format using web scraping technologies [2]. To automate the data harvesting process, web scraping tools or software are used [1].

The tools can analyze the webpage or website and collect the required data. Following data collection from the website(s), the data is parsed into a file or database to structure the data before it can be processed. Adoption models, if they exist, will be investigated for applicability and use. Because it can be difficult for businesses to adopt web scraping technology, research will be conducted on the use of web scraping adoption models as well as generic information technology adoption models to determine whether they apply to web scraping technology. This paper comprises six sections. After the introduction in section 1, we have discussed the various definitions of web scraping and the motivation behind the work in section 2. In section 3, we have discussed various applications of web scraping. In the next section, i.e., section 4, we have presented the available tools for web scraping. Finally, in the fifth section, we have concluded our paper.

II. DEFINITIONS AND MOTIVATIONS

An automated technique known as web scraping is used to retrieve large amounts of data from websites. Usually, the internet contains unstructured data. Web scraping aids in the collection and storage of unstructured data. There are several methods for scraping websites, including the use of internet services, APIs, or custom programs. Web scraping, also known as web extraction or harvesting, is defined as "a technique to extract data from the World Wide Web (www) and save it to a file system or database for later retrieval or analysis" by Wikipedia. Web scraping can be extremely beneficial in this day and age when data retrieval is required. Fig. 1 depicts the web scraping procedure. Web scraping is made up of two parts: a web crawler and a web scraper. To put it simply, the web crawler is a horse, and the scrapper is a chariot. The scrapper is led by the crawler, who extracts the requested data.

The Crawler

Scraping the internet using Python, a web crawler is commonly referred to as a "spider." It is an artificial intelligence technology that crawls the internet to index and search for content via given links. It looks up the information requested by the programmer.

The Scrapper

Python for Web Scraping A web scraper is a dedicated tool designed to quickly and effectively extract data from multiple websites. The design and complexity of web scrapers vary greatly depending on the project. "The main goal of Web Scraping is to extract information from one or more websites and process it into simple structures such as spreadsheets, databases, or CSV files," proposed by Diouf et al. [3-4]. Web scraping is done physically and through software that simulates human web surfing to collect specific information from websites. Scraping websites can be done in a variety of ways, including using internet services, APIs, or own programmes.

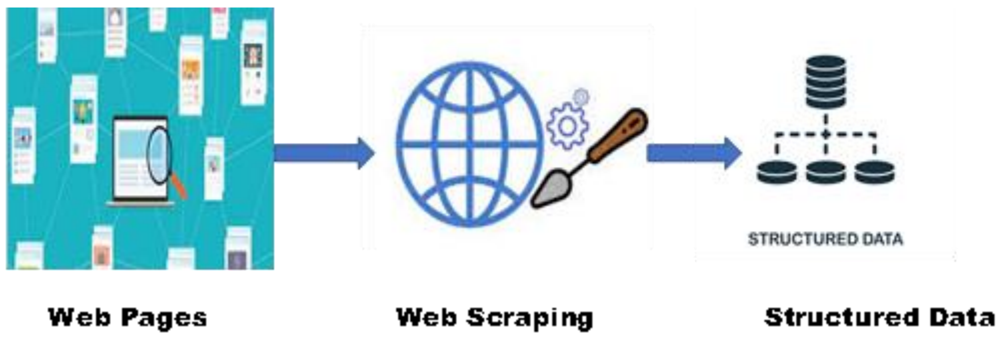

Fig. 1. The Basic Process of Web Scraping

According to Saurkar et al. [5] web scraping is "a technique used to crop information from web pages based on script routines." Those documents, according to them, are written in either HTML or XHTML.

These documents are typically represented using a hierarchical document object model, or simply a DOM tree. According to Chaulagain et al. [6] "web scraping is one of the major sources for the extraction of unstructured data from the internet; we have analyzed the scraping process when introduced to a large amount of data extraction." According to Techopedia[7], "Web scraping is essentially a form of data mining." Other experts [8] define web scraping as "the process of retrieving or scraping data from a website, also known as web data extraction." Web scraping, as opposed to the tedious, mind-numbing process of manually extracting data, uses intelligent automation to retrieve hundreds, millions, or even billions of data points from the internet's seemingly limitless frontier." According to Vargui et al. [9], "web scraping is a set of techniques used to automatically get some information from a website rather than manually copying it." According to them, the basic goal of the market's available web scrapers is to select and pick the required data and information and then combine it into new web pages.

"Web scraping bridges the gap between human-understandable and machine-readable data and opens up a new world of data to researchers by automatically extracting structured data sets from human-readable content," stated Mitchell [10]. Web scrapers are defined as "a tool that accesses web pages, finds specified data elements on the page, extracts them, transforms them if necessary, and finally saves these data as a structured data set" by Boeing et al. [11]. The most valuable asset in the world is information, but to retrieve it, user need data. Data, the second most important asset, is not available to everyone. Because everyone does not have access to the data they require, web scraping has risen to the surface.

III. APPLICATIONS OF WEB SCRAPING

Each emerging framework gains importance as its applications broaden. The same is true for web scraping. Many intellectuals [12-17] discussed various web scraping applications.

A. Cost Analysis: Services like ParseHub use web scraping to collect data from online shopping website and compare product prices.

B. Email address collection: Many businesses that use email as a marketing medium use web scraping to collect email ID and then send bulk emails.

C. Web Scraping for social media: Web scraping is used to collect data from social media websites such as Twitter to determine what's trending.

D. R&D: Web scraping is used to collect a large amount of data (Statistics, General Information, Temperature, etc.) from websites, which is then analysed and used to conduct surveys or R&D.

IV. WEB SCRAPING TOOLS

Web scraping tools are designed to help businesses collect open-source web data, such as:

- Synthesized
- Cleaned
- Structured
- Processed
- Analysed by teams and algorithms.

Web scraping can be done manually, but it is a time-consuming and resource-intensive effort, so many businesses prefer to use a tool to help automate this process.

1. Most common use cases for which businesses are currently employing web scraping tools:

A. Market Research: Industries seeking to introduce new products or enter new markets gather information on potential target audiences while also investigating successful competitor activities that can be replicated/learned from.

B. Stock Market Data: Hedge funds, portfolio managers, and venture capitalists all collect financial data such as share market capacity, enterprise media articles, and growth based on employee count or geospatial data (e.g. satellite imagery on the progress of a building site or factory).

C. Travel Aggregation: To better compete, online travel agencies (OTAs) gather real-time information about competing sites' vacation bundles, special offers, and flight/car rental/hotel pricing.

D. Food Delivery Market: As demand for food delivery has increased over the last two years, companies are increasingly looking to collect restaurant menu data, trending cuisines via search (Chinese? Japanese?, etc.), and order volume based on consumer geolocation.

E. Collection of Search Engine Optimization (SEO) / Search Engine Results Pages (SERPs): Many consumer journeys start with a simple search query, propelling businesses to the top of search engine results. As a result, they gather and analyse top search results for relevant search queries and keywords in their industry in order to optimise their own pages and rank higher in the future.

F. Website Testing: Companies that build sites/apps for different geographies or that launch new User Experiences (UX) and User Interfaces (UI) use web scraping tools to view front-end results from a consumer standpoint. This allows them to improve their Quality Assurance (QA) and load balancing.

G. eCommerce: This is a highly competitive field with many value-conscious customers. Product pricing, customer reviews, Sell-Through Rates (STRs), and other data points are collected by vendors, marketplaces, and brands to optimize item listings, design, and production lines to capture higher conversion rates.

H. AdTech: Marketing teams and agencies use web scraping tools to ensure that localized campaigns are shown to target audiences with the correct copy, images, and URLs. They also gather data on competitor ad campaigns to gain insights and optimize campaigns for higher click-through rates (CTRs).

I. Social Media for Marketing: Companies use web scraping tools to gain insights into their target audience's social sentiment, to find influencers with whom they can collaborate, and to identify posts that consumers are engaging with so that they can join the narrative and generate newfound interest.

2. Pros and Cons of Web Scraping Tools

2.1 Web Scraping Tool Pros

A. Freeing up Resources: Scraping for web data on your own necessitates the development and upkeep of software. Cloud servers, networks, and APIs are all included (Application Programming Interfaces). Furthermore, teams of engineers, IT personnel, and DevOps are required to clean and structure data points as well as perform code enhancements to address site architecture changes. Web scraping tools allow businesses to offload this burden to a third-party provider, freeing up internal resources for product development and user experience enhancement.

B. Tapping into Data Collection know-how: Scraping open-source data can be difficult; some target sites may block multiple information requests from a single IP address (rate limitations). Other sites serve false information to IP addresses that have been identified as suspicious or as belonging to a competing entity. Web scraping tools have created global peer networks, as well as technology that uses Machine Learning and Retry Logic to navigate these issues with ease, saving time and putting projects on hold.

C. No-code Based Agility: Web scraping can be a difficult task for small and medium-sized businesses (SMBs) with limited manpower. Furthermore, team leaders and portfolio managers want to be able to access critical data points without having to delegate tasks to other departments only to receive data points that are no longer relevant due to backlog. Web scraping tools enable anyone in the company (marketing/sales/design) to gain real-time access to relevant data with no coding. They can also enable and disable data collection jobs based on current needs, allowing for workflow/budget flexibility.

2.2 Web scraping tool cons

A. Maintaining Information Security/ Market Dominance: Some businesses may prefer to retain complete control over all of their data collection systems. Their belief in developing and maintaining proprietary data scraping mechanisms may be related to data security, maintaining a hermetic informational advantage over competitors, and/or ensuring their Unique Sales Proposition (USP) in the context of a product whose value is derived from data (e.g., a stock trading platform or an eCommerce vendor's dashboard). Because web scraping tools use end-to-end encryption, parsing information into millions of information particles that can only be deciphered by the receiving party, these concerns are typically fear/emotion-driven rather than fact-based. Compliance teams and real-time mechanisms that detect and stop malicious activity monitor data collection networks.

Data Collector is the ultimate scraping tool. Data Collector is a web data collection tool that is fully automated and requires no infrastructure. It is extremely simple to use and comes with ready-made templates to help businesses get results faster. These templates include:

E-commerce: Amazon/Alibaba/Walmart

Social media: Facebook/Instagram/TikTok

Travel sector: Kayak/Booking/Airbnb

2.3 Features of Scraping Tool

A. Simple, no-code web data collection: Instead of hiring data acquisition specialists who specialize in proxy management and web data extraction, team members can simply use 'Click & Collect' technology to obtain the information they require.

B. Fully automated data structuring and cleaning: AI capabilities handle all necessary data processing so that it is ready for analysis by management and/or algorithms.

C. Data delivery: When data is delivered, it is matched, synthesized, and structured.

D. Scalability and adaptability in data collection: Data Collector provides companies with increased agility by serving as an alternative to permanent data-related overhead. When a new client, Proof of Concept (PoC), or discussion of entering a new target market arises, the 'data collection machine' can be activated. When there is no need for it, it can be turned off and funds can be allocated elsewhere.

E. Compliance and best practices that are unrivalled in the industry: Ongoing systemic log reviews, usage monitoring, Know Your Customer (KYC) calls, code-based prevention/technological response mechanisms, and an independent Compliance Department are all part of this.

3. Web Scraping Tools

3.1. Scrapy

Scrapy is an open-source web scraping framework written in Python. It is designed to crawl websites and extract structured data from them, making it an ideal tool for web data mining, information processing, and automated testing. With Scrapy, users can build a spider that defines how to extract data from a website. The spider will then navigate through the site, following links and collecting data according to the rules defined. Scrapy provides a powerful mechanism for handling cookies, redirects, and other aspects of web crawling, making it a reliable tool for scraping even the most complex websites [12-17].

Scrapy also includes an item pipeline, which allows a user to process and store the data extracted in a variety of formats. Users can use Scrapy to store data in CSV, JSON, XML, or even in a database such as MySQL, PostgreSQL, or MongoDB. Scrapy provides many features to make web scraping easier, including support for handling common web scraping tasks such as pagination, handling of forms, and managing user sessions. It also includes a command-line tool for managing and running spiders, making it easy to automate web scraping tasks.

Overall, Scrapy is a powerful and flexible web scraping framework that makes it easy to extract data from websites in a structured and efficient manner. Whether collecting data for research, building a web application, or just automating tasks, Scrapy provides an excellent solution for web scraping.

Advantages

• Scheduled scrapy requests

Disadvantages

• Python 2.7 is the sole supported version of Scrapy.
• The operating system affects how the installation is done.

2. Heritrix

Heritix is an open-source web crawler designed for web archiving, web content mining, and data extraction tasks. Developed by the Internet Archive, it is widely used by academic and cultural heritage institutions, as well as commercial organizations, to collect and preserve web content for future use [12-17].

Heritrix is written in Java and can be run on multiple platforms, including Windows, Mac, and Linux. It supports multiple protocols such as HTTP, HTTPS, FTP, and DNS, and can be customized to handle specific web scraping requirements. The tool is highly configurable, allowing users to set rules for crawling, filtering, and indexing web content.

One of the key features of Heritrix is its ability to perform focused crawling, which means it can target specific portions of a website or a specific domain, rather than crawling the entire web. This makes it an efficient tool for targeted web scraping tasks. Heritrix also supports distributed crawling, which means multiple instances of the tool can be run in parallel to speed up the scraping process.

Heritrix produces output in a variety of formats, including WARC (Web ARChive), ARC (Internet Archive ARC), and CDX (Compressed Index). These formats allow the crawled data to be easily stored, searched, and accessed by other tools.

Overall, Heritrix is a powerful and flexible web scraping tool that is widely used in the web archiving and data mining communities. Its customizable and configurable nature makes it suitable for a wide range of web scraping tasks.

Advantages:

- Pluggable modules that can be changed
- Concerning the robot.txt and Meta robot tags, a web-based interface
- Fantastic extensibility

Disadvantages:

- Crawlers with just one instance are unable to cooperate.
- When machine resources are constrained, complicated operations are necessary.
- Only Linux is formally tested and supported.
- Each crawler runs on its own, therefore the update doesn't need to be changed.
- In the event of hardware and system failure, recovery options are constrained.
- Performance improvement gets little attention.
- Heritrix is only a crawler tool without a search engine, in contrast to Nutch.
- Need to apply a sophisticated algorithm comparable to Pagerank to sort the crawled websites.

3. Web-Harvest

Web Harvest is a Java-based open-source web scraping and data extraction tool. It allows users to extract data from web pages and transform it into various output formats such as XML, CSV, and JSON [12-17].

Web Harvest is easy to use and can be run from the command line or within a Java application. It supports a variety of web protocols such as HTTP, HTTPS, FTP, and SFTP, and can handle different types of data such as text, images, and multimedia files.

One of the key features of Web Harvest is its ability to extract data from web pages using XPath and CSS selectors. This means that users can easily target specific elements of a web page, such as text, links, and images, for extraction. Web Harvest also allows users to define their own extraction rules using regular expressions, making it a flexible tool for a wide range of web scraping tasks.

In addition to data extraction, Web Harvest supports data transformation and cleansing. It includes a variety of built-in functions for manipulating and formatting data, such as date/time parsing, string manipulation, and mathematical operations. This makes it easy to clean and structure scraped data for further processing and analysis.

Web Harvest can be configured using an XML-based configuration file, which allows users to define the scraping rules and specify the output format. The tool is highly customizable, with a wide range of options for handling cookies, authentication, and HTTP headers.

Overall, Web Harvest is a powerful and flexible web scraping tool that is widely used in the data mining and analytics communities. Its ease of use and flexibility make it suitable for a wide range of web scraping tasks, from simple data extraction to complex data transformation and analysis.

Advantages:

- Highly capable text and XML manipulation processors for data handling and control flow
- The variable context is used to store and use variables.
- True scripting languages are supported, and can be easily integrated into scraper configurations.

4. MechanicalSoup
MechanicalSoup is a Python library that helps automate web interactions. It allows developers to interact with web pages and automate form submissions by simulating the interaction of a human user with a web browser [12-17].
MechanicalSoup provides a simple, easy-to-use API for sending requests, parsing HTML, and interacting with web pages. It is built on top of the popular Python libraries Beautiful Soup and Requests, which allow users to parse HTML and interact with HTTP requests respectively. This makes it an ideal choice for web scraping and automation tasks.
With MechanicalSoup, developers can write Python code to automate interactions with web pages, including submitting forms, clicking buttons, and following links. This can be particularly useful for automating repetitive tasks such as data entry, web testing, and web scraping. MechanicalSoup also supports handling cookies, authentication, and other common web interactions.
MechanicalSoup is designed to be easy to use and accessible to developers of all skill levels. It provides an intuitive API for interacting with web pages, and the library's documentation is extensive and well-written. Additionally, MechanicalSoup is an open-source library, so it is free to use and modify as needed.
Overall, MechanicalSoup is a powerful and flexible Python library that makes it easy to automate web interactions and perform web scraping tasks. It is an ideal choice for developers who need to interact with web pages programmatically and want a simple, accessible library to do so.
Advantages:

- Capability to simulate human behaviour
- Exceptionally fast for scraping relatively simple websites
- CSS and XPath selectors are supported.

5. Apify SDK
Apify SDK is a powerful open-source tool for web scraping and automation. It is a comprehensive platform for developers to build, run, and manage web scraping and automation tasks at scale. Apify SDK allows developers to build custom web scraping and automation solutions, and provides an extensive set of tools and features to facilitate this process [12-17].
The Apify SDK provides a collection of pre-built tools and libraries that allow developers to write code in JavaScript or Node.js to create web scraping and automation tasks. These tasks can be executed locally or in the cloud using the Apify platform. The SDK provides tools for crawling web pages, parsing HTML, and handling data extraction and storage.
One of the key features of Apify SDK is its support for headless browsers. This means that developers can use tools like Puppeteer or Playwright to interact with web pages as if they were using a real browser. This allows for more complex interactions with web pages, such as clicking buttons, filling out forms, and performing other actions that are not possible with simple HTML parsing.
Apify SDK also includes a powerful data storage and management system. This allows developers to store data scraped from the web in a variety of formats, including JSON, CSV, and MongoDB. Additionally, the SDK provides tools for managing and monitoring scraping tasks, including the ability to schedule tasks and track their progress.
Overall, Apify SDK is a comprehensive and flexible tool for web scraping and automation. It provides developers with a powerful set of tools and features to build custom web scraping solutions, and can be easily integrated into existing workflows. With its support for headless browsers and robust data management features, Apify SDK is an ideal choice for developers who need to perform complex web scraping tasks
Advantages:

- Large-scale and high-performance scraping
- Apify Cloud with a proxy pool to avoid detection Built-in support for Node.js plugins such as Cheerio and Puppeteer

6. Apache Nutch
Apache Nutch is an open-source web crawler software project that is widely used for data extraction and web searching purposes. It is designed to handle large amounts of data and can crawl billions of web pages. Nutch provides a scalable architecture and modular framework for the development of web crawlers and search engines [12-17].
One of the key features of Nutch is its distributed architecture, which enables it to run on a cluster of computers, thereby enhancing its scalability and performance. It uses Hadoop Distributed File System (HDFS) and MapReduce to distribute the crawling and indexing tasks across multiple nodes in the cluster.
Nutch supports various protocols, such as HTTP, HTTPS, FTP, and NNTP, and can be configured to crawl websites, RSS feeds, and other types of web content. It also supports pluggable parsers, which can be used to extract structured data from web pages. The extracted data can be stored in various formats, including Apache Solr, Elasticsearch, and HBase.
Apache Nutch is widely used in various applications, such as web search engines, web archiving, and data mining. Its modular architecture and extensibility make it a popular choice among developers who need to build customized web crawlers and search engines.
Overall, Apache Nutch is a powerful and flexible web crawler software that provides a robust solution for data extraction and web searching. It is a mature and stable project that has been used by many organizations and has a large and active community of users and contributors.

DOI: 10.17148/IJARCCE.2024.13596

Advantages:
- Highly extensible and scalable
- Comply with txt rules
- Active community and development
- protocols, storage, pluggable parsing, and indexing

7. Jaunt

Jaunt is a web scraping and automation tool that allows users to extract data from websites with ease. It is a Java-based library that provides a simple API for web scraping and web automation tasks [12-17].

One of the key features of Jaunt is its ability to handle dynamic web content. It uses an embedded web browser to render web pages and execute JavaScript, which enables it to scrape websites that use AJAX, JavaScript, and other dynamic web technologies.

Jaunt provides a variety of methods for locating and extracting data from web pages, including CSS selectors, XPath expressions, and regular expressions. It also provides advanced features such as form filling, file uploading, and handling cookies and sessions.

Jaunt can be integrated with other Java applications and frameworks, such as Spring and Hibernate, to provide a seamless web scraping experience. It also supports various output formats, such as JSON, XML, and CSV, making it easy to integrate with other data processing tools.

Overall, Jaunt is a powerful and versatile web scraping and automation tool that provides a simple and intuitive API for developers. Its ability to handle dynamic web content, along with its support for various output formats and integrations, makes it a popular choice among developers who need to extract data from websites.

Advantages:
- Process individual HTTP Requests/Responses
- Interfacing with REST APIs is simple
- Support for HTTP, HTTPS, and basic authentication
- RegEx-enabled querying in DOM and JSON

8. Node-crawler

Node Crawler is an open-source web crawling and scraping library for Node.js, a popular server-side JavaScript runtime environment. It provides a simple and flexible API for developers to extract data from websites and automate web-based tasks [12-17].

Node Crawler uses a callback-based approach to handle the crawling and scraping of web pages. It supports various protocols, such as HTTP and HTTPS, and can be configured to crawl and scrape specific sections of a website, such as links, images, or text.

One of the key features of Node Crawler is its support for parallel crawling. It can execute multiple crawling and scraping tasks concurrently, which enables it to handle large amounts of data and improve performance.

Node Crawler provides various options for customizing the crawling and scraping behaviour, such as setting the maximum number of requests per second, the maximum depth of crawling, and the user agent string to use when making requests.

Node Crawler also supports pluggable middleware, which can be used to customize the behaviour of the crawler and add additional functionality. It also integrates with other Node.js libraries and frameworks, such as Express and Hapi, making it easy to build web-based applications that include web scraping functionality.

Overall, Node Crawler is a powerful and flexible web crawling and scraping library that provides a simple and intuitive API for developers. Its support for parallel crawling, customization options, and middleware make it a popular choice among developers who need to extract data from websites or automate web-based tasks.

Advantages:
- Flow control
- Various URL request priorities
- Configurable pool size and retries
- Server-side DOM & fully automated jQuery induction with Cheerio (default) or JSDOM

9. PySpider

PySpider is an open-source web scraping and web crawling framework written in Python. It provides a simple and flexible API for developers to extract data from websites and automate web-based tasks [12-17].

PySpider uses a web-based user interface to configure and execute web scraping and crawling tasks. It supports various web protocols, such as HTTP and HTTPS, and can be configured to extract data from websites, RSS feeds, and other types of web content.

One of the key features of PySpider is its support for distributed crawling. It can be deployed to a cluster of machines, which enables it to handle large amounts of data and improve performance.

PySpider provides various options for customizing the web scraping and crawling behavior, such as setting the maximum depth of crawling, the user agent string to use when making requests, and the number of concurrent requests to execute.
PySpider also supports pluggable spiders, which can be used to customize the behavior of the crawler and add additional functionality. It also integrates with other Python libraries and frameworks, such as Django and Flask, making it easy to build web-based applications that include web scraping functionality.
Overall, PySpider is a powerful and flexible web scraping and crawling framework that provides a simple and intuitive API for developers. Its support for distributed crawling, customization options, and pluggable spiders make it a popular choice among developers who need to extract data from websites or automate web-based tasks.
Advantages:

- A powerful web interface that includes a script editor, task monitor, project manager, and result viewer.
- The message queues are RabbitMQ, Beanstalk, Redis, and Kombu.
- Architecture distributed

10. StormCrawler

Storm Crawler is an open-source web crawling framework designed to run on Apache Storm, a distributed real-time computation system. It is written in Java and allows for the efficient and scalable processing of web data [12-17].
Storm Crawler has a modular architecture that allows for customization and extension. It includes several built-in components for fetching, parsing, indexing, and storing web content, as well as tools for monitoring and managing the crawling process.
The framework also supports a variety of data sources, including RSS feeds, sitemaps, and various social media platforms. Additionally, it can be configured to extract and analyze specific types of content, such as images or video.
One of the key features of Storm Crawler is its ability to distribute crawling tasks across a cluster of machines, allowing for high-throughput and fault-tolerant crawling. It also includes features such as URL filtering and deduplication to ensure that only relevant and unique content is processed.
Overall, Storm Crawler is a powerful and flexible tool for web data crawling and processing, suitable for a wide range of applications including web search, data mining, and social media analysis.
Advantages:

- Highly scalable and suitable for large-scale recursive crawls
- Simple to extend with additional libraries
- Excellent thread management, reducing crawl latency

The Table 1 below shows the comparison between different Web Scraping Tools [12-19].

Table 1 Comparison of Different Web Scraping Tools

S.No	Web Scraping Tool	Language Used	Prerequisites	Advantages
1.	Scrapy	Python	Computer Programming XPath is a plus	Cross-platform Built-in service called Scrapyd
2.	Heritrix	Java	Java	Replaceable pluggable modules Excellent extensibility
3.	Web-Harvest	Java	Java, HTML, XML	Highly capable text and XML manipulation processors True scripting languages are supported
4.	MechanicalSoup	Python	HTML, XML	fast for scraping relatively simple websites CSS and XPath selectors are supported
5.	Apify SDK	Java Script	Node.js, HTML	Large-scale and high-performance scraping
6.	Apache Nutch	Java	Java runtime (JRE) and development environment (JDK),Hadoop	Highly extensible and scalable

7.	Jaunt	Java	HTML	Process individual HTTP Requests/Responses Interfacing with REST APIs is simple
8.	Node-crawler	Java Script	Java Programming, Node.js	Flow control Various URL request priorities Configurable pool size and retries Server-side DOM & fully automated jQuery induction with Cheerio (default) or JSDOM
9.	PySpider	Python	Python	A powerful web interface that includes a script editor, task monitor, project manager, and result viewer.
10.	Stromcrawler	Java	Docker	Highly scalable and suitable for large-scale recursive crawls Simple to extend with additional libraries Excellent thread management, reducing crawl latency

V CONCLUSION

The contemporary literature on web scraping applications across domains, web scraping methods, and web scraping tool usage is reviewed in this study. Utilizing this research to enhance our online scraping methodology, we found that the majority of web scrapers are somewhat generic and similar in nature, intended to do simple and typical tasks. Due to its speed, extensibility, and power, Web Scrapy yields superior results when comparing the features and performance of various tools and frameworks.

REFERENCES

[1] Sirisuriya, S.C.M. De S. 2015. A Comparative Study on Web Scraping. URI: http://ir.kdu.ac.lk /handle/345/1051

[2] Schintler, L.A., McNeely, C.L. 2017. Encyclopedia of Big Data, Web scraping, ch483-1. ISBN: 978-3-319-32001-4

[3] Diouf, Rabiyatou, Edouard Ngor Sarr, Ousmane Sall, Babiga Birregah, Mamadou Bousso, and Sény Ndiaye Mbaye. "Web Scraping: State-of-the-Art and Areas of Application." In 2019 IEEE International Conference on Big Data (Big Data), IEEE, (2019). pp. 6040-6042

[4] Priya Matta et al., International Journal of Advanced Trends in Computer Science and Engineering, 9(5), September - October 2020.

[5] Saurkar, Anand V., Kedar G. Pathare, and Shweta A. Gode. "An Overview On Web Scraping Techniques And Tools." International Journal on Future Revolution in Computer Science & Communication Engineering 4, no. 4 (2018): 363-367.

[6] Chaulagain, Ram Sharan, Santosh Pandey, Sadhu Ram Basnet, and Subarna Shakya. "Cloud-based web scraping for big data applications." In 2017 IEEE International Conference on Smart Cloud (SmartCloud), pp. 138-143. IEEE, (2017).

[7] Bradley, Alex, and Richard JE James. "Web scraping using R." Advances in Methods and Practices in Psychological Science 2, no. 3 (2019): 264-270.

[8] Landers, Richard N., Robert C. Brusso, Katelyn J. Cavanaugh, and Andrew B. Collmus. "A primer on theory-driven web scraping: Automatic extraction of big data from the Internet for use in psychological research." Psychological methods 21, no. 4 (2016):
475.

[9] Vargiu, Eloisa, and Mirko Urru. "Exploiting web scraping in a collaborative filtering-based approach to web advertising." Artif. Intell. Research 2, no. 1 (2013): 44-54.

[10] Mitchell, Ryan. "Web scraping with Python: Collecting more data from the modern web. " O'Reilly Media, Inc., 2018.

[11] Tengku Mazlin, Tengku Ab Hamid, Roselina Sallehuddin, Zuriahati Mohd Yunos and Aida Al,“Ensemble Based Multi Filters Algorithm for Tumor Classification in High Dimensional Microarray Dataset”, International Journal of Advanced Trends in Computer Science and Engineering, Vol. 8, Issue.1.6, pp. 116-123, (2019)
[12] https://www.javatpoint.com/web-scraping-using-python
[13] Makino, Yuma, and Vitaly Klyuev. "Evaluation of web vulnerability scanners." In 2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems:Technology and Applications (IDAACS), vol. 1, pp.399-402. IEEE, 2015.
[14] Junjoewong, Lalita, Supatsara Sangnapachai, and Thanwadee Sunetnanta. "ProCircle: A promotion platform using crowdsourcing and web data scraping technique." In 2018 Seventh ICT International Student Project Conference (ICT-ISPC), pp. 1-5. IEEE, 2018.
[15] Indra, Evta, and Tamarai Dinesh. "Designing Android Gaming News & Information Application Using Java-based Web Scraping Technique."In Journal of Physics: Conference Series, vol. 1230,no. 1, p. 012069. IOP Publishing, 2019.
[16] Le, Quang Thai, and Davar Pishva. "Application of Web Scraping and Google API service to optimize convenience stores' distribution." In 2015 17th International Conference on Advanced Communication Technology (ICACT), pp. 478-482.IEEE, 2015.
[17] Barcaroli, Giulio, Alessandra Nurra, Marco Scarnò, and Donato Summa. "Use of web scraping and text mining techniques in the state survey on information and communication technology in enterprises." In Proceedings of the quality conference, pp. 33-38.
[18] https://brightd-scraping-tool
[19] https://www.scrapehero.com/best-web-crawling-tools-and-frameworks/