JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

WEB SCRAPER: TOOL FOR REVIEW EXTRACTION

[1]D Rajeev,[1]Arun Badiger,[1]Prabhu A Angadi,[2]Prof. Reshma B

[1]Student, [2]Assistant Professor
[1]Department of Computer Science and Engineering,
[1]Srinivas Institute of Technology, Mangalore,India

Abstract:In the current digital landscape, customer reviews significantly influence e-commerce sales. From small items to major products like automobiles, companies rely heavily on authentic user feedback to enhance their services and drive sales. However, gathering reviews across diverse platforms can be cumbersome. This project addresses that challenge by using a web scraper to extract relevant review data efficiently. Leveraging Python, along with libraries such as BeautifulSoup and Selenium, the system processes dynamic content and includes machine learning for real-time analysis while maintaining ethical scraping standards.

***IndexTerms*–JavaScript, web design,userinterface (UI),front-enddevelopment,animations,interactive webtools, parsing, webtechnology, userengagement, front-end functionality, and data visualization.**

I. INTRODUCTION

In the digital age, user-generated reviews play an important role in shaping business success [8][9]. These reviews often offer valuable insights into customer satisfaction and help companies understand user preferences [10]. Efficiently collecting and analyzing this data is essential for data-driven decision-making. This project introduces a web scraper designed to extract such data from websites, focusing on reviews and related metadata like usernames, ratings, and timestamps. Reviews not only convey customer's opinions but also serve as a key metrics for the businesses to assess their products or services [10]. Collecting and analyzing these insights is essential for leveraging this data effectively.

This project focuses on development of web scraper designed to extract reviews and other things from our websites reliably and efficiently. A web scraper is an automated tool which navigates whole website to find the important details we specify to extract them. In this case we mainly focus on reviews from our website or other publicly available forms to extract the required data ethically like the username, review text, star ratings, and timestamps. The focus of this project is to reduce the tedious work of extracting the reviews which is usually a long process we try to automate it and reduce time spent on it as much as possible. This enables the organization to make a data-driven decisions that improve the product quality.

II. RELATED WORK

From the literature review,itis evident that advancementsin Web scraper have progressed significantly. Modern tech has provided with easy development of scrapers for specific needs reducing the work in the process and overall time spent.Aconsiderableamountofresearchhasbeendedicatedtoimprovingweb scrapers.Forexample, Researchers such as Latif [1] and Ismail [2] explored integrating BeautifulSoup with Selenium to extract reviews from e-commerce platforms like Amazon. They emphasized handling JavaScript-rendered content, where Selenium retrieves dynamic content and BeautifulSoup parses the HTML. Their study mainly focused on how to handle the JavaScript the mainly used in development of websites nowadays. BeautifulSoup effectively parses the HTML structure of a webpage, while Selenium interacts with dynamically rendered JavaScript elements, ensuring comprehensive data extraction from modern websites.

Inasimilarway, Bansal et al focused on overcoming the CAPTCHA and IP blocking [3], they proposed a combination of rotating proxies and delay mechanisms as solutions. These techniques mostly switch the IP addresses to prevent detection by anti-scraping tools, by simulating multiple users the likelihood of being blocked during scraping of getting blocked is reduced. Another method is to time the requests by introducing a time-delay between requests which mimic a real user further reducing the detection. These methods proved highly effective in not getting blocked and improved data scraping effectiveness.

Additionally, Potthast et al and companions emphasized on ethical considerations of the scraping surrounding a web scraper [6][7], a critical yet overlooked aspect of the topic. The explained about the importance of cases leading on legal disputes due to scraping and their violations. They further explained about the importance of respecting user and website policies and limiting scraping activity to avoid sever overload and exploring alternatives like public APIs for authorized data access.

III. METHODOLOGY

The WebScraper app allows users to extract specific data from websites and save it for later use. When the user opens the app, they see two options: Start Scraping and Saved Data. If the user chooses Start Scraping, they can either enter a website URL manually or select a saved template. After filling in the required data fields like product names, prices, or titles, they click the Start Scraping button

to begin. The app then processes the site and saves the extracted data in a structured format. If the user selects Saved Data, they can view, download, or delete previously scraped results. This ensures easy and reusable web data collection.

IV. SYSTEM DESIGN

The web scraper system to collect data of reviews and other things require several features design to ensure reliable, efficient, and user-friendly extraction of data, which include simple user interface and support of manual and automated inputs.

A. Automated Data Extraction

Web scraper extracts the data from a website without any manual effort or human intervention which makes it highly useful and fast.

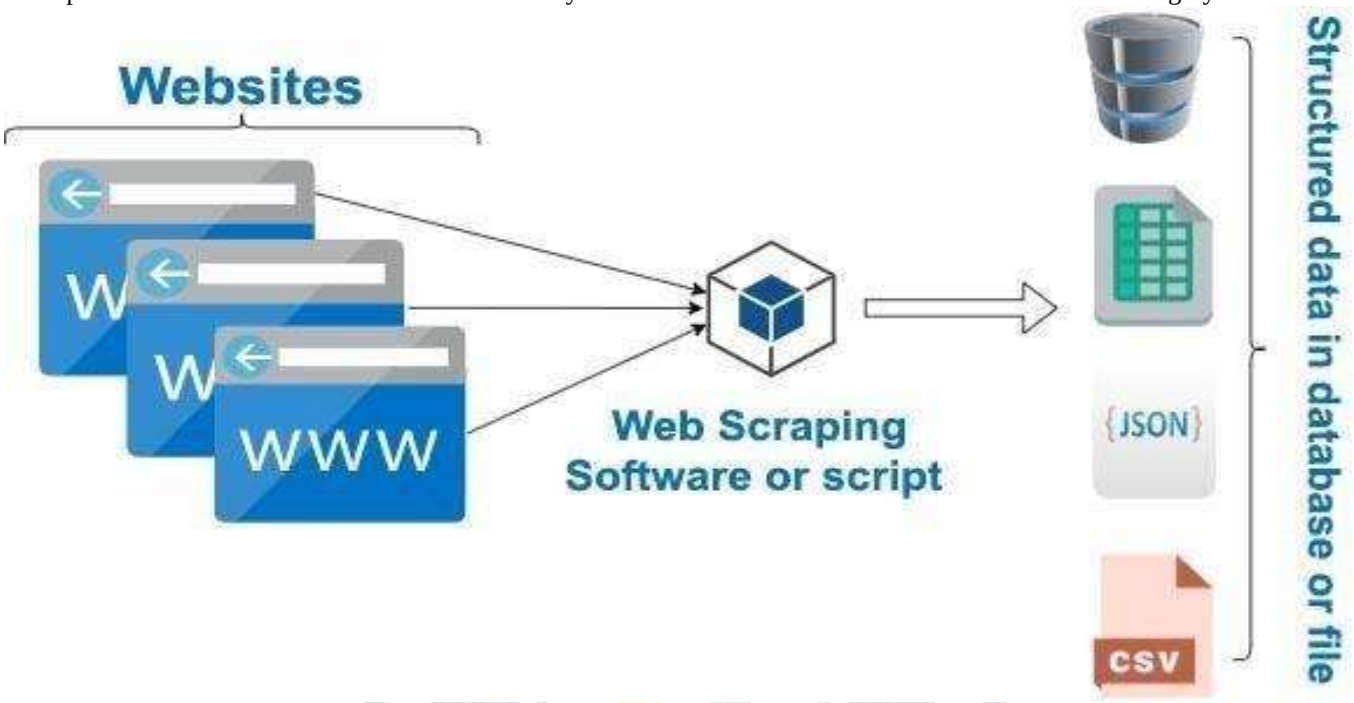

fig. 1 data flow diagram for web scraper project

Fig. 1 shows the flow of data in the web scraper project where the actual webs scraper with the URL of the websites accesses them and extract the useful data faster and store them in a structured database like tables with different types of data like username, review in different columns of the table making it easy for the organization to read and understand the extracted data.

B. Data Cleaning and Preprocessing

Once the data is extracted, it undergoes preprocessing, including the removal of duplicate entries, irrelevant text, and incomplete records. This ensures cleaner, more usable data for subsequent analysis [4][5]. It also removes the duplicate reviews making the data does not repeat even if extracted from different sources and makes the memory consumption little. Finally, it filters out the invalid or incomplete data entries as they do not help in analyzing that data and get meaningful details from them.

C. Data Storage

The data which is extracted by scraping is stored in the database or made available as a csv file. It also supports both local and cloud storage like MySQL, MongoDB. It also ensures scalability to ensure scraping of large data without any problems making extraction of large data easy.

D. User Interface Design

The user interface (UI) design of the project focuses on creating an easy to use for the end user point of view which makes it accessible to larger community. It also provides an area to enter the website URL or specific parameters like reviews, keywords, categories for scraping reviews. Its simple interface makes it easy to use and it being able to use both manual and automated inputs further help with the usability of the scraper.

fig. 2 interface screenshot

Fig. 2 shows the interface of the parser with inbuilt keywords to parse and its outcomes with URL of the website to parse. The design ensures that this information is clearly presented and easily accessible to users. The layout is optimized for readability and user engagement.

VI. RESULTS AND DISCUSSION

To assess the efficacy and resilience of the suggested web scraper, an extensive series of experiments was carried out on a live online shopping website—namely, the Skechers website—on a product page that contained many customer reviews. The purpose of this assessment was to examine the performance of the system on a number of key metrics, such as data extraction accuracy, parsing speed, error handling ability, user interaction, and scalability in processing large datasets.

1. *Accuracy and Completeness*

The scraper was initially set up to scrape user-generated reviews for a chosen product. The main performance indicators were:

- The number of pages crawled in total,
- The number of reviews scraped successfully,
- The precision of the scraped fields, and
- The duration taken to finish scraping.

Every review was scraped with its corresponding attributes: the name of the reviewer, star rating, review content, and timestamp. The data was then exported and saved in a CSV format, which made it easier for further manual verification. This validation process was crucial in determining missing or improperly parsed fields. The system had a high success rate of extraction in repeated trials, with only a very small percentage of data missing or malformed. This proved the tool's dependability in extracting structured review data from websites with moderately complex structures.

2. *Parsing Efficiency and Error Handling*

Parsing efficiency was evaluated using data extraction speed versus web element counts processed. The tool demonstrated effective parsing even when the target pages included dynamically embedded content like AJAX-loaded reviews and pagination based on JavaScript. When unexpected content layouts or HTML variations were introduced, the error-handling capabilities of the scraper—such as exception capture, retry logic, and fallback routines—enabled it to skip or log erroneous elements instead of bringing down the entire process.

Specifically, the scraper was tested against multiple review pages with different layouts and content loading patterns. It successfully handled:

- Missing fields such as absent reviewer names or dates,
- Dynamic loading of reviews (infinite scrolling),
- Broken or inaccessible links, and
- Temporary network interruptions.

Fallback strategies ensured continuity in data collection, and all failures were logged for review. These capabilities enhanced the tool's robustness, especially for real-world deployments where site structures frequently change.

3. *User Interface and User Experience Testing*

The front-end of the system was tested through beta testing sessions carried out with a small set of users, both technical and non-technical. The graphical user interface provided manual entry by users to enter target URLs and filter arguments like keywords or review types. Another automated mode was also tested, where input and predefined sites were utilized for unattended scraping.

User feedback revealed that the interface was responsive, intuitive, and took little training to use. Operations like starting a scrape, tracking progress, and downloading output data were seamless and easy to use.

4. *Scalability Testing*

To determine the scalability of the scraper, it was run on sites that had very high numbers of reviews, some with several thousand records. The system consistently performed stably and did not suffer any noticeable slowdowns in response time or accuracy as the volume of data grew. Resource use was tracked to make sure the system stayed responsive and did not consume local memory or processing resources.

These tests validated the scraper's ability to perform at scale efficiently, making it ready for enterprise-grade deployments where batch processing of numerous products or categories is necessary.

Summary of Results

The experimental testing proved that the web scraper satisfies the major performance requirements across the board:

- High precision in review extraction and field mapping.
- Effective parsing and processing of dynamic and static content.
- Robust error handling, guaranteeing uninterrupted operations.
- Ease of use, supporting rapid onboarding and efficient manual operation.
- Scaling to support big data and busy websites.
- All these outcomes confirm the readiness of the proposed system for real-world application in fields like market analysis, sentiment analysis, and online shopping analytics.

VII. CONCLUSION

This study proposes an effective and practical approach to automating the retrieval of user-generated data, specifically product reviews, using a web scraping system that is tailor-made. The tool tackles an important chokepoint in data acquisition—i.e., the human-intensive and time-consuming method of collection and organization of user feedback on dynamic websites. By using robust Python-based libraries like Selenium and BeautifulSoup, the system can crawl sophisticated web structures, deal with dynamic content, and extract structured data elements like review text, star ratings, reviewer identities, and timestamps.

The system complies with website policies and users' privacy through the implementation of measures like rate limiting, request delay mechanisms, and selective scraping to avoid overwhelming target servers. These ethical issues are crucial in ensuring legal and responsible data gathering practices, particularly when dealing with public web platforms. In summary, the constructed web scraper effectively fulfills its purpose of automating review collection, cutting down on human effort by a great margin, and enhancing access to data. With ongoing refinements in precision, scalability, and analytical strength, the system has immense potential to be a solid, ethical, and scalable platform for massive review mining and feedback analysis.

VIII. REFERENCES

[1] R. Latif, "Web Scraping for Data Analytics: A BeautifulSoup Implementation," *ResearchGate*, 2022. [Online]. Available: https://www.researchgate.net/publication/371467193

[2] A. M. Ismail, "Web Scraping of Google Reviews Using Selenium and BeautifulSoup," *LinkedIn*, 2023. [Online]. Available: https://www.linkedin.com/pulse/web-scraping-google-reviews-using-selenium-abdul-ismail-mohammad

[3] A. K. Jadhav, "Web Scraping for E-Commerce Website," *SSRN Electronic Journal*, 2023. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4794897

[4] G. Y. Lee et al., "A Survey on Data Cleaning Methods for Improved Machine Learning Model Performance," *arXiv preprint*, arXiv:2109.07127, 2021. [Online]. Available: https://arxiv.org/abs/2109.07127

[5] P.-O. Côté et al., "Data Cleaning and Machine Learning: A Systematic Literature Review," *arXiv preprint*, arXiv:2310.01765, 2023. [Online]. Available: https://arxiv.org/abs/2310.01765

[6] I. C. Mihai, "Why Is More Efficient to Combine BeautifulSoup and Selenium in Scraping," *Ovidius University Annals*, vol. XXII, no.2, pp.114–117,2022. [Online].

[7] E. Wu et al., "Towards Reliable Interactive Data Cleaning: A User Survey and Recommendations," in *Proc. HILDA '16: Workshop on Human-In-the-Loop Data Analytics*, 2016. [Online]. Available: https://www.cs.columbia.edu/~ewu/files/papers/cleaning-hilda16.pdf

[8] P. R. Arya and M. Jain, "Consumer's Preferences and the Influence of Online Reviews in the Context of Electronic Commerce," *IJRAR*, vol. 6, no. 1, pp. 670–675, 2019. [Online]. Available: https://ijrar.org/papers/IJRAR19K9941.pdf

[9] V. Gupta and H. Rai, "A Study on the Impact of Online Reviews on Consumer Decision Making in the Indian E-Commerce Market," in *Proc. AIMS Int. Conf.*, 2022. [Online]. Available: https://aims-international.org/aims22/22AProceedings/PDF/A518-Done.pdf

[10] Y. Yin, L. Luo, and G. Y. Lee, "Influence of Review Credibility and Reviewer Expertise on Purchase Intention," *Decision Support Systems*, vol. 142, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0304394021005176