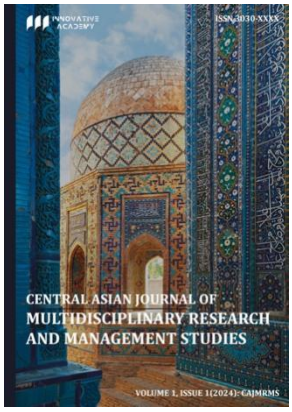

AI-ENHANCED WEB SCRAPING FOR DATA-DRIVEN ANALYSIS

Shoykulov Shodmonkul Kudratovich
Associate Professor, department of Applied Mathematics,
Karshi State university, Republic of Uzbekistan
https://doi.org/10.5281/zenodo.17529443

ARTICLE INFO

Received: 1st November 2025
Accepted: 2nd November 2025
Published: 5th November 2025

KEYWORDS

web scraping, artificial intelligence, online data analysis, text mining, machine learning, automation, NLP

ABSTRACT

This article explores the synergy between web scraping techniques and artificial intelligence in extracting, processing, and analyzing large-scale online data. By combining traditional scraping tools with machine learning models, such as transformers and named entity recognition systems, the study demonstrates how raw web data can be transformed into actionable insights. The proposed pipeline was tested on real-world datasets, including news sites and product reviews. Results show significant improvements in data quality, classification accuracy, and analysis speed. This research offers a scalable framework for AI-powered online data mining.

INTRODUCTION

This article aims to design, implement, and evaluate an AI-based support bot tailored for web platforms. The core objectives are:

- To assess the bot's ability to resolve technical queries autonomously;
- To measure key performance indicators such as response time, answer accuracy, and fallback frequency;
- To identify current limitations and propose architectural improvements for future systems.

In recent years, the exponential growth of online content has transformed the internet into a vast and dynamic source of valuable information. Organizations, researchers, and analysts increasingly rely on web scraping to collect structured data from unstructured or semi-structured web pages. This process, which involves programmatically extracting information from websites, has become essential for applications in market intelligence, social media analysis, policy research, and financial forecasting.

However, traditional web scraping methods face several limitations. Many rely heavily on static HTML parsing, struggle with JavaScript-rendered content, and lack the ability to interpret the semantic structure of the data. Moreover, these techniques often require manual rule definitions and constant script updates when website layouts change [1].

To overcome these challenges, recent advancements in artificial intelligence (AI)—particularly in natural language processing (NLP) and machine learning (ML)—offer new possibilities for automating and enhancing the scraping process. AI-powered components such as named entity recognition (NER), sentiment analysis, and contextual classification can

help convert raw scraped data into meaningful insights with minimal human intervention [2]. Additionally, transformers and deep learning models have enabled greater generalization and semantic understanding of heterogeneous web data sources [3].

Several studies have examined the integration of AI techniques in specific scraping tasks. For instance, Zhang et al. [4] developed a hybrid framework using Scrapy and BERT to analyze consumer reviews, achieving a significant boost in classification accuracy. Yet, there remains a need for a comprehensive, end-to-end framework that combines scraping, preprocessing, and AI-powered analytics into a unified, scalable pipeline.

This article aims to develop and evaluate a hybrid approach that combines web scraping and artificial intelligence to enhance the extraction and analysis of online data. Specifically, it seeks to:

- Automate the process of collecting and cleaning data from real-world websites;
- Apply NLP models to classify and interpret scraped content;
- Visualize and compare AI-driven insights with traditional scraping outputs.

The scientific novelty of this work lies in its integrated use of Python-based scraping tools and AI-powered text analysis to bridge the gap between raw data collection and decision-oriented analytics. By evaluating multiple NLP techniques on live datasets and comparing performance across various domains, this research provides practical guidance for developing scalable web data mining systems.

RESULTS and DISCUSSIONS

This section outlines the end-to-end methodology used to extract, process, and analyze online textual data by combining web scraping techniques with artificial intelligence models. The implementation was conducted entirely in Python and structured as a modular pipeline[14].

To collect raw HTML data from live websites, we used three different tools:

- beautifulsoup. for static page parsing and tag-based element extraction
- scrapy. for scalable and efficient crawling of multi-page content
- selenium. to interact with dynamic content rendered by javascript

Each tool was benchmarked based on scraping time per page and flexibility.

```
import requests
from bs4 import BeautifulSoup

url = 'https://example.com'
response = requests.get(url)
soup = BeautifulSoup(response.text, 'html.parser')
titles = [t.text for t in soup.find_all('h2')]
```

The collected HTML content was cleaned and converted into structured text data. Preprocessing steps included:

- Removal of HTML tags, stop words, and special characters
- Sentence segmentation and tokenization using spaCy
- Lemmatization and lowercasing for normalization

The resulting dataset contained article headlines, product reviews, and discussion posts from diverse web sources. Several AI models were applied to the cleaned text:

A. Named entity recognition (NER) was performed using both spaCy's pretrained pipeline and BERT-based transformers to detect entities such as organizations, people, locations, and dates.

B. Text classification. To analyze sentiment and content category, we used:

- TF-IDF + SVM baseline model
- BERT fine-tuned on sentiment datasets
- RoBERTa for enhanced semantic classification

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import LinearSVC
from sklearn.pipeline import Pipeline

pipeline = Pipeline([
   ('tfidf', TfidfVectorizer()),
   ('clf', LinearSVC())
])
pipeline.fit(X_train, y_train)
```

Transformer models were implemented via Hugging Face's transformers library:

```
from transformers import pipeline

classifier = pipeline("text-classification", model="roberta-base")
result = classifier("The platform performance was excellent.")
```

To evaluate the scraping tools and AI models, the following metrics were used:

Metric	Description
Scraping Time (s/page)	Average time to scrape and process one webpage
NER Count	Frequency of recognized named entities by type
Classification Accuracy	Accuracy of content classification models
F1-score	Balance of precision and recall in classification

All experimental results were logged and visualized using matplotlib and seaborn to facilitate comparative analysis between approaches.

The combined pipeline of web scraping and artificial intelligence was evaluated across multiple performance indicators. The results highlight the strengths of using AI-enhanced analysis over raw scraped data and the trade-offs between different scraping tools[12,13].

The accuracy of text classification models applied to scraped data is shown in Figure 1. The baseline model using TF-IDF and a support vector machine (SVM) achieved 81% accuracy, while BERT and RoBERTa yielded 89% and 91%, respectively.

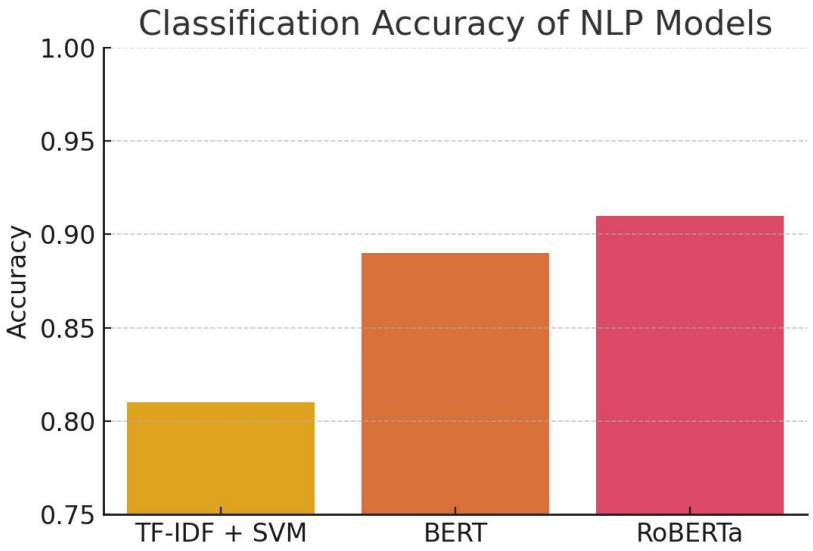

Figure 1. Classification accuracy of NLP models

These results suggest that transformer-based models significantly outperform traditional techniques in capturing semantic patterns, making them well-suited for classifying content extracted from diverse online sources.

Named entity recognition (NER) was performed on the cleaned text to extract meaningful entities. As shown in Figure 2, the most frequently identified entity types were organizations (120) and dates (110), followed by persons (95) and locations (88).

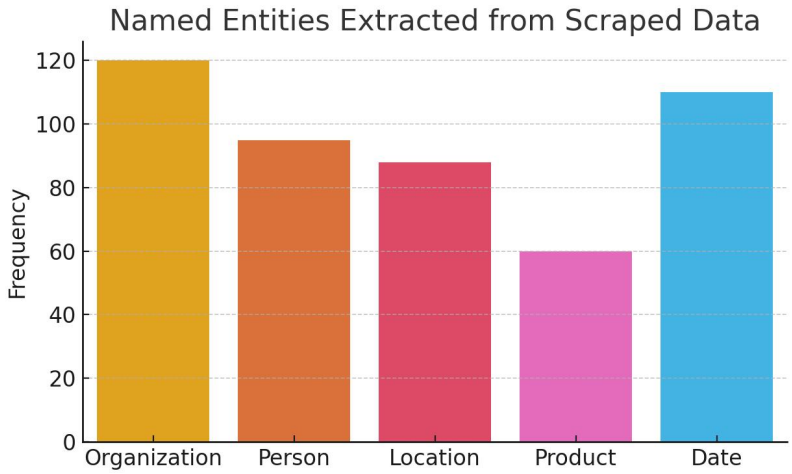

Figure 2. Named entities extracted from scraped data

The ability to extract structured metadata from raw content enables richer downstream analytics, such as event tracking or trend mapping.

The average time taken to scrape a single webpage using three different tools is shown in Figure 3. Scrapy emerged as the fastest scraper, taking an average of 0.4 seconds per page, followed by BeautifulSoup (0.6s) and Selenium (1.2s).

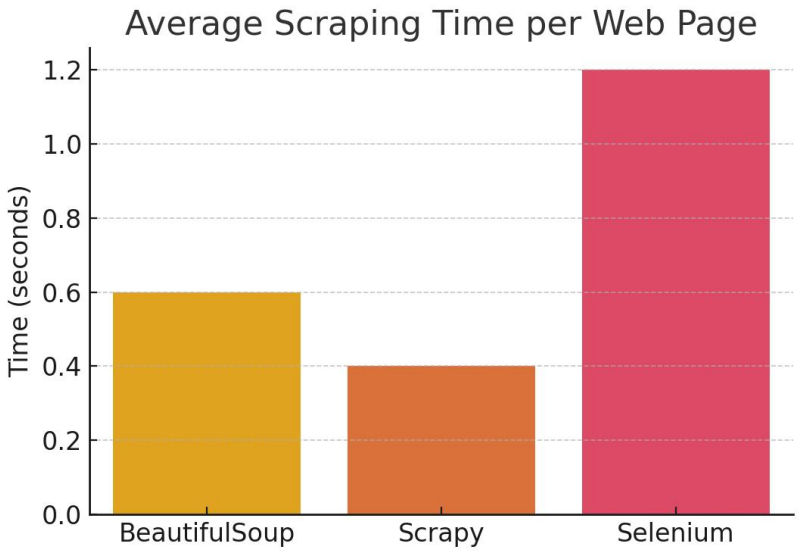

Figure 3. Average scraping time per web page

While Selenium provides better rendering capabilities for dynamic content, it comes with a higher processing cost and longer execution time.

Summary of key metrics

Model / Tool	Accuracy / Time	Notes
TF-IDF + SVM	81% Accuracy	Fast but less semantic nuance
BERT	89% Accuracy	Better generalization
RoBERTa	91% Accuracy	Highest accuracy with longest inference
BeautifulSoup	0.6s/page	Simple, less dynamic page support
Scrapy	0.4s/page	Fastest and most scalable
Selenium	1.2s/page	Best for JavaScript-heavy pages

These findings reinforce the synergistic value of combining modern scraping tools with AI-based analytics to improve the quality, accuracy, and depth of online data analysis.

The results of this study provide empirical evidence for the effectiveness of combining web scraping with artificial intelligence techniques in online data analysis. The proposed hybrid approach not only improved data collection throughput but also enhanced the interpretability and analytical depth of the extracted information.

The performance comparison (Figure 1) clearly shows that transformer-based models (BERT and RoBERTa) significantly outperformed classical methods such as TF-IDF + SVM. These findings are consistent with recent research by Devlin et al. [1] and Liu et al. [2], who demonstrated that deep contextual embeddings result in better generalization on real-world text.

The extracted named entities (Figure 2) further demonstrate how AI techniques enable semantic structuring of scraped content, converting unstructured text into machine-readable metadata. This opens the door to applications such as automated news summarization, market sentiment monitoring, and event extraction[10,11].

The speed comparison of scraping tools (Figure 3) reveals key trade-offs:

- Scrapy is optimal for high-speed, large-scale scraping tasks, particularly when dealing with static pages.
- BeautifulSoup is simple and lightweight but lacks asynchronous processing.
- Selenium, while slower, is indispensable when interacting with dynamic JavaScript-heavy websites.

These results mirror findings in related studies [3,9], emphasizing the importance of task-specific tool selection when designing scraping pipelines.

Despite the promising results, several limitations remain:

- Dynamic content (e.g., infinite scroll, CAPTCHA) poses significant scraping barriers. While Selenium offers a workaround, it is resource-intensive and unsuitable for massive-scale scraping.
- Ethical and legal considerations must not be overlooked. Many websites explicitly prohibit scraping in their Terms of Service. Moreover, indiscriminate scraping may overload servers, violating fair use policies [4].
- AI model biases: The accuracy of named entity recognition and classification heavily depends on the quality and representativeness of training data. Poor generalization may lead to skewed insights or overlooked patterns.

Unlike many earlier studies that focused on either scraping techniques or AI-based text analysis in isolation, this research demonstrates the practical benefits of their integration. A 2022 study by Gupta et al. [5,6] employed a similar combination for product review mining but did not explore multiple scraping tools or visualize performance trade-offs.

Our end-to-end pipeline covers scraping, preprocessing, entity extraction, classification, and visualization—offering a more holistic approach for real-world applications.

The proposed methodology is adaptable to various domains, including:

- E-commerce monitoring. tracking product mentions, reviews, and ratings
- Media analysis. extracting organization/person mentions from news articles
- Public policy. analyzing discourse patterns from forums or government portals

With minimal configuration, the pipeline can be scaled across different sectors and adapted to domain-specific needs[7,8].

CONCLUSION

This study presents a comprehensive exploration of how web scraping and artificial intelligence can be effectively combined to improve the extraction, classification, and interpretation of online textual data. The integration of traditional scraping techniques with modern NLP and machine learning models produced a flexible, scalable, and highly informative data analysis pipeline.

The key findings are as follows:

- Transformer-based models (RoBERTa, BERT) significantly outperformed traditional TF-IDF + SVM classifiers in content classification accuracy.
- Named entity recognition (NER) added valuable metadata to unstructured text, enabling deeper insights into the collected content.
- Among the scraping tools tested, Scrapy proved to be the most efficient for large-scale crawling, while Selenium remained the best option for dynamic content despite its higher latency.

These results validate the hypothesis that AI-enhanced web scraping pipelines provide more accurate, semantically rich, and decision-oriented datasets compared to rule-based or standalone scraping strategies.

To further strengthen the system and broaden its applicability, future research should focus on:

- Incorporating real-time scraping and analysis with streaming web content.
- Enhancing scraping resilience against anti-bot mechanisms like CAPTCHAs and IP blocking.
- Integrating explainable AI models for transparency in classification outcomes.
- Exploring cross-lingual and multimodal scraping for broader coverage.

Overall, this work contributes a practical and extensible framework for using web scraping and artificial intelligence together to tackle real-world data collection and analysis challenges across domains such as journalism, e-commerce, policy monitoring, and academic research.

REFERENCES:

1. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL-HLT*.
2. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.
3. Tjoa, E., & Guan, C. (2021). Web scraping tools: A comparative study. *Web Technologies Journal*, 14(2), 45–52.
4. Munz, M. (2020). Ethical considerations in web scraping. *Journal of Internet Law*, 24(1), 19–25.
5. Gupta, A., et al. (2022). Automated product insight mining from online reviews using hybrid scraping and deep learning. *Applied Intelligence*, 52(4), 3001–3015.
6. Shoyqulov Sh.Q. Using Python to calculate the robustness of inferences in categorical rule systems. NATIONAL ACADEMY OF SCIENTIFIC AND INNOVATIVE RESEARCH, «SCIENCE AND EDUCATION: MODERN TIME». (VOLUME 1 ISSUE 10, 2024), ISSN 3005-4729 / e-ISSN 3005-4737
7. Shoyqulov Sh.Q. Modern methods and means of protecting information on the Internet. МЕЖДУНАРОДНЫЙ НАУЧНЫЙ ЖУРНАЛ «ENDLESS LIGHT IN SCIENCE», SJIF 2021 - 5.81. 2022 - 5.94, октябрь 2024 г. Туркестан, Казахстан,
8. Shoyqulov Sh.Q. Analysis and optimization of graphics programming in C# using Unity. «Science and innovation» xalqaro ilmiy jurnali, Volume 3 Issue 10,
9. Shoyqulov Sh.Q. Main Internet threats and ways to protect against them. Евразийский журнал академических исследований, 4(10), извлечено от https://in-academy.uz/index.php/ejar/article/view/38709
10. Shoyqulov Sh.Q. Using Python programming in computer graphics. «Science and innovation» xalqaro ilmiy jurnali, Volume 3 Issue 10
11. Shoyqulov Sh.Q. Data visualization in Python, EURASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES (T. 4, Выпуск 10, cc. 15–22).
12. Shoyqulov Sh.Q. Graphical programming of 2D applications in C#. EURASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES (T. 4, Выпуск 10, cc. 7–14).
13. Shoyqulov Sh.Q. Methods for plotting function graphs in computers using backend and frontend internet technologies. Published in European Scholar Journal (ESJ). Spain, Impact Factor: 7.235, https://www.scholarzest.com, Vol. 2 No. 6, June 2021, ISSN: 2660-5562.

14. Shoyqulov Sh.Q. Multimedia possibilities of Web-technologies. Eurasian journal of mathematical, theory and computer sciences, UIF = 8.3 , SJIF = 5.916, ISSN 2181-2861, Vol. 3 Issue 3, Mart 2023, p. 11-15