Agentic AI for Multimodal Tabular Data Extraction

Surya Rayala srayala5@asu.edu
Hashwanth Sutharapu hsuthara@asu.edu
Ashish Thanga athanga@asu.edu
Dipanshu Singh dsingh47@asu.edu
Vibhu Dixit vdixit5@asu.edu
Sanjay Kumar skuma350@asu.edu

Abstract

This project developed an AI agent system for automatically detecting, extracting, and structuring multimodal tabular data from web sources into standard formats like HTML, JSON, and CSV. Efforts focused on the use of Selenium to scrape webpages to get screenshots of potential existing tables. Open-source LLMs like Gemini are used to extract tables in the form of HTML, CSV, and JSON. The current direction presents a robust automated pipeline to extract tables along images, from a given website, along with the option of cross-questioning from the extracted data. It preserves semantic relationships between tables and visualizations, addressing a critical gap in automated data processing. The final system can deliver a scalable, user-friendly pipeline, deployable via a Streamlit application, to streamline multimodal data extraction for applications in finance, scientific research, and e-commerce.

1 Introduction

Currently, in this data-driven world, huge oceans of knowledge lie trapped in web tables floating under financial reports, scientific papers, government statistics, and e-commerce interfaces. Tables are multi-modal structures where text is one of the elements besides images, symbols, and formatting, thereby posing a real challenge for automated extraction. Existing methods glean information either with HTML scrapping that fails on visually rendered tables or by means of computer vision, which at best gets the issues of text extraction and semantic comprehension wrong. Neither preserves the primary relationships amid textual and visual anterior elements that bestow full meaning upon multi-modal tables.

Subjecting this approach, the authors intend to solve the problem by designing an integrated artificial intelligence agent framework that combines web rendering, computer vision, optical character recognition, and large language models to detect, extract, and structure multimodal tabular data. The system performs webpage rendering with Selenium, uses it to capture high-fidelity screenshots, then bridges the gap between visual comprehension and structured data extraction workflows, thus allowing an organization to massively cut down on manual processing cost, unlock knowledge contained in siloed documents, and start automated analysis of multimodal tabular content on a large scale.

2 Problem Statement

This research addresses the fundamental challenge of automatically extracting and structuring multimodal tabular data from web sources while preserving the semantic relationships between textual and visual elements. Specifically, we aim to solve the following problems:

1. **Multimodal Detection:** How can we accurately identify and isolate tables that may be represented through diverse HTML structures, CSS layouts, or rendered as images across various webpage designs?

2. **Semantic Preservation:** How can we maintain the critical relationships between tabular data and embedded visual elements during extraction, ensuring these semantically linked components remain associated in the structured output?

3. **Format Standardization:** How can we transform heterogeneous table structures into standardized formats (HTML, JSON, CSV) that preserve both the structural integrity and multimodal characteristics of the original data?

4. **Extraction Accuracy:** How can we minimize errors introduced during OCR-based extrac-

tion, particularly for tables with complex layouts, merged cells, and embedded visual elements?

5. **Interactive Analysis:** How can we enable non-technical users to not only extract multimodal tabular data but also derive insights through natural language queries about the extracted information?

Our objective is to develop a comprehensive, scalable system that addresses these challenges through an integrated pipeline combining web rendering, computer vision, language models, and data transformation techniques. The system must be accessible to non-technical users through an intuitive interface while maintaining high accuracy across diverse web sources and table formats. Furthermore, it must support downstream applications in domains including finance, scientific research, and e-commerce, where multimodal tabular data plays a crucial role in decision-making processes.

3 Related Work

3.1 Knowledge-Aware Reasoning over Multimodal Semi-structured Tables

(Mathur et al., 2024)

The paper presents MMTABQA, a dataset meant for evaluating AI reasoning over multimodal tables that integrate text and images. While previous works concentrated on tables with text alone, real-world data usually feature visual elements, which present specific challenges for present-day Vision-Language Models (VLM). In the study, issues related to entity disambiguation, visual understanding, and table structure comprehension are raised to illustrate the kind of improvements required in AI models to better handle complex multimodal reasoning tasks.

3.2 Tables as Texts or Images: Evaluating the Table Reasoning Ability of LLMs and MLLMs

(Deng et al., 2024)

This paper investigates the performance of large language models on table-related tasks, emphasizing text and visual representations. With some discussion on comparison, it concludes that inputting images combined with appropriate prompting brings about superior outcomes. The exploration proves the efficacy of such models in tasks like table question answering and table fact-checking.

Nevertheless, there is still the challenge of understanding the multimodal data in a combined and reasoned fashion by tying the visual and textual elements in tables. Most current models do not establish links from visual components, like charts, symbols, etc., to their corresponding tabular data, leading to incomplete and fragmented interpretations. Additionally, it has been observed that errors are introduced in the text due to the OCR-based extraction methods, affecting the accuracy and usability of data. This project is a follow-up to some earlier works and will focus mainly on developing a more robust multimodal framework to improve spatial and semantic reasoning for integrating modalities, develop AI-derived validation to improve OCR correction, and refine the data acquisition methods so as to ensure greater precision in data representation and retrieval.

3.3 Multimodal Knowledge Extraction and Accumulation for Knowledge-based Visual Question Answering

(Ding et al., 2022)

This study introduces a framework for enhancing visual question answering (VQA) by integrating multimodal knowledge. Unlike text-only approaches, it constructs explicit triplets linking visual objects to factual answers through implicit relations, capturing complex visual-textual associations. The framework uses three learning objectives—embedding structure, topological relations, and semantic space—to model these triplets. A two-phase training strategy, combining pre-training on diverse datasets and fine-tuning on domain-specific data, enables the accumulation of general and specialized knowledge. This approach achieves significant improvements, outperforming state-of-the-art models by 3.35% on OK-VQA and 6.08% on KRVQA, demonstrating its effectiveness in multimodal knowledge integration.

3.4 An Efficient Approach to Informative Feature Extraction from Multimodal Data

(Wang et al., 2019)

This paper introduces Soft-HGR, a framework for extracting informative features from multimodal data. Unlike traditional methods relying on Hirschfeld-Gebelein-Rényi (HGR) maximal correlation, which imposes strict whitening constraints, Soft-HGR removes these limitations while main-

taining feature geometry. Its objective function, based on two inner products, ensures efficient and stable optimization. The framework extends to multiple modalities, handles missing data, and incorporates semi-supervised learning for partially labeled datasets, enhancing feature discriminability. Empirical results show that Soft-HGR learns more informative feature mappings and achieves superior optimization efficiency compared to existing approaches, making it a practical solution for multimodal feature extraction.

However, these approaches present several challenges. Building and maintaining a robust multimodal knowledge base demands significant resources, including extensive data collection and annotation efforts. Aligning and integrating information across diverse modalities can introduce complexity, potentially resulting in inconsistencies or ambiguities in the knowledge representation. Furthermore, the reliance on pre-training and fine-tuning strategies requires substantial computational resources, which may hinder the framework's accessibility and scalability in real-world applications. These limitations highlight the need for efficient methods to address resource constraints and ensure seamless integration of multimodal knowledge.

3.5 Positioning Our Work

Our project builds upon these foundations while addressing several key limitations identified in prior research. Unlike approaches that focus solely on reasoning over existing tables or extracting features from multimodal data, we develop an end-to-end pipeline for detecting, extracting, and structuring multimodal tables from web sources. We extend beyond traditional HTML parsing by incorporating visual understanding through screenshot analysis, enabling extraction from complex layouts and image-based tables that confound conventional scrapers.

While previous studies have demonstrated the effectiveness of multimodal models for table question answering, our work focuses on the critical preprocessing step: accurate extraction and structuring of multimodal tables that can then serve as input for downstream reasoning tasks. By developing a specialized pipeline for this purpose, we enable applications in domains where tables must first be extracted from their source documents before analysis can occur.

Furthermore, our approach addresses the practical challenges of deployment and scalability by integrating open-source models rather than relying on resource-intensive pre-training and fine-tuning. This makes our system accessible to a broader range of users and applications, particularly those with limited computational resources. The integration of natural language querying capabilities directly into our extraction pipeline also represents a novel contribution, allowing users to extract insights from multimodal tables without requiring separate reasoning systems.

4 Dataset and Methodology

4.1 Dataset Used

Traditional machine learning projects use static datasets; our system operates on dynamic web data, collected in real-time. Our system ingests HTML tables and visualized table structures off websites, thus extracting data from disparate sources without relying on static datasets. This results in the system gaining flexibility and adaptability regarding the plurality of table formats found in the wild.

Our system supports two major types of web sources: web sources supporting general webpages with HTML tables, CSS Grid layouts, DIV tables, or list-based tabular structures. Then it offers special processing for Amazon product pages, which often contain their kinds of comparison tables and specification tables with a unique layout and content structure.

The raw HTML and visual data are subjected to several preprocessing steps crucial to extraction. Cookie consent banners are handled automatically, and popups get dismissed, so the page renders cleanly; bounding boxes are then established for potential table structures by identifying element positions. High-quality screenshots are recorded for visual processing, while the text undergoes normalization in accordance with standardized whitespace handling. We also extract images embedded inside tables and save them locally to retain the multimodality of the information.

Although our system does not implement traditional dataset augmentation for model training, it performs several on-the-fly data enhancements. These include converting relative image URLs to absolute URLs to ensure proper resource loading, transforming data between various formats (CSV, JSON, and HTML) according to user preferences, and enriching the extracted data with metadata such as table titles and source URLs for improved con-

textual understanding.

4.2 Methodology and Approach

Our methodology integrates advanced web scraping techniques, computer vision, and artificial intelligence to extract structured tabular data from diverse web sources. This comprehensive approach enables the system to handle the variability and complexity of real-world table structures found across the web.

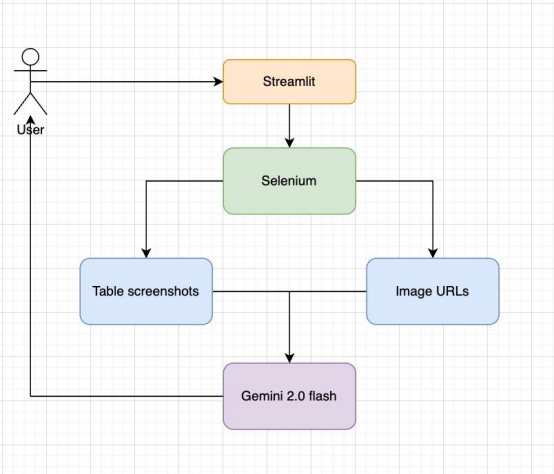

Figure 1: System pipeline of the Web Table Extractor. The user interacts with a Streamlit interface, which uses Selenium to fetch tables and images, forwarding them to Gemini 2.0 Flash for processing.

4.2.1 System Architecture

The system is designed as a multi-stage pipelined system to cater for flexibility, and robustness. On the front end is an obvious user interaction layer built upon Streamlit to present users with controls for URL input, format selection (CSV, JSON, HTML), and triggering an extraction. The interface also tracks extraction progress, previews the tables with an option to download, and provides an AI-assisted question-answering interface that interacts with the extracted data through natural language queries.

Behind the user interface, for rendering websites and performing extraction, Selenium WebDriver is used-configured with a headless Chrome browser for efficient rendering. It handles dynamic webpages with timeout, and automatically overrides cookie consent banners and pop-up dismissals. It captures screenshots in high quality: screenshots of the tables and of full pages while parsing the HTML elements for structural identification.

Table detection and classification utilize two complementary strategies for multi-format table identification. HTML standard tables are detected by tag-based selectors, whereas ARIA role tables are identified in accessibility-compliant pages. For the latest modern web designs, table structures through CSS Grid or DIV are recognised, as are list-based (UL/LI) table representations. The system analyses the row-column structure by similarity detection and removes duplicate tables through spatial overlap analysis.

On the other hand, our specialized data extraction engine executes fully customized extract procedures per table type: HTML tables extracting with a full treatment of colspan and rowspan, and ARIA tables according to their semantic structure. CSS Grid-based tables allow for analyses of position-based cell grouping; meanwhile, list-based structures utilize pattern-recognition techniques. For Amazon product pages, the system uses dedicated extractors for comparison tables and for the conversion of bullet point specifications into structured tabular data.

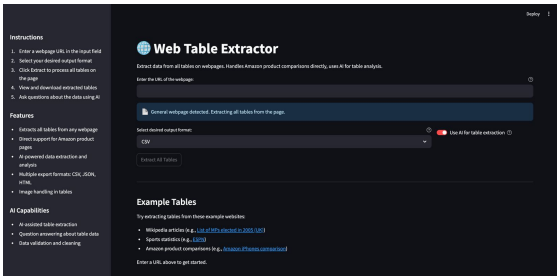

Figure 2: Figure 2: User interface of the Web Table Extractor. Users input the webpage URL and select the export format.

4.2.2 Image Processing and AI Enhancement

The image processing pipeline forms a critical component of the multimodal extraction system. It detects image tags within table cells and normalizes URLs that may be relative, protocol-relative, or absolute paths. Using ThreadPoolExecutor for efficiency, the system downloads images in parallel, stores them locally using unique IDs, and generates corresponding HTML image tags for visualization in the extracted output.

To allow for extra extraction beyond traditional parsing means, the Gemini 1.5 Flash AI model from Google is utilized for assistance. This AI help becomes especially useful when HTML parsing fails, basically extracting tables straight from

screenshots. It will then facilitate format conversion between CSV, JSON, and HTML, and also support the question-answering system whereby users can query table content in natural language.

The next and last stage is the data transformation layer, wherein the structured output in many formats is generated. This generation also includes creating standardized Pandas DataFrames while handling difficult cases such as missing or inconsistent column headers. It then exports the data into the user's desired format (CSV, JSON, or HTML) and can ZIP the output when multiple tables are extracted from the same source.

4.2.3 Specialized Extraction Techniques

This system introduces a variety of novel techniques for facing web table extraction challenges. In the identification of table titles, a multi-strategy approach has been created, analyzing caption tags, proximity-based heading associations, parent container elements, and ID and class names-apart from which camelCase and snake_case identifiers are translated to readable titles.

Modern web design frequently employs CSS Grid layout for the tabular presentation of data. The robust grid detection system evaluates the computed styles via JavaScript to detect grid-template-rows and grid-template-columns, makes guesses on the row structure based on visual positioning, and groups children into logical rows even without an explicit grid marker.

Enhanced screenshot techniques are employed in the visual table capture over the commonly used element capture. The system scrolls elements into the viewport so that it is properly positioned, fixes overflow settings to keep the full table visible, applies bottom margins to ensure that the table content is fully captured, and employs fallback strategies for layouts that prove too tricky for standard methods.

For e-commerce applications, we developed Amazon-specific processing capabilities that identify product comparison tables and extract specification tables with accurate title associations. The system converts detailed bullet points into structured data and handles product images with high-resolution source detection to maintain visual fidelity.

4.2.4 Implementation and Optimization

This system is implemented with a stack of complementary technologies centered on Python 3.x. Streamlit is used for the web app interface; Selenium, for web rendering and interaction; and BeautifulSoup for HTML parsing. Pandas takes care of data manipulation and conversion between formats, while Pillow (PIL) is responsible for image processing. Finally, the Google Generative AI SDK was integrated to enable AI-assisted extraction through the Gemini model.

Our implementation tries to leverage a set of key design patterns aiming at robustness and maintainability. Extractors run isolated within a headless browser to avoid failures corrupting the application. Download tasks are run concurrently with ThreadPoolExecutor for higher efficiency. The system is error-resilient and degrades gracefully, allowing it to carry on working if some parts fail. In the user interface, session state management is maintained for continuity with incremental progress tracking to keep users informed, especially for lengthy operations.

By implementing various efficiency improvements, the performance expects to be optimized. Using ThreadPoolExecutor parallelizes image downloads, and with such concurrency, may greatly reduce image-heavy-table rendering wait times. Early filtering checks for element visibility and size so that unnecessary processing of non-table elements does not take place. Memory management is performed on-demand for resource loading and processing, while screenshot caching rejects redundant rendering operations. The UI progressively loads with incremental updates during the extraction, while the entire system runs atop a highly optimized custom-built configuration of Chrome for headless operation.

This all-encompassing approach surely marks the promise of a leap forward in web table extraction, going all the way to account for the entire gamut of tabular representations found on the modern web, combining classical web scraping techniques with contemporary AI assistance and sturdy visual-processing methods.

5 Experiments

5.1 Experimental Setup

We evaluated our Selenium + LLM pipeline on a diverse set of real-world websites containing complex tables. The pipeline was executed on a macOS environment using Chrome with ChromeDriver. The implementation was based in Python and included tools such as Selenium for automated web browsing and DOM extraction, Gemini 2.0 (via API) for

vision-language inference, and Streamlit for the user interface.

The experimental process followed these steps: users provided a target URL via the Streamlit interface; Selenium navigated to the webpage, captured both the HTML DOM and screenshots of the relevant table regions, and extracted any embedded image URLs. These multimodal components were passed to the Gemini 2.0 model, which returned structured representations in HTML, CSV, and JSON formats.

We conducted evaluations on a variety of sites, including:

- Amazon product listing and comparison pages with mixed content (text, images, and pricing).
- Wikipedia articles containing complex tables with merged headers and footnotes.
- Data-centric blog and e-commerce sites presenting promotional and tabular product data.
- HTML-rendered scientific papers from arXiv featuring annotated or multi-span tables.

Each test case involved executing the full pipeline and saving the structured outputs for analysis and comparison.

5.2 Evaluation Metrics

To assess the performance and reliability of our extraction pipeline, we adopted both automated and manual evaluation metrics. These metrics focused on structural accuracy, visual parsing, and semantic preservation:

- **Cell-level Accuracy:** We manually compared the extracted text from each table cell against a ground truth to quantify parsing precision.
- **Row/Column Alignment:** The structural integrity of the output tables was checked for alignment with the original table layout, accounting for merged headers, multi-span cells, and nested rows.
- **Image Link Accuracy:** We verified whether the pipeline accurately identified and preserved embedded image URLs in the extracted tables.
- **Semantic Consistency:** Human evaluators rated the extracted outputs on a 1–5 scale, reflecting how well the final representation retained the meaning and relationships of the original table.
- **Runtime Performance:** The time taken from URL input to output generation was recorded to evaluate the pipeline's efficiency for practical applications.

These metrics collectively allowed us to holistically evaluate the robustness, usability, and real-world applicability of our multimodal table extraction system.

6 Results & Analysis

6.1 Amazon Product Comparison

The pipeline accurately extracted product comparison tables from Amazon, including image URLs, textual details (price, ratings, delivery), and formatting like multi-line cells. The outputs in HTML, CSV, and JSON formats retained high semantic fidelity. Manual inspection showed:

- **Cell-level Accuracy:** Above 90% for textual data.
- **Row/Column Alignment:** Preserved merged headers and columns.
- **Image Link Accuracy:** Extracted all product thumbnails successfully.
- **Semantic Consistency Score:** 5/5
- **Runtime:** ~8 seconds per table

Column 1	Column 2	Column 3	Column 4	Column 5	Column 6
	Try again!Details Added to CartqsCSRF_TreatmentAdd to cart	Try again!Details Added to CartqsCSRF_TreatmentAdd to cart	Try again!Details Added to CartqsCSRF_TreatmentAdd to cart	Try again!Details Added to CartqsCSRF_TreatmentAdd to cart	Try again!Details Added to CartqsCSRF_TreatmentAdd to cart
Price	-28$51,298.00$1,298.00List $1,799.99	$849.00$849.00	-19$2,198.00$2,198.00List $2,699.99	$1,709.95$1,709.95	$2,016.95$2,004.95
Delivery	—	Get it as soon asTuesday, May 6	Get it as soon asThursday, May 8	—	—
Customer Ratings	4.64.6 out of 5 stars2,233	4.54.5 out of 5 stars18	4.74.7 out of 5 stars967	4.74.7 out of 5 stars21	4.94.9 out of 5 stars14
Sold By	Cardinal Camera and Video Center	Minty Gadgets (we track serial numbers)	Amazon.com	6ave	6ave
display type	LCD	LCD	LCD	LCD	LCD
display size	3 inches	3 inches	3 inches	3 inches	3 inches
lens type	Zoom	Sony E-mount	Wide Angle	—	Zoom
zoom type	optical	—	—	—	Optical Zoom
shooting modes	AUTO (iAuto), Programmed AE (P), Aperture priority (A), Shutter-speed priority (S), Manual (M), Movie modes	Movie	Auto, Programmed, Aperture, Shutter speed, Manual, Movie	Automatic, Movie	Automatic, Movie, Panorama, Portrait
connectivity tech	USB	HDMI	HDMI, USB	Wi-Fi, USB, Micro HDMI, Micro USB, NFC	Wi-Fi, USB, Micro USB, NFC
video resolution	4K UHD 2160p	1080p	4320p	1080p	1080p
optical zoom	1 multiplier x	—	8 multiplier x	—	—
magnification	0.78x	—	0.78x	0.78x	—
wireless tech	Bluetooth	Wi-Fi	Bluetooth, Wi-Fi	Wi-Fi, NFC	Wi-Fi, NFC
model name	Sony a7 III	Alpha a7 II	Sony Alpha 7 IV	ILCE7M3/B	ILCE7M3K/B

Figure 3: Amazon product comparison: extracted table (HTML view)

6.2 Wikipedia Stadium Table

Wikipedia's stadium table with embedded images and hyperlinks was also successfully extracted. Although minor formatting errors appeared in footnote handling, key data (names, locations, capacities) were preserved.

- **Cell-level Accuracy:** ~85%
- **Row/Column Alignment:** Mostly consistent

- **Image Link Accuracy:** Partially successful (cropped image links)
- **Semantic Consistency Score:** 4/5
- **Runtime:** ~6 seconds

Figure 4: Wikipedia stadium table (CSV output)

6.3 NeurIPS Poster PDF (Image Only)

This case served as a negative test: the NeurIPS 2023 poster URL contained no HTML table elements and was a static PNG image. As expected, the system returned a message stating no tables were found. No structured output was produced, demonstrating proper error handling in the pipeline.

- **Cell-level Accuracy:** Not applicable
- **Image Link Accuracy:** Not applicable
- **Semantic Consistency Score:** 0/5 (no table to extract)

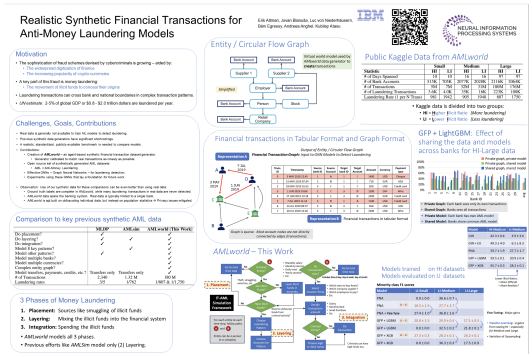

Figure 5: NeurIPS poster (no extractable table detected)

6.4 arXiv Scientific Paper (HTML Form)

The pipeline detected multiple tables in the HTML-rendered version of arXiv paper 2504.19878v1. Complex math symbols and multi-row headers posed partial challenges. While structure was maintained, formulaic expressions were inconsistently parsed.

- **Cell-level Accuracy:** ~75%
- **Row/Column Alignment:** Mixed (issues with nested headers)
- **Image Link Accuracy:** Not applicable
- **Semantic Consistency Score:** 3.5/5
- **Runtime:** ~9 seconds

	Chiplet area relative to a Mesh topology		
Topology	37mm^2	74mm^2	148mm^2
Mesh	0.00 ± 0 %	0.00 ± 0 %	0.00 ± 0 %
FoldedTorus	0.00 ± 0 %	0.00 ± 0 %	0.00 ± 0 %
HexaMesh	4.34 ± 0 %	2.27 ± 0 %	1.16 ± 0 %
FoldedHexaTorus	4.34 ± 0 %	2.27 ± 0 %	1.16 ± 0 %
OctaMesh	8.69 ± 0 %	4.54 ± 0 %	2.32 ± 0 %
FoldedOctaTorus	8.69 ± 0 %	4.54 ± 0 %	2.32 ± 0 %

Figure 6: Table extracted from arXiv HTML paper

7 Expected Contributions & Impact

7.1 Contributions

This project will introduce a novel multimodal extraction pipeline that effectively integrates object detection, OCR, and AI-driven semantic linking to accurately extract and structure tabular data from diverse web sources. By leveraging state-of-the-art models like YOLOv8, DePlot, and GPT-4V, it will establish a robust framework for preserving relationships between numerical and visual elements in complex documents.

7.2 Impact

The system will significantly reduce manual effort in extracting and processing multimodal data, benefiting industries like finance, academia, and market intelligence. By providing an automated, scalable, and context-aware extraction solution, it will enhance decision-making, streamline research workflows, and enable more effective data utilization across various domains.

8 Shortcomings and Future Work

While our multimodal extraction system accomplishes further improvement compared to existing mechanisms, there are several limitations that still await curing. Though Selenium web scraping is considered best from the perspective of technicality, it faces challenges with anti-scraping measures, heavy JavaScript, and CAPTCHA implementations from sites; thus, there arises the problem of incomplete extraction of data. The system is unable to deal with infinite-scroll pages and those requiring user credentials, thereby restricting the ability to access some content types. Furthermore, although we

are able to detect and extract charts and graphs, the actual replication of some of these visuals becomes a challenge to our system, especially when it comes to reproducing data points with absolute precision for multi-level charts or irregular formatting.

Future work will include implementing an adaptive scraping methodology using headless browsers with rotating proxies to circumvent restrictions and designing specialized models for extracting chart data points so as to guarantee numeric accuracy. Integrating more robust OCR error correction systems using contextual language models is part of our plan, including an examination of lightweight alternatives that might reduce computational overhead. Providing support extension for interactive visualization formats and a real-time collaborative extraction feature would further increase the utility of the system. Increased research in domain-specific fine-tuning of extraction models, particularly those specialized in fields of scientific publications and financial reports, will further enhance extraction precision. We also plan to provide an active learning setup wherein the user can give feedback for continuous improvement of the model, thereby ensuring the system adapts to new web design patterns and data presentation formats.

References

Naihao Deng, Zhenjie Sun, Ruiqi He, Aman Sikka, Yulong Chen, Lin Ma, Yue Zhang, and Rada Mihalcea. 2024. Tables as texts or images: Evaluating the table reasoning ability of llms and mllms.

Yang Ding, Jing Yu, Bang Liu, Yue Hu, Mingxin Cui, and Qi Wu. 2022. Mukea: Multimodal knowledge extraction and accumulation for knowledge-based visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5089–5098. IEEE.

Suyash Vardhan Mathur, Jainit Bafna, Kunal Kartik, Harshita Khandelwal, Manish Shrivastava, Vivek Gupta, Mohit Bansal, and Dan Roth. 2024. Knowledge-aware reasoning over multimodal semi-structured tables.

Lichen Wang, Jiaxiang Wu, Shao-Lun Huang, Lizhong Zheng, Xiangxiang Xu, Lin Zhang, and Junzhou Huang. 2019. An efficient approach to informative feature extraction from multimodal data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):5281–5288.

Appendix

1. Google Drive Link: **NLP-Tokenizers-G Drive**
2. GitHub Repository Link : **NLP-Tokenizers-GitHub**
3. Working Document: **Tokenizers-Working-document**