ODLP: Scalable Multimodal Information Retrieval Using Visual and Structural Integration

Anonymous Author(s)

Abstract

The effective application of Large Language Models (LLMs) in domain-specific settings (e.g., e-commerce, finance, science) hinges on their ability to access and reason over reliable structured knowledge. However, extracting structured knowledge from e-commerce webpages presents significant challenges. First, webpage content is typically represented by HTML and CSS, which, when directly inputted into LLMs, frequently exceeds token limitations. Second, effectively transforming unstructured web data into structured information, such as extracting detailed lists of product descriptions, remains problematic and necessitates sophisticated parsing techniques. This paper investigates the integration of unstructured knowledge, exemplified by HTML, with other modalities (like visual representation) to derive structured knowledge tailored for e-commerce domain tasks. This method enables accurate, context-aware extraction and alignment of item information (e.g., product attributes, pricing), overcoming the limitations of methods reliant solely on unstructured text or inconsistent tags. Experimental evaluations across 31 shopping websites with more than 1,200 products validate the effectiveness of this structured knowledge integration, achieving 96% recall/precision and demonstrating robustness. ODLP significantly outperforms LLM-based tools like ZeroX, showcasing the power of combining multimodal data with LLM reasoning for domain-specific problems. This work provides a reliable method for processing extensive amount of unstructured web information. Dataset for Testing is available on Github [1].

Keywords

Web Information Extraction, Optical Character Recognition, Document Object Model, Large Language Models, Multimodal Integration

ACM Reference Format:

Anonymous Author(s). 2025. ODLP: Scalable Multimodal Information Retrieval Using Visual and Structural Integration. In *Proceedings of Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '25)*. ACM, New York, NY, USA, 9 pages. https://doi.org/XXXXXX X.XXXXXXX

[1]Github link is pending

KDD '25, Toronto, ON, Canada

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

The rapid growth of online data has made Web Information Extraction (WIE) a cornerstone for fueling domain-specific applications across e-commerce, finance, science, and industry. A common requirement in these domains, particularly e-commerce, involves extracting unstructured knowledge about a large amount of items, such as lists of products, search results, or financial tickers, from a single page. Compounding this challenge is the need to generalize this extraction process across different websites, each with unique layouts and coding practices. The ability to accurately capture key structured information—like product attributes, pricing details, scientific data points, or company financials—from these diverse sources is critical for downstream tasks, including price comparison, inventory management, market analysis, and crucially, for grounding Large LLMs with factual, real-time data[10].

Traditional approaches to WIE often rely on rule-based methods or simplistic parsing techniques. These are frequently brittle, struggling to adapt to the heterogeneity of web designs and, crucially, often failing to reliably interpret the semantic meaning of information embedded within diverse HTML tags (Challenge 1). Consequently, they struggle to consistently extract accurate structured knowledge. Recent advancements in LLMs [11] offer promising alternatives due to their powerful contextual understanding. Their potential to adapt dynamically has been shown in related complex environments, such as enhancing virtual healthcare interfaces [3] or dynamically modifying web structures for accessibility [8]. These examples highlight LLMs' capability to handle dynamic contexts, inspiring their exploration for WIE where structural and semantic complexities hinder traditional methods. However, when applied directly to WIE for domain-specific applications, standalone LLMs face their own significant limitations. They may struggle to inherently grasp the complex structural relationships and dependencies between different tags and elements on a webpage (Challenge 2), which is vital for assembling coherent structured knowledge records (e.g., linking a product name correctly to its price). Furthermore, LLMs can be prone to hallucination or misinterpretation, raising concerns about the factual accuracy and reliability of the extracted information (Challenge 3), a critical issue when this data feeds decision-making or other AI systems in sensitive domains like price analysis in e-commerce domain. Thus, while promising, LLMs require integration with methods that can explicitly handle structure and validate information to reliably extract structured knowledge from the web.

This paper presents a multimodal method that addresses the inherent challenges of WIE by integrating visual representation captured from webpages (e.g., screenshots) with the non-structural data embedded within HTML content, utilizing LLMs for interpretation and extraction. Optical Character Recognition (OCR) is employed to extract aligned text from visual representations. The DOM (Document Objective Model) tree parsing approach enables

the methodology to split HTML into sections that contain information for same item. Finally, LLMs are utilized to extract data from HTML sections, overcome its token limitations.

Challenges 1 and 3 are addressed by leveraging the LLM's powerful semantic interpretation, informed by the visual representation of webpages (via OCR), which resolves ambiguities in tag meanings and enhances information accuracy. Meanwhile, Challenge 2 is tackled through the analysis structure of HTML, providing the necessary framework to comprehend complex inter-tag relationships. This adaptability is critical for applications where traditional methods often fail due to variations in webpage layouts. To validate the effectiveness of this approach, the system was tested on a diverse dataset of 31 shopping websites, encompassing over 1,200 products from various domains, including e-commerce, groceries, and electronics. Comparative analysis with existing tools, such as the ZeroX[2] multimodal LLM, highlights the proposed system's high performance, achieving a 100% precision, recall, and F1 score. These results demonstrate the methodology's scalability, accuracy, and versatility across dynamic and heterogeneous web environments.

This paper is organized as follows. Section 2 reviews the related work, categorizing existing approaches for WIE based on their methodologies, including visual representation processing, DOM tree parsing, and text-based approaches while summarizing their respective limitations. Section 3 details the proposed methodology, explaining how OCR, DOM tree parsing, and LLMs are integrated to achieve a robust and adaptable solution for WIE. Section 4 combines the experimental setup and discussion, presenting the dataset, evaluation metrics, and comparative results while highlighting the strengths and weaknesses of the proposed system relative to existing tools. Finally, Section 5 concludes with insights into the limitations of the current approach and directions for future research.

2 Related Works

WIE is a critical area of natural language processing (NLP) that involves transforming unstructured data into structured formats by identifying predefined classes of entities, relationships, and events[13]. Within this domain, advancements have been made in different approaches, including visual representation-based methods, DOM tree parsing and text processing. Each approach offers unique contributions while facing distinct limitations, which this section categorizes and discusses in detail.

2.1 Web Data Extraction Based on Visual Representation

Visual representation-based approaches treat web pages as images, leveraging visual cues to locate and extract relevant data. The Visual representation-Based Segmentation (VIBS) algorithm by Wang et al. [18] builds on earlier methods by integrating convolutional neural networks (CNN) to enhance segmentation efficiency and accuracy. By employing heuristic rules to filter irrelevant tags, VIBS improves data extraction from visually complex web pages. Similarly, vision-based monitoring has been explored in construction management, where a novel method integrating deep learning object detection and image captioning achieved a Consensus-based Image Description Evaluation (CIDEr) score of 1.84, transforming construction images into graph-formatted semantic information to assist decision-making for managers. However, VIBS struggles with generalizability across diverse web page designs, where structural inconsistencies can limit its effectiveness. Li et al. [7] introduced WIERT, a web information extraction method utilizing a render tree-based framework to parse webpages. WIERT enhances data extraction by leveraging visual and structural information, but challenges remain in balancing scalability and efficiency for large-scale applications. Similarly, Kumar et al. [5] introduced the Context-aware Visual Attention (CoVA) pipeline, integrating syntactic structure with visual features through graph attention networks (GATs). Despite achieving superior cross-domain accuracy, especially in price extraction, CoVA requires extensive annotation efforts and computational resources.

2.2 Web Data Extraction Based on DOM Tree Parsing

DOM tree parsing methods focus on the structural hierarchy of HTML elements to locate data records. AutoRM by Shi et al.[15] identifies candidate records by distinguishing boundary markers in the DOM tree, effectively handling nested data and achieving high precision. UzunExt[17] optimizes DOM-based methods by leveraging metadata to predict target tag positions, significantly reducing processing time. Zhou et al.[21] further refined this approach with SimpDOM, a model that simplifies DOM trees to extract transferable contextual features without relying on visual features. By incorporating innovations like friend circle features and semantic embedding, SimpDOM achieved a 1.44% improvement in F1 scores compared to previous methods and demonstrated significant cross-domain adaptability. However, DOM tree-based approaches still face challenges in adapting to diverse web structures and extracting target information effectively.

2.3 Web Data Extraction Based on Text Processing

Text processing methods focus on analyzing raw textual content to extract structured information. The Unified Structure Generation for Universal Information Extraction (UIE) framework by Lu et al.[9] uses schema-based prompts to adapt pre-trained models for various extraction tasks, offering state-of-the-art performance. However, the reliance on pre-training limits adaptability to niche or evolving domains. In contrast, the proposed method leverages advanced language models like GPT-3[1], GPT-3.5/4[12], BART[6], and DeBERTaV3[4] to achieve high recall and near-perfect precision while remaining adaptable to future models. Combining automation with human oversight, it ensures high-quality data extraction and demonstrates scalability across domains, as evidenced by the creation of a critical cooling rate database for metallic glasses, significantly reducing human effort compared to traditional methods[14]. This aligns with the broader trend identified by Zhang et al.[20] of utilizing LLMs for structuring knowledge from text. Similarly, rule-based approaches, such as the suffix-matching algorithms for named entity recognition (NER) proposed by Wu et al. [19], are

[2]ZeroX GitHub repository: https://github.com/getomni-ai/zerox

effective for specialized tasks but lack the scalability and flexibility of modern data-driven methods. Beyond direct extraction, LLM reasoning is also being applied to discover complex patterns; Chen et al. [2] introduced ReStruct to find and explain semantically coherent meta-structures within Heterogeneous Information Networks (HINs) using an LLM-integrated evolutionary approach.

3 Methodology

To address these limitations, this research proposes a novel, integrated methodology combining OCR, DOM tree parsing, and LLMs. This approach synergizes the strengths of visual and structural data processing, enabling robust, scalable, and accurate information extraction across diverse web environments. By capturing both visual and contextual elements, the proposed system overcomes the constraints of generalizability, efficiency, and adaptability observed in existing approaches.

OCR is utilized to extract textual information from visual representations of web pages. OCR is a reliable and stable technology for converting images of text into machine-readable text[16]. It is preferred over visual models of LLMs because it consistently captures textual data with high accuracy. By taking full-page screenshots, OCR ensures all visible information, including product names, prices, and descriptions, is captured for further processing. The DOM tree, which represents the hierarchical structure of a webpage containing all HTML elements, is parsed to map extracted text to its corresponding HTML elements. This step is crucial for understanding the structural relationships within the webpage and locating the tags that contain relevant product information. Large language models are employed to classify and validate key elements identified in the OCR stage. LLM determines which extracted values represent key elements, such as product names and prices. By leveraging LLM's contextual understanding capabilities, this approach ensures greater accuracy than simple keyword matching.

3.1 Extraction Process Illustration

The information extraction process begins (step 1) by initializing the system. First, (step 2) a complete screenshot of the target webpage is captured, encompassing all visible textual elements like product names and prices. (step 3) OCR is then conducted on this screenshot, converting the visual representation into machine-readable text. (step 4) All characters from webpage screenshot are sent to a LLM for extracting key element_1 or key element_2 (e.g. product name, product price). (step 5) The LLM analyzes this bulk text to identify and parse initial key information, such as product prices (referred to as key element_1).

The core challenge addressed here is partitioning the entire HTML document into individual containers, each containing all information relevant to a single product. To accomplish this, the approach identifies a specific node within the DOM whose immediate child nodes represent these desired containers. Utilizing OCR (step 3), the process extracts elements visible on the webpage and selects a particular type (e.g., price).Using the identified key element values (e.g., specific prices). Three instances of this element type are then identified, and their positions within the DOM are determined via string matching. By comparing these DOM paths, the

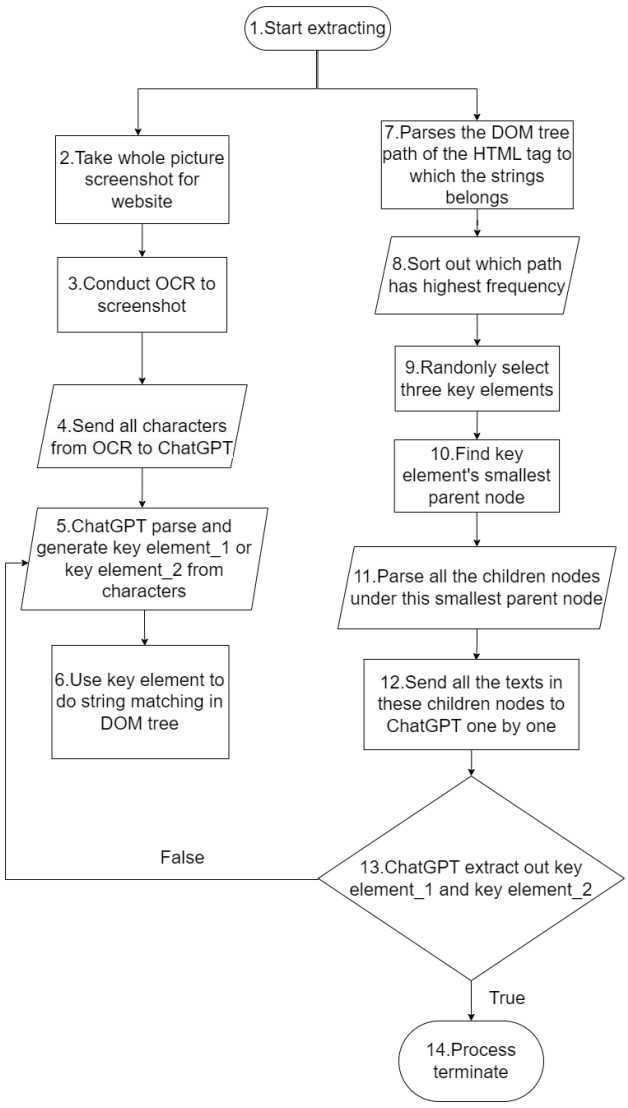

Figure 1: ODLP's extraction process

system locates a common section, which corresponds to the least common parent node (e.g. Stage E in figure 2) encompassing all target elements. Upon identifying this node, all child nodes beneath it are parsed, and their content is sequentially submitted to the LLM for finer-grained analysis. For verification and robustness, multiple instances (e.g., three randomly selected key elements like prices) found via this common path are considered. The system identifies the least common parent node in the DOM tree that contains all these verified instances, effectively isolating the relevant data block. Subsequently, all child nodes under this smallest parent node are parsed to access detailed information. (step 6-12)

The content (e.g., HTML code or text) of these child nodes is sent sequentially to the LLM for finer-grained analysis. (step 13) The LLM processes the content of each child node to extract the secondary

required information, such as product names (key element_2), associating them with the previously identified key element_1 (price). A fallback and retry mechanism exists: if the process up to finding the common parent node fails (e.g., indicating missing price information for structuring the output) or if the LLM fails to extract the required elements in this step (False condition), the system may reinitiate prompting or loop back to Step 6 to retry the matching and extraction process, potentially focusing on key element_2 first. If the extraction is successful (True), (step 14) the process terminates, and the final extracted information, including associated prices and product names, is structured and stored for downstream use.

3.2 Walk Through Example

To further illustrate the ODLP extraction process, we use an example from Metro.ca in Figure 2, where key element_1 is the product's price and key element_2 is the product's name. It is important to note that while this example is simplified for demonstration purposes, real-world webpages often feature diverse and complex nested structures. ODLP is designed to handle such scenarios effectively. A key capability of our method is its ability to decompose complex HTML code into smaller, related chunks. This allows the system to leverage an LLM to accurately extract information and discern the relationships between different data fields without being constrained by token input limitations. The process shown in Figure 2 begins with extracting visual representation from the webpage using OCR tools (Stage A). The extracted texts, such as "$0.29", "$0.79", and "$1.99" (along with corresponding product names), are sent to the LLM for further processing (Stage B). Even within complex layouts, the LLM, operating on the decomposed chunks, can accurately identify the relationships between data fields. It then outputs key element_1, such as the food prices, in a structured format (Stage C). For example, it generates a JSON output where prices are clearly labeled. This demonstrates how ODLP, through its decomposition strategy and LLM integration, can effectively handle intricate webpage structures to achieve precise data extraction.

Using the identified prices, the system performs string matching within the DOM tree to locate the corresponding HTML tags (Stage D). The DOM traversal begins by identifying the nodes containing the target price values, such as `<SPAN>$0.29</SPAN>` or `<SPAN>$1.99</SPAN>`. The system then traverses upwards through the DOM structure to find the least common parent node that contains all matched price instances (Stage E). This least common parent node acts as the container for all relevant product information.

Once the least common parent node is identified, all child nodes under this parent node are parsed (Stage F). These child nodes contain detailed product information, such as product names and prices, organized within nested structures. The HTML content of these child nodes is sequentially sent to the LLM (Stage G), which processes the data to extract key element_2, such as product names. The final output is structured as a JSON object containing paired product names and prices. For example:

```
"item": "Banana", "price": 0.29;
"item": "English Cucumber", "price": 1.99;
"item": "Bicolor Corn", "price": 0.79.
```

This structured process ensures that all relevant product information is accurately extracted by combining OCR-based text extraction, DOM structure analysis, and context-aware LLM processing.

4 Experiment

4.1 Experiment Setup

This experiment was designed to evaluate the effectiveness of web information extraction methods for ODLP. The setup involved processing a diverse dataset of 31 shopping websites encompassing over 1,200 products. Various techniques, including OCR, HTML parsing, and large language model text processing were employed to extract and analyze product information. Evaluation metrics such as precision, recall, and F1-score were utilized to assess the accuracy and reliability of the methods across different website structures. The experimental framework also incorporated comparative analysis for an open-source tool, highlighting the strengths and limitations of each approach in addressing the challenges of structured and unstructured data extraction from dynamic web environments.

To conduct this experiment, Puppeteer[3] was employed to capture screenshots from HTML files as part of the web information extraction process. Subsequently, Tesseract[4] was utilized to perform OCR on the screenshots, enabling the extraction of textual data for further analysis. String matching and parsing of child nodes under the least common parent node were performed using BeautifulSoup[5]. Furthermore, the GPT-4o-mini-2024-07-18 model[6] was applied to analyze the visual representation extracted through OCR and to process the corresponding HTML code, facilitating comprehensive data analysis.

For comparative analysis, we selected an open-source project named ZeroX, which uses a multimodal LLM for information extraction. ZeroX processes user-submitted PDF files by converting them into images and leveraging its multimodal capabilities to extract information. In this experiment, we also configured GPT-4o-mini-2024-07-18 to perform the same tasks for comparison with ZeroX's LLM.

4.2 Testing Dataset

The testing dataset comprised 31 shopping websites which downloaded from Chrome browser into MHTML format, if the page has a lazy loading element, we will manually pull the scroll bar to the bottom of the page and then download it. There are 1,243 products that categorized into diverse groups to comprehensively evaluate information extraction performance. All data includes product names and product prices have been extracted manually by us to ensure data accuracy and completeness. Table 1 lists the tested websites and their categories. These categories included e-commerce platforms (e.g., Amazon, Staples), art and collectibles platforms (e.g., Artfinder), books and media providers (e.g., Barnes & Noble, G2A), electronics retailers (e.g., Sony, Best Buy Canada), outfit stores (e.g., Gap, The North Face), food and grocery vendors

[3]Puppeteer documentation: https://pptr.dev/
[4]Tesseract GitHub repository: https://github.com/tesseract-ocr/tesseract
[5]BeautifulSoup documentation: https://www.crummy.com/software/BeautifulSoup/bs4/doc/
[6]GPT-4o-mini documentation: https://platform.openai.com/docs/models/gpt-4o-mini

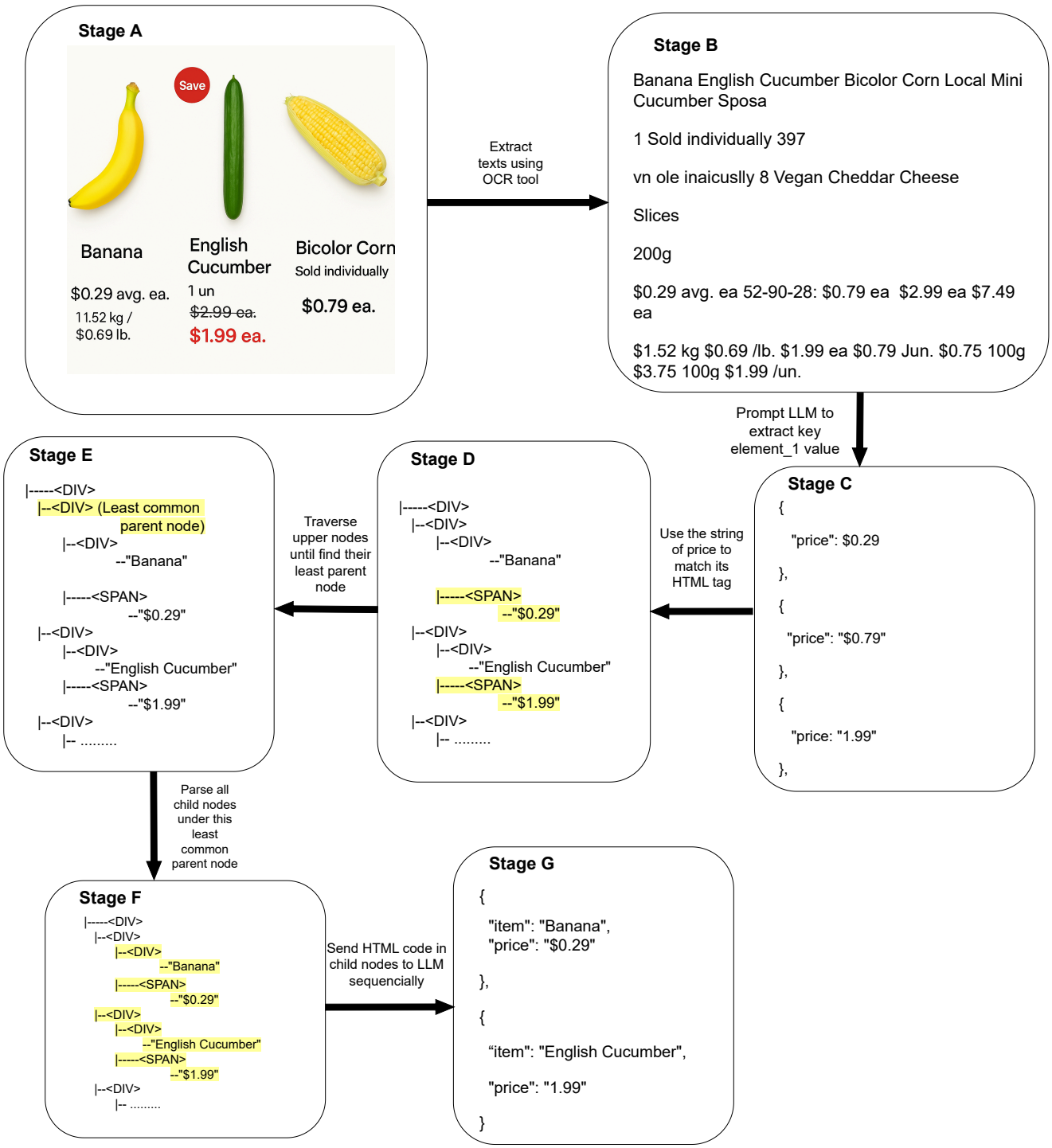

Figure 2: An example of extracting price information from a grocery store website

(e.g., Food Basics, Walmart), and second-hand goods platforms (e.g., OfferUp, Proxibid). The details are given in Table 1

Table 1: Categories and Corresponding Websites Used for Testing

Category	Websites
E-commerce	Amazon, Staples, Wayfair, Etsy, Canadian Tire
Art and Collectibles	Artfinder
Books and Media	Barnes & Noble, G2A
Electronics	Sony, Best BuyCanada, LG, Dell, HP
Outfits	Gap, The North Face, The Children's Place, Macy's, buybuy BABY, Disney Store
Food and Groceries	Food Basics, Swiss Colony, Bonanza, Nuts, Walmart, Raley's, Instacart, Metro, Costco, Wine
Second-hand goods	OfferUp, Proxibid

4.3 Evaluation Metrics

Both ODLP and ZeroX produced results in JavaScript Object Notation (JSON) format, presenting extracted product names and prices. Figure 3 illustrates a sample output. After retrieving product information and prices from 31 shopping websites, we manually verified the accuracy of the extracted data.

The evaluation of the extraction process focused on addressing two key challenges: 1) accurately associating related pieces of information, such as a product's name and its corresponding price, even when they are not structurally adjacent in the HTML, and 2) maintaining high accuracy even when processing webpages with a large volume of product information, which can often cause performance degradation (lower precision and recall) if the entire content is fed directly into an LLM without preprocessing. To quantify performance against these challenges, the results were categorized into True Positive (TP), False Positive (FP), and False Negative (FN):

- **TP:** The product name is correct, and its corresponding price is accurately matched.
- **FP:** The product name is incorrect, or it does not match the price.
- **FN:** The system failed to extract the product name and its price from the webpage.

Table 2: examples of entries classified as TP, FP, and FN.

Classification	Example
TP	{ "item": "Banana", "price": 0.29 }
FP	Price did not match with item: { "item": "Banana", "price": 0.79 } Item name was not correct: { "item": "Apple", "price": 0.29 }
FN	Missed in JSON file: { "item": "Banana", "price": 0.29 }

Table 3: Average number of input tokens for different models

Model	Input tokens (average)
ODLP	121322
ZeroX	1763.16

Figures 3, 4, and 5 illustrate the comparative performance of ODLP and ZeroX across three key metrics: precision, recall, and F1-score. To evaluate how accurately each method extracts information when faced with increasing data density, we strategically designed the analysis around three product group sizes per webpage screenshot: 10, 15, and 20 items. This tiered approach allows for a systematic assessment of scalability. In terms of precision (Figure 3), ODLP consistently achieved a relatively high score of 0.960 across all group sizes, while ZeroX showed a decline as the item count increased, dropping from 0.778 for 10 items to 0.697 for 15 items and further down to 0.423 for 20 items. For recall (Figure 4), ODLP again maintained at 0.960, whereas ZeroX started strong with 0.955 for 10 items but decreased to 0.876 for 15 items and significantly dropped to 0.515 for 20 items. The F1-score comparison (Figure 5) mirrors these trends, as ODLP sustained a perfect score of 1.000, while ZeroX fell from 0.858 for 10 items to 0.777 for 15 items and reached its lowest point at 0.465 for 20 items. These results demonstrate ODLP's superior performance and robustness in handling larger datasets within the tested range. The selection of 10, 15, and 20 items was specifically informed by preliminary observations indicating a substantial degradation in ZeroX's precision and recall when processing webpages containing more than 20 items. Thus, this setup effectively highlights the scalability challenges ZeroX faces when dealing with increased data complexity. In contrast, ODLP is designed for high-density information extraction; further tests confirmed its capability to maintain perfect precision and recall even on webpages containing as many as 98 items, underscoring its significant advantage in scalability. ZeroX's declining performance might also relate to limitations like resizing webpage screenshots to 1024×1024, potentially preventing effective information extraction from larger or denser pages. Although our method has high precision rate and recall rate, it still has limitations that cause extraction process to fail. If key element_1 extracted by LLM has two tags in it (e.g. "$1.99", "$" and "1.99" are separated into two tags), the string matching will fail because it cannot match two tags

at the same time. The whole extraction process will fail in this case.

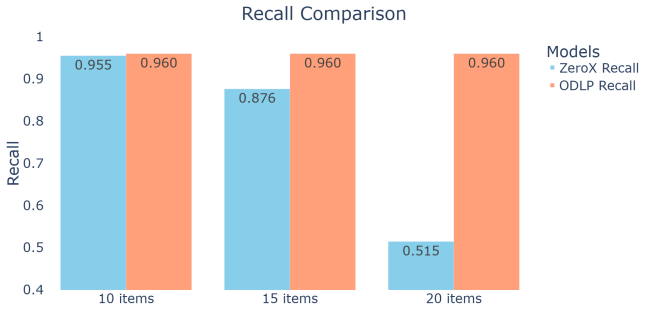

Figure 3: the recall rate comparison between ODLP and ZeroX

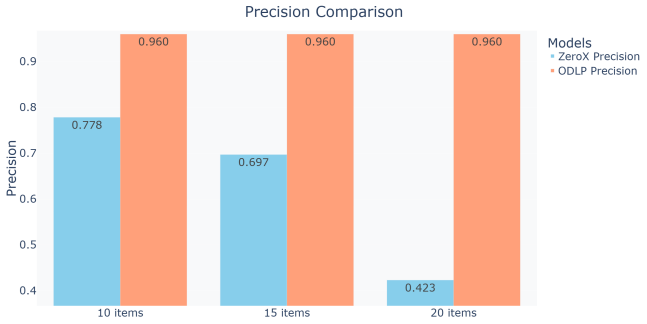

Figure 4: the precision rate comparison between ODLP and ZeroX

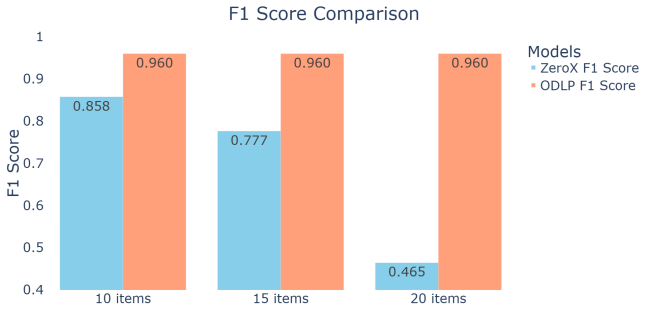

Figure 5: shows the f1-score comparison between ODLP and ZeroX

4.4 Case Analysis

During the testing process, we discovered that even when product price information was not displayed on the webpage, ODLP could extract the prices from the HTML code. However, for ZeroX, utilizing a visual modality large language model, it was unable to extract product prices. Figure 6 illustrates a website named Artfinder [7], which sells paintings and other artworks. On this website, the visual representation only includes the creator's name and price, requiring users to click on a product link to access a secondary web page and retrieve the artwork's name. When processing full-page screenshots with high visual scaling, ZeroX's performance deteriorated. This failure was due to the high degree of scaling applied to the screenshot, which compressed the visual representation related to the products, rendering it unrecognizable by the visual model. If the full-page screenshot of this website was simply sent to ChatGPT-4o-mini-2024-07-18, the model would fail to output valid product information.

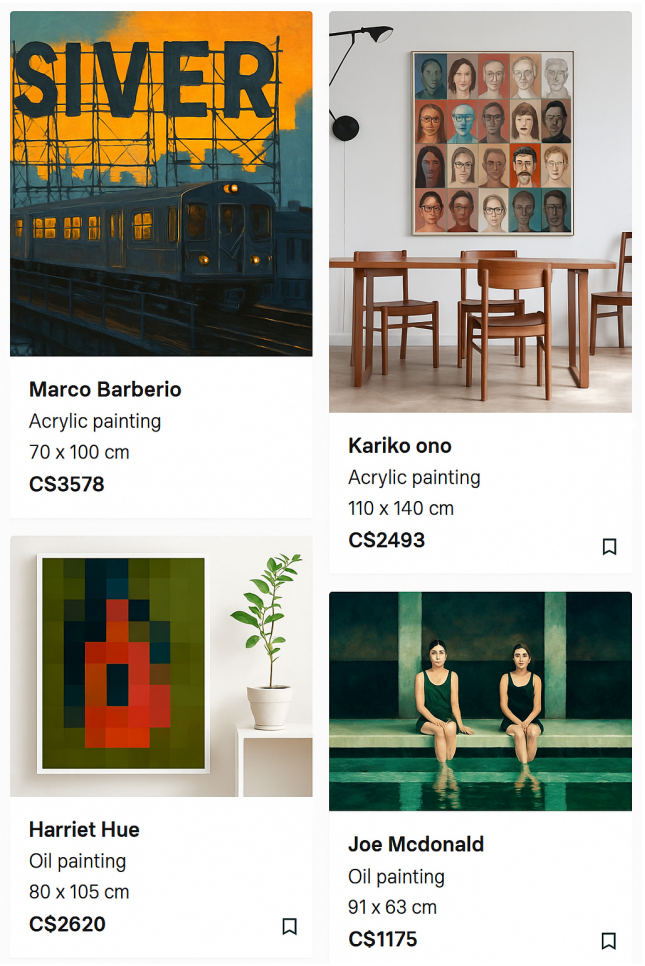

Figure 6: shows the web page screenshot from Artfinder

4.5 Disscussion

The primary advantage of ODLP lies in its exceptional adaptability across all shopping websites in the testing dataset, achieving 96% precision, recall, and F1 score. These results demonstrate that the

[7]Artfinder webpage link: https://www.artfinder.com/. Images inside figure 6 have been processed to avoid copyright issues

ODLP method can reliably and accurately extract product information, irrespective of the structural and content variations among shopping websites. Although we used commercial websites to form out testing dataset, the main focus is not about web information scrapping. Therefore we did not provide solutions for dealing with anti-scrapping mechanisms on websites or webpage's lazy loading issue. It should be noted that the present study does not prioritize the reliability of information extraction to form structural knowledge, as the majority of tasks under consideration are not critically constrained by time in typical application scenarios.

In addition to evaluating open-source programs, several commercial web information extraction tools were analyzed, including AgentQL[8] and Firecrawl[9]. Both products leverage LLMs for their information extraction capabilities. Among these, AgentQL demonstrated exceptional performance, achieving 100% precision, recall, and F1-score on the testing dataset, thereby rivalling ODLP in effectiveness. However, these commercial tools have not disclosed the methods they use to integrate large language models for information extraction, leaving their specific approaches unknown to the public. This lack of transparency makes it challenging to include them as subjects for rigorous comparative analysis in academic research. Conversely, Firecrawl exhibited significant shortcomings, failing to extract valid information from five websites within the dataset and achieving recall, precision, and F1-score metrics below 10%, underscoring its lack of robustness and reliability. Furthermore, a significant practical limitation encountered during this evaluation process was the difficulty posed by anti-scraping mechanisms commonly implemented by websites. These mechanisms often complicated the automated downloading of webpage content necessary for testing, representing a hurdle for systematic evaluation across diverse online sources.

Future improvements will focus on optimizing ODLP to address its slower processing time by reducing the number of LLM API calls and enhancing token utilization efficiency. The token usage for ODLP is now five times higher than that of ZeroX in the 10-item scale measurement. We plan to design a prompt system that can let LLM interact with DOM Tree and output suitable programming language to extract all relevant information on a certain website. These advancements aim to further solidify ODLP's applicability in scenarios requiring completeness, accuracy, and cost-effectiveness in web information extraction.

5 Conclusion

This paper presents an innovative and integrated methodology for WIE that effectively combines OCR, DOM tree parsing, and LLMs. By addressing the inherent challenges posed by diverse webpage structures and inconsistent tag naming conventions, the proposed approach demonstrates significant advancements in the accuracy, reliability, and adaptability of extracting key information from web pages.

One of the key strengths of this methodology lies in its holistic integration of visual and structural data processing. By leveraging OCR for precise text extraction and DOM tree parsing for structural analysis, the system ensures comprehensive data retrieval. The incorporation of LLMs for contextual classification and validation bridges the gap between raw extracted data and meaningful information, enhancing both accuracy and interpretability. This synergy not only mitigates the limitations of individual techniques but also establishes a scalable framework capable of adapting to a wide array of web designs and content variations.

Looking ahead, there are several promising directions for further improvement. Developing more efficient prompt systems to enable direct interaction between LLMs and DOM trees could significantly reduce the number of API calls and enhance processing speed. Additionally, exploring advanced image preprocessing techniques and leveraging higher-resolution captures may further improve OCR accuracy. Expanding the methodology to encompass a broader range of web applications beyond shopping websites will also test and enhance its generalizability and robustness.

[8]AgentQL website: https://www.agentql.com/
[9]Firecrawl website: https://www.firecrawl.dev/

References

[1] T. B. Brown et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*. https://arxiv.org/abs/2005.14165.

[2] Lin Chen, Fengli Xu, Nian Li, Zhenyu Han, Meng Wang, Yong Li, and Pan Hui. 2024. Large language model-driven meta-structure discovery in heterogeneous information network. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (KDD '24). Association for Computing Machinery (ACM), Barcelona, Spain, (Aug. 2024), 307–318. ISBN: 979-8-4007-0490-1. doi:10.1145/3637528.3671965.

[3] A. Ghosh, B. Huang, Y. Yan, and W. Lin. 2024. Enhancing healthcare user interfaces through large language models within the adaptive user interface framework. In *Information and Communication Technology: ICICT 2024*. Springer, (Feb. 2024), 1–10.

[4] P. He, J. Gao, and W. Chen. 2021. Debertav3: improving deberta using electra-style pre-training with gradient disentangled embedding sharing. *arXiv preprint*, arXiv:2111.09543. Preprint. doi:10.48550/arXiv.2111.09543.

[5] A. Kumar, K. Morabia, J. Wang, K. C. C. Chang, and A. Schwing. 2021. Cova: context-aware visual attention for webpage information extraction. *arXiv preprint arXiv:2110.12320*.

[6] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. 2019. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. *arXiv preprint*, arXiv:1910.13461. Preprint. doi:10.48550/arXiv.1910.13461.

[7] Z. Li, B. Shao, L. Shou, M. Gong, G. Li, and D. Jiang. 2023. Wiert: web information extraction via render tree. In *Proceedings of the AAAI Conference on Artificial Intelligence* number 11. Vol. 37. (June 2023), 13166–13173.

[8] W. Lin, B. Adewale, M. Li, M. Nasir, A. Sultana, R. H. Khokhar, and Y. Zhang. 2024. Dynamic web page modification for accessibility using ai and large language models. In *Proceedings of the 27th ACIS International Summer Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2024)*. Beijing, China, (July 2024).

[9] Y. Lu, Q. Liu, D. Dai, X. Xiao, H. Lin, X. Han, and H. Wu. 2022. Unified structure generation for universal information extraction. *arXiv preprint arXiv:2203.12277*.

[10] J. L. Martinez-Rodriguez, A. Hogan, and I. Lopez-Arevalo. 2020. Information extraction meets the semantic web: a survey. *Semantic Web*, 11, 2, 255–335.

[11] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, and A. Mian. 2023. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*.

[12] OpenAI. 2023. GPT-4 Technical Report. Tech. rep. Available online: https://openai.com/research/gpt-4. OpenAI. https://openai.com/research/gpt-4.

[13] J. Piskorski and R. Yangarber. 2013. Information extraction: past, present and future. In *Multi-source, Multilingual Information Extraction and Summarization*. T. Poibeau, H. Saggion, J. Piskorski, and R. Yangarber, (Eds.) Springer Berlin Heidelberg, 23–49.

[14] M. P. Polak, S. Modi, A. Latosinska, J. Zhang, C. W. Wang, S. Wang, and D. Morgan. 2024. Flexible, model-agnostic method for materials data extraction from text using general purpose language models. *Digital Discovery*, 3, 6, 1221–1235.

[15] S. Shi, C. Liu, Y. Shen, C. Yuan, and Y. Huang. 2015. Autorm: an effective approach for automatic web data record mining. *Knowledge-Based Systems*, 89, 314–331.

[16] R. Smith. 2007. An overview of the tesseract ocr engine. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*. Vol. 2. IEEE, (Sept. 2007), 629–633.

[17] E. Uzun. 2020. A novel web scraping approach using the additional information obtained from web pages. *IEEE Access*, 8, 61726–61740.

[18] Y. Wang, B. Xiao, A. Bouferguene, M. Al-Hussein, and H. Li. 2022. Vision-based method for semantic information extraction in construction by integrating deep learning object detection and image captioning. *Advanced Engineering Informatics*, 53, 101699.

[19] L. T. Wu, J. R. Lin, S. Leng, J. L. Li, and Z. Z. Hu. 2022. Rule-based information extraction for mechanical-electrical-plumbing-specific semantic web. *Automation in Construction*, 135, 104108.

[20] Yunyi Zhang, Ming Zhong, Siru Ouyang, Yizhu Jiao, Sizhe Zhou, Linyi Ding, and Jiawei Han. 2024. Automated mining of structured knowledge from text in the era of large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (KDD '24). Association for Computing Machinery (ACM), Barcelona, Spain, (Aug. 2024), 6644–6654. ISBN: 979-8-4007-0490-1. doi:10.1145/3637528.3671469.

[21] Y. Zhou, Y. Sheng, N. Vo, N. Edmonds, and S. Tata. 2021. Simplified dom trees for transferable attribute extraction from the web. *arXiv preprint arXiv:2101.02415*. https://arxiv.org/abs/2101.02415.