CS397 Independent Study

Jason Zalewski

May 18, 2024

1 The Problem

The project focuses on the challenge of automating web page scraping and information extraction. Initially, I delved into the manual process of scraping websites to gain a deeper understanding of the DOM tree and HTML structure. This hands-on experience provided a solid foundation for exploring more efficient and scalable approaches.

During this exploration, I discovered that large language models (LLMs) demonstrate remarkable accuracy in information extraction without requiring explicit training. However, these models come with limitations, such as constraints on context length and inconsistent response formatting, which can complicate the parsing process. To mitigate the context length issues, I utilized Pydepta to classify similar subtrees within the HTML and feed specific record information to the model instead of the entire HTML.

After observing the results of using LLMs in various ways, the next objective was to develop generalized questions and methodologies for evaluating the capabilities of different LLMs in web information extraction. However, I encountered challenges in formulating broadly useful questions for testing models in this manner. Consequently, I shifted my focus to exploring alternative extraction methods using LLMs, drawing inspiration from web navigation agents controlled by multimodal models.

Examining the strengths and weaknesses of two specific agents, SeeAct and Webvoyager, provided insights into how vision could enhance extraction accuracy. Attempts to adapt these web agents for information extraction revealed that they were too slow and costly for scraping individual pages, particularly for longer webpages requiring extensive scrolling. This led me to incorporate ideas primarily from Webvoyager, such as obtaining the accessibility tree and capturing screenshots of the viewport, then scrolling until reaching the bottom of the page.

Ultimately, the problem crystallized into comparing the accuracy and performance of multimodal models across different image sizes to determine the opti-

mal information density for an image to be effectively extracted using a multimodal model.

2 My Approaches

During this semester, I investigated various approaches for web navigation and information extraction using MultiModal models. Initially, I experimented with SeeAct, a web navigation system that showed potential for solving specific tasks based on textual input. However, when applied to information extraction tasks on longer pages, SeeAct encountered performance issues and limitations in handling extensive web scraping compared to manual methods or my previous approach using Pydepta and LLMs.

Seeking an alternative, I evaluated Webvoyager, another web navigation system. While Webvoyager demonstrated faster and more reliable navigation capabilities, it also had limitations, such as getting stuck in infinite loops similar to SeeAct and being too slow for efficient extraction. Inspired by Webvoyager's techniques, I incorporated some of its ideas and code into my own approach. This included capturing screenshots of the current viewport and utilizing their code to generate an accessibility tree of the HTML within the viewport, which could be fed into the model. This method yielded high accuracy, but due to the overlapping information between each viewport screenshot and the accessibility tree providing slightly more information between each image, automating the grading process proved challenging. Additionally, the accessibility tree's performance degraded exponentially as the HTML tree became longer and more complex.

To address the discrepancy between the accessibility tree and the information extracted by the MLM from the images, I developed a new approach. Instead of capturing screenshots of varying sizes and scrolling through the entire page, this method involved modifying the HTML by removing all elements except the desired section for extraction in the screenshot, which in this case was the professors' faculty information on university websites. This script allowed for configurability, enabling the comparison of MLM accuracy for extraction on the same website but with varying amounts of information in each image.

For instance, one test showcased six professors' information within an image, and this process was repeated for the entire webpage. A second test could involve displaying twelve professors' information in an image. By comparing the extraction accuracy for images with different information densities, this approach aimed to provide insights into the optimal image size for accurate extraction. Moreover, this method eliminated the use of the accessibility tree to reduce variables and focus solely on the impact of different image sizes on accuracy.

Throughout the semester, I made progress in exploring and refining techniques for web navigation and information extraction. Starting with off-the-shelf web

navigation systems like SeeAct and Webvoyager, I identified their strengths and limitations. Building upon their ideas, I developed my own approach that leveraged viewport screenshots and accessibility trees, achieving high accuracy but facing challenges in grading automation and performance scalability. Finally, I refined my approach by focusing on specific sections of the webpage and comparing extraction accuracy across different image densities.

3 Results

Link to all Datasets and information that was extracted

3.1 Manual vs Multimodal Extraction Accuracy

Number of Faculty Member Cards	Multimodal Extraction Accuracy
9	0.943
15	0.950
21	0.834

Table 1: Multimodal Extraction Accuracy for UC Faculty Webpage

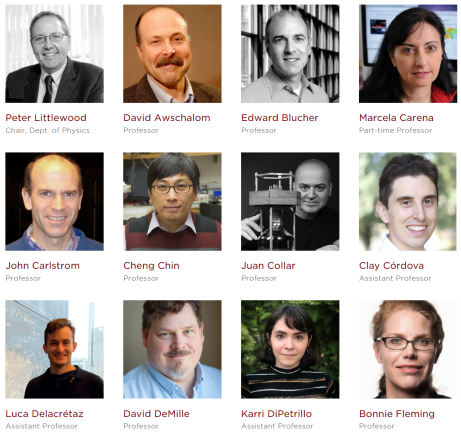

Figure 1: Example of an Image with 12 Faculty Member Cards

Number of Faculty Member Cards	Multimodal Extraction Accuracy
9	0.837
15	0.859
42	0.292

Table 2: Multimodal Extraction Accuracy for Texas Faculty Webpage

Note: The results for the Texas page had UTF-encoding and string matching issues, which made the accuracy lower than it could have been if fixed. When I tried rerunning the results I had issues with the openai api of insufficient funds, so I was not able to rerun the Texas webpage with these issues fixed.

3.2 Multimodal Extraction Accuracy and Viewport Sizes for Stanford Webpage

Viewport Size	Multimodal Extraction Accuracy
800x600	42/44
1024x768	18/18
1366x768	21/21
1920x1080	16/18
2560x1440	15/18

Table 3: Multimodal Extraction Accuracy and Viewport Sizes for Stanford Webpage

Note: The results were graded by hand. The numbers reflect issues due to scrolling on the webpage and taking screenshots. Sometimes professors' information would be cut off, and it was not considered against the model if it was not possible to extract. The grading was done by each image instead of by the full webpage, and then all the scores were added up.

Link to the data

4 Assessment

I think my progress was stronger after I started experimenting with the Webvoyager research code implementation for web navigation, as it gave me more ideas to try and use for extracting information. Initially, when trying to create my own methods, I struggled significantly to come up with new efficient ways to find information from the DOM tree. I was also attempting to devise general questions and methods to compare the accuracy of different language models. However, I couldn't find a great solution to this problem and kept encountering the same issues repeatedly. Overall though, I believe that using viewport screenshots and the accessibility tree was an effective and innovative method for automatically extracting information from a website, which was the main goal.

I think that I could have done some things better, such as pushing harder on using state-of-the-art prompting techniques. Additionally, I should have ensured cleaner results, addressing issues like the UTF encoding problem that affected the accuracy of my results.

5 Reflection

I learned a lot about the capabilities and limitations of web navigation agents controlled by LLMs. This included learning about SAM(Set-of-Mark Prompting) and other grounding methods that improve the accuracy of an MLM's understanding of images. I also discovered that MLM web navigation agents can solve basic web captchas, navigate webpages, and complete specific tasks specified through text. However, I found that certain webpages, like searching for specific Unreal Engine documentation, can confuse the agents and cause them to get stuck in an infinite loop.

One of the most useful things I learned was how to calculate the viewport of a webpage and create an image based on it. This allowed me to develop methods to feed only part of a webpage to the MLM, preventing it from being overwhelmed by the entire page. Another interesting insight was observing some of the strange behaviors of an MLM, such as flipping letters around if it can't read a word from an image.

Additionally, I learned how to build datasets for manual extraction by deleting elements of the webpage and capturing an image of the entire page. I found that I could select a specified number of children in the HTML area I was examining, delete the rest, then reset and select the next specified children. In hindsight, I think I could have experimented with inserting new elements into the HTML, which might have provided an opportunity to use Set-of-Mark Prompting to see if it improved the multimodal model's ability to extract information correctly.

6 Future Work

One area that could be further explored is developing a more efficient algorithm for extracting the accessibility tree of a viewport. As the complexity and length of the HTML increase, the current algorithm's performance degrades exponentially. Designing a faster and more accurate version of this algorithm would not only benefit information extraction tasks but also find applications in web navigation agents and other related domains.

The next aspect that could be further explored is developing an improved grading system for comparing manually extracted strings (validation tests) with LLM-generated strings. Instead of relying solely on exact string matching, which can mark answers as incorrect even if they are semantically similar but worded slightly differently, it would be beneficial to investigate alternative methods such as using distance metrics or employing natural language processing techniques like semantic similarity measures to assess the correctness of the generated answers more effectively.

Another avenue that can be explored is discovering an efficient method to leverage state-of-the-art prompting techniques for extracting information from webpage images. In the context of web navigation, visual bounding boxes have been drawn over elements on a webpage before capturing a screenshot, which aids the vision model in selecting the next action to take. Similarly, it seems plausible that incorporating visual cues or annotations on the webpage images prior to processing could potentially enhance the accuracy and effectiveness of information extraction using large language models.

References

[1] SeeAct: Visually-Grounded Program Synthesis,
https://osu-nlp-group.github.io/SeeAct/

[2] WebVoyager : Building an End-to-End Web Agent with Large Multimodal Models,
https://arxiv.org/pdf/2401.13919

[3] The Future of Web Data Mining: Insights from Multimodal and Code-based Extraction Methods
https://aclanthology.org/2024.case-1.1.pdf

[4] GPT-4V-Act: Github Repository.
https://github.com/ddupont808/GPT-4V-Act

[5] Set-of-Mark Prompting Unleashes Extraordinary Visual Grounding in GPT-4V,
https://arxiv.org/pdf/2310.11441

[6] LLaVA-Grounding: Grounded Visual Chat with Large Multimodal Models.
https://arxiv.org/pdf/2312.02949

[7] WEBLINX: Real-World Website Navigation with Multi-Turn Dialogue,
https://arxiv.org/pdf/2402.05930

[8] MIND2WEB: Towards a Generalist Agent for the Web,
https://arxiv.org/pdf/2306.06070