AccessMenu: Enhancing Usability of Online Restaurant Menus for Screen Reader Users

Nithiya Venkatraman*
Old Dominion University
Department of Computer Science
Norfolk, Virginia, USA
nvenk001@odu.edu

Akshay Kolgar Nayak*
Old Dominion University
Department of Computer Science
Norfolk, Virginia, USA
anaya001@odu.edu

Suyog Dahal
Old Dominion University
Department of Computer Science
Norfolk, Virginia, USA
sdaha005@odu.edu

Yash Prakash
Old Dominion University
Department of Computer Science
Norfolk, Virginia, USA
yprak001@odu.edu

Hae-Na Lee
Michigan State University
Department of Computer Science and Engineering
East Lansing, Michigan, USA
leehaena@msu.edu

Vikas Ashok
Old Dominion University
Department of Computer Science
Norfolk, Virginia, USA
vganjigu@odu.edu

ABSTRACT

Online food ordering has become commonplace due to its convenience. The wide variety of culinary choices, combined with fast and economical door-delivery services, encourages more people to order food online. To facilitate this process, food vendors, including restaurants, often provide full menus on their websites, typically in visual formats such as images or PDFs. While this is convenient for sighted users, blind and visually impaired (BVI) individuals face significant challenges accessing these visual menus with their screen reader assistive technology. An interview study with 12 BVI screen reader users revealed that present assistive tools do not adequately satisfy the needs of these users, with issues ranging from text-ordering errors, to inaccurate inferences (e.g., incorrectly categorizing a Caesar salad with anchovies as vegetarian), to misinterpretation of symbols and legends. Moreover, the users expressed a need for a screen reader-tailored interface to access the information in menus. To address these access barriers and users' needs, we present AccessMenu, a browser extension that automatically detects visual menus in restaurant websites, uses multi-modal large language models to extract and analyze the menu content, and re-renders it in a conveniently navigable HTML format accessible with screen readers. AccessMenu also enables BVI users to issue natural language queries, allowing them to efficiently distill specific information from the menus. In a user evaluation with 10 blind participants, AccessMenu significantly outperformed a state-of-the-art solution in usability and task workload, by providing convenient menu navigation and query-based menu filtering capabilities.

*Both authors contributed equally to this research.

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

CCS CONCEPTS

• **Human-centered computing** → **Accessibility technologies**; *Empirical studies in accessibility*.

KEYWORDS

blind, screen reader, visual impairment, restaurant menu, usability, accessibility, large language models

ACM Reference Format:
Nithiya Venkatraman, Akshay Kolgar Nayak, Suyog Dahal, Yash Prakash, Hae-Na Lee, and Vikas Ashok. 2018. AccessMenu: Enhancing Usability of Online Restaurant Menus for Screen Reader Users. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The landscape of the restaurant industry has witnessed a profound transformation in the last decade with the proliferation of online food ordering platforms. The convenience offered by online platforms has revolutionized the way consumers engage with restaurants and other dining establishments. According to recent statistics, the global online food delivery market has experienced exponential growth of $294 billion[1]. In the fast-moving and busy world, ordering food online from restaurants has become more efficient and convenient for people all over the world. To facilitate convenient online ordering, food establishments present digital online menus on their websites, so that customers can obtain an overview of available dishes along with associated information such as price, ingredients, customization options, and sometimes even pictures. While these menus significantly elevate the food-ordering experience for sighted customers, the menus pose significant access challenges for blind and visually impaired (BVI) customers, particularly those who interact with digital content using screen reader assistive technology (e.g., JAWS [72], NVDA [7], VoiceOver [10]).

[1]https://en.wikipedia.org/wiki/Online_food_ordering

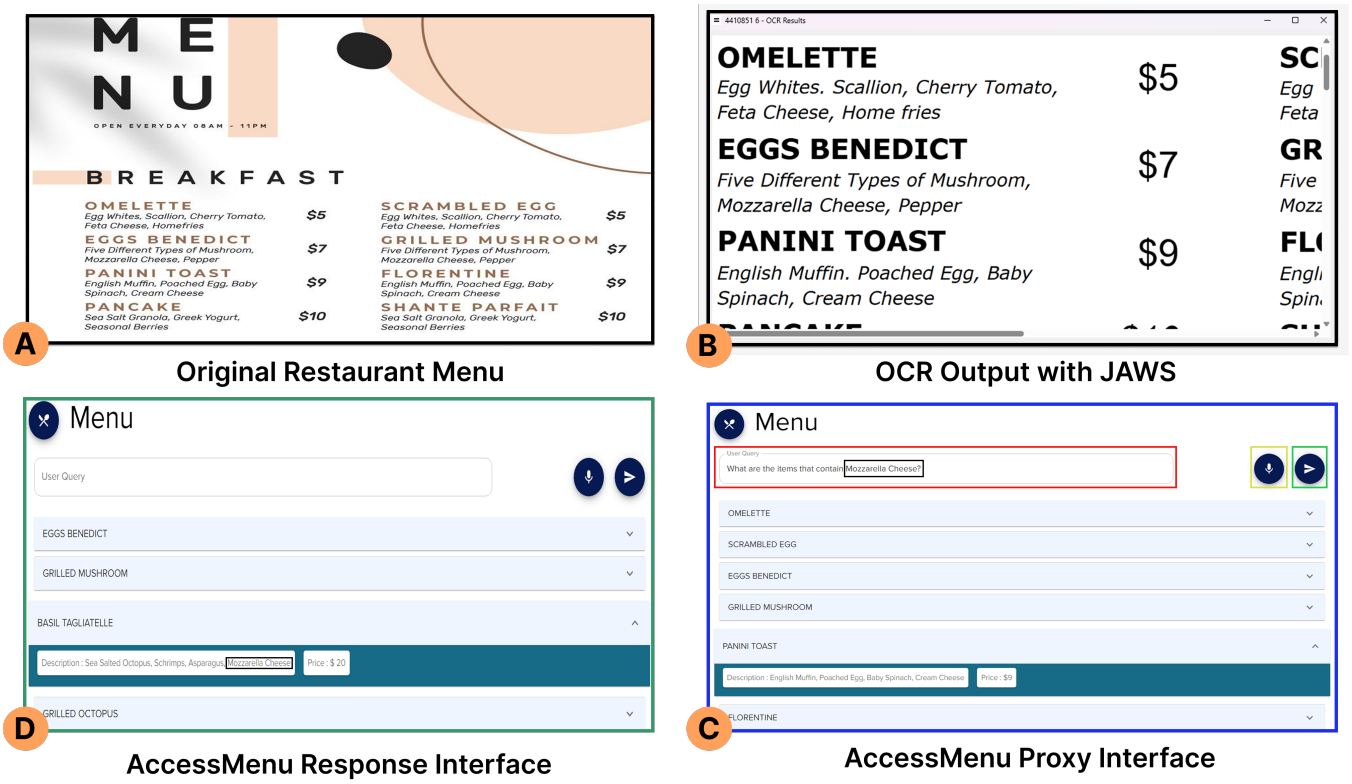

Figure 1: (A) The original restaurant menu. (B) Output from JAWS Convenient OCR. (C) The AccessMenu interface, where the red box highlights the natural language query field, the yellow box indicates the voice-input button, and the green box is the submit button. (D) The updated AccessMenu interface showing only the subset of menu items matching the user's query.

A screen reader narrates content and assists BVI users in navigating the web using keyboard shortcuts. However, this navigation is predominantly one-dimensional, requiring users to methodically traverse elements on a web page to find the desired element or information. While text elements are simply read out using text-to-speech, visual elements such as images are handled via either website-provided alternative textual descriptions, i.e., alt text, or automatic AI-generated texts describing the visual content [5, 6, 31, 41]. While such alt texts are suitable for simple images that can be fully described using captions, they are impractical and inadequate in case of complex two-dimensional document images such as restaurant menus (see Figure 1A). Consequently, blind screen reader users typically rely on AI-driven assistive tools [22, 37, 53, 75, 81] to access information in image documents such as menus.

However, in an interview study with 12 BVI participants, we found that the present assistive tools are inadequate in their ability to address the BVI users' challenges and needs with regards to interaction with visually complex and information-rich documents such as restaurant menus. The participants stated that interacting with assistive-tools' OCR outputs was cumbersome and mentally taxing with a screen reader, as the information layout in the outputs often did not *logically* match the screen reader narration order. For example, in the information extracted from the menu in Figure 1A, the screen reader reads out "Scrambled Egg" after "Omelette", instead of reading out the ingredients and price of "Omelette". The participants also reported hallucinations in AI applications due to lack of contextual awareness (e.g., answering 'vegetarian' to a question about a Caesar salad in the menu that clearly lists anchovies as one of the ingredients) and legend misinterpretations (e.g., mistaking vegetarian icon for the vegan icon). Additionally, a majority of the participants expressed a need for an alternative BVI-friendly user interface for quick-and-easy perusal of menu items, and they also provided design ideas regarding this user interface.

Informed by the findings of our interview study, we designed and developed AccessMenu, a browser extension that automatically builds a semantics-based menu model (list of menu items and properties of each item including customization options for each item, item category, etc.) from online restaurant menu images using multimodal large language models (MLLMs), and then leverages this menu model to provide an alternative menu interface tailored for both convenient screen reader navigation and natural language query access. As illustrated in Figure 1C, AccessMenu presents users with a *proxy* menu interface, where the linearly-organized menu items are conveniently navigable using basic screen reader shortcuts. Moreover, as seen in the figure, AccessMenu enables the user to quickly and accurately access specific information about menu items via natural language queries (e.g., *suggest meat-based appetizers*). Key technical innovations of this work include the development of a robust multimodal pipeline capable of extracting

and contextualizing menu information, and a seamless integration of this pipeline into a real-time browser extension. This system effectively bridges the gap between visual menu content and accessibility requirements, offering BVI users a faster and more intuitive interaction with online menus.

An evaluation of AccessMenu in a user study with 10 blind participants showed significant improvements in the usability, task workload and overall user experience with online restaurant menus, compared to a state-of-the-art OCR-based solution. A majority of the participants also stated that the AccessMenu interface would motivate them to be more active in ordering food online and consider a plethora of food options and make an informed dining choice. In sum, this paper makes the following contributions.

- The findings of an interview study detailing the usability issues faced by BVI screen reader users while accessing restaurant menus online for ordering food.
- The design and evaluation of a novel AccessMenu browser extension that presents inaccessible/unusable restaurant menus via alternative BVI-friendly interfaces using MLLMs.

2 RELATED LITERATURE

2.1 Web Interaction Using Screen Readers

Extensive research has been conducted to explore the complexities and challenges involved in interacting with web content using a screen reader [4, 9, 12, 13, 16, 43, 44, 64, 80]. An early work by Lazar et al. [43] identified significant access barriers that persist despite established accessibility guidelines [5], including poorly designed page layouts, technical conflicts between screen readers and web applications, and the absence of alt text for images. Borodin et al. [16] further investigated the strategies adopted by screen reader users to circumvent the web-interaction issues, and found that the users typically resorted to increasing the speech rate and using the headings' hotkeys to efficiently navigate the web content. More importantly, they observed that the screen reader vocabulary of most users was limited to a handful of basic screen reader shortcuts. Similar to these seminal works, more recent research efforts in this area have also investigated and uncovered numerous accessibility and usability issues of screen reader users in different web interaction scenarios [33, 70, 79]. While such investigations of issues are generic across the web, and therefore applicable to a certain extent to online restaurant menus, they do not capture the unique domain-specific issues that screen reader users face when interacting with online restaurant menus. We address this knowledge gap via an interview study with 12 blind participants, aiming to uncover their pain points, needs, and preferences while navigating and interacting with restaurant menus online.

Prior research has also explored solutions to overcome the numerous accessibility and usability challenges for screen reader users [11, 26, 27, 30, 36, 45, 59, 61, 63–65, 73, 77, 82, 83]. These solutions include automatic captioning of visual content [25, 48, 50, 60], web automation [11, 66, 67, 84], natural language assistants [12, 20, 28, 54], and even alternative third-party navigation devices [14, 15, 27, 30, 36, 46, 61, 64, 65]. While these solutions do significantly enhance usability of web screen reading in general, they are currently limited in their ability to address the specific issues that arise when interacting with online restaurant menus. The arrangement of content in a typical menu is highly visual, with the document layout itself used to implicitly convey the semantics associated with the listed menu items. Moreover, many restaurant menus are in PDF or image formats so it is not possible to use the aforementioned generic web-based solutions to address the interaction problems. To fill this void, we present AccessMenu, a solution that specifically focuses on enhancing usability of visual-rich online documents, particularly restaurant menus.

2.2 Visual Document Understanding

Visual document understanding (VDU) tasks (e.g., visual question answering) involves the interpretation and analysis of a wide range of digital documents, including but not limited to forms, tables, reports, and academic papers [8, 51, 92]. The techniques employed in VDU can be broadly classified into two primary categories. The first category focuses on accomplishing the VDU tasks by aligning images with annotations sourced from external optical character recognition (OCR) systems [34, 35, 78, 88], whereas the second category comprises approaches that process document images directly, without relying on external OCR tools [42, 47, 55].

LayoutLMv2 [88], a notable example of the first category, leverages OCR to extract text and bounding boxes from visually-rich documents, combining text, layout, and image data for enhanced document understanding. By integrating OCR output during pre-training with spatial-aware self-attention, LayoutLMv2 captures document context more effectively. A contemporary example of the second category, the OCR-free Donut model [42] simplifies VDU tasks by eliminating dependency on OCR engines, directly mapping document images to structured outputs using a 'transformer-only' architecture. Through pre-training with custom curated synthetic data (SynthDoG [42]) and fine-tuning across diverse VDU tasks, Donut has demonstrated strong performance and has also been generalized across multiple languages and document types.

More recently, large language models (LLMs) such as LMDX [62], BLIP [49], LLaVA [56], MiniGPT-4 [91], and mPLUG-Owl [90] have demonstrated significant capabilities in accomplishing VDU tasks in visually-rich documents via minimal instructions [19, 86]. However, despite the impressive zero-shot reasoning capabilities demonstrated by multimodal LLMs, studies have shown that these LLMs face challenges in comprehending text-rich images [57]. Recent studies have also explored the effectiveness of MLLMs in Visual Question Answering (VQA), an important VDU task that involves accurately responding to questions based on the visual information in documents such as receipts, forms, and research papers [39, 40, 58]. Current VQA solutions employ an assortment of natural language processing and computer vision techniques to accurately answer posed questions [38, 76, 85]. However, none of the existing multimodal LLMs have been previously investigated for their efficacy in handling unique-style documents such as restaurant menus. In this paper, we conduct an in-depth investigation of MLLMs like GPT-4 [89], Claude [21], and LLaMA 3 [24] for information extraction and reasoning tasks on documents such as restaurant menus. Additionally, we explore the efficacy of MLLMs in comprehending menu-related queries and reasoning logically over menu content to generate valid responses.

3 UNCOVERING USABILITY ISSUES

We conducted an Institutional Review Board (IRB)-approved semi-structured interview study with 12 blind participants to uncover their current interaction challenges and needs while accessing online restaurant menus.

3.1 Participants

We recruited 12 blind screen reader users (6 female, 6 male), with an average age of 49.41 years (Median = 49, SD = 16.59, Range = 31-68). The inclusion criteria required the participants to be proficient in web screen reading and familiar with restaurant websites. All participants stated that they order food through phone at least once every week. Also, none of the participants had residual vision good enough to visually interact with digital content using screen magnifiers. The participants did not have any additional impairments, such as motor or hearing difficulties, that could affect their ability to complete study tasks effectively.

3.2 Interview Design and Procedure

The interviews were semi-structured with seed questions pertaining to the following topics:

- ***Food ordering habits.*** E.g., How often do you order food? How do you order food? How do you choose restaurants for ordering food?
- ***Experience with restaurant menus.*** E.g., What assistive technologies do you use to access menus online? What issues do you typically face while accessing these menus? How do you tackle these issues?
- ***Needs and preferences.*** E.g., Do you have any design suggestions for making these menus more screen reader friendly? What kind of additional support do you think you will need to better access online menus?

The interviews were conducted remotely via Zoom conferencing software[2]. At the beginning of the study, informed consent was obtained remotely via the Docusign service [23]. The experimenter then engaged the participant in conversations about the topics, starting with the seed questions. During the interview, the participant was also encouraged to explain responses through illustrations on actual restaurant websites. Each interview lasted about 45 to 60 minutes. Each of the participants was compensated with a $25 Amazon gift card.

3.3 Data Collection and Analysis

With the participants' permission, all interviews were audio-recorded and also screen-captured (for capturing illustrations). We did not retain any personal or identifiable information besides the basic demographic details. We analyzed the collected and transcribed qualitative data using the standard open coding technique followed by axial coding [68]; we iteratively went over the user responses and identified key insights and patterns that reoccurred in the data.

3.4 Findings

The notable themes that emerged from the qualitative analysis are presented next.

[2]https://www.zoom.com/

Access menu online but order food through phone. Most participants stated that they preferred ordering over a phone call after perusing the menu online either on their computers or smartphones. These participants stated that most restaurant employees do not have the time to patiently describe the menu over the phone, and they often put them on hold for long durations. Therefore, they prefer to be 'more-or-less decided' before calling the restaurant. As for not ordering online, the participants stated that most restaurant websites are not usable and sometimes not accessible, which previously caused them to make mistakes such as ordering extra portions of food and ordering unintended dishes.

Most restaurant menus require additional assistive tools besides screen reader for access to its information. Almost all participants mentioned that they often had to rely on additional tools, predominantly OCR software to access content in menus, since these menus were mostly in image or PDF formats, both of which are not conducive to screen reader-based interaction. A few participants, who were adept at using screen readers, also mentioned using AI assistants often to query information in the menus.

Current assistive tools do not provide sufficient support to interact with menus. Most participants mentioned that the OCR outputs of present assistive tools (e.g., JAWS Convenient OCR, ABBYY FineReader) often contained errors or inconsistencies. Moreover, they also stated that mentally parsing OCR output based on audio alone was cognitively taxing, as the screen reader narration order of the OCR output did not often match their expected 'logical' order implicitly conveyed through visual cues. This was best expressed by the participant P8: "*The OCR output is often a mess. I need to figure out which part is linked to which other part. Suppose I hear Appetizers from the screen reader, I am naturally expecting the next thing to be the name of an appetizer, instead I hear eleven dollars, and now I need to figure out which dish costs eleven dollars*". Some of the participants who used other LLM-based assistive tools such as ChatGPT, mentioned that these tools often provided incorrect or confusing responses to their queries. For example, P4 stated: "*I am careful when picking food, because I don't want fish or meat in what I order. I once asked ChatGPT to list vegetarian dishes in a restaurant menu, and its response contained many dishes which my friend said had fish sauce or seafood ingredients in them. Sometimes, I think it also gets confused between vegetarian and vegan, as it only mentions vegan dishes when I ask for all vegetarian dishes.*"

Ask friends or family members for obtaining specific information. Nearly two-thirds of the participants stated that they often 'jointly' explored the menus with their sighted companions. The participants further stated that this joint interaction mostly entailed question-answering, where they asked their sighted companions a variety of questions or 'doubts' regarding the menu.

Need for an alternative interface to access menus. All participants specified a need for a 'new' interface to peruse menus using a screen reader. Seven participants mentioned that linear organization of menu items was more convenient for screen reader navigation. Two participants further suggested the idea of a popup

interface that could present the menu items in a linear arrangement, preferably as a list. One of these participants, P5, asked: "*Is it possible to put the menu items in one single list within a popup window? I can then go through menu linearly without missing anything*". Four other participants suggested including an assistant in the interface for quickly querying information in the menu. One of these participants P2 stated: "*I would rather just ask the AI to give me all the gluten-free menu items instead of going over all the items myself and filtering them out one-by-one.*"

Summary. The interview study revealed several pain points and needs of blind screen-reader users when they interact with online menu documents. From the study observations, it is clear that an alternative non-visual interface is needed that enables users to conveniently navigate the menu items, while also providing an option to query specific information in the menu. Specifically, the interface must enable convenient perusal of menu items, with the items arranged in a simple linear list. All information about a menu item should also be available at one place in the alternative interface, i.e., where the menu item is listed, irrespective of how the information is scattered in the original menu. Guided by these findings, we designed and developed the AccessMenu prototype interface which is described next.

4 SYSTEM DESIGN

4.1 AccessMenu Overview

Figure 2 presents the operational workflow of AccessMenu, embodied as a browser extension, that generates an alternative screen reader-friendly interface to peruse menu items. On any restaurant's webpage that contains the menu, users can access the AccessMenu's alternative menu interface using the 'Ctrl+J' keyboard shortcut. Specifically, this hotkey triggers the following sequence of operations in the background: (i) Extract the menu items from the image menu by instructing a multimodal large language model (MLLM) with a custom crafted prompt; and (ii) Use the MLLM output to re-render the information of menu items in the AccessMenu's conveniently-navigable linear menu interface. The AccessMenu's interface also enables the user to issue natural language queries (e.g., *list only the gluten-free items in the menu*) to obtain specific information about the menu in the interface.

For extracting the menu items using an MLLM, we adopted the Chain-of-Thought (CoT) prompting strategy [87], where we carefully handcrafted 'reasons' or 'thoughts' to ensure that the MLLM accounted for the unique aspects of information presentation in menus, for instance, use of icons or symbols (e.g., a leaf) next to items with the legend describing these icons/symbols (e.g., vegan) placed somewhere else in the menu. In the prompt, we also included instructions for the MLLM to generate the output or the 'menu model' as a collection of JSON objects (i.e., one object per menu item) to ensure consistency and prevent potential 'phantom information' arising from model hallucinations. For supporting users' natural language queries, we again crafted a custom CoT prompt with guardrails and few-shot examples [19] to ensure that the MLLM strictly based its responses on the extracted menu model. The MLLM output in this case too was in JSON format, to facilitate convenient rendering of the query responses in the AccessMenu's interface. The details of AccessMenu are provided next.

4.2 Extraction of Menu Data Items

When a user presses the 'Ctrl+J' keyboard shortcut to access the AccessMenu's interface, AccessMenu first captures a series of menu images and sends them to a backend server. This raw input of menu images are diverse and complex comprising a mix of textual, graphical, and decorative elements. These images are then used as input contextual information in a custom 'prompt' for instructing an MLLM to accurately extract menu items. We specifically employed Chain-of-Thought (CoT) prompting [87], given its suitability for this task. A snippet of our custom prompt template is shown below.

CoT Prompt Template for Menu Item Extraction

Menu: [INSERT MENU IMAGES HERE]
Task: Extract structured menu info from an image with proper categorization, icon detection, and JSON formatting.
Steps:

(1) Extract all visible text from the image and identify menu headers, item names, prices, and descriptions.
(2) Detect visual cues such as icons (e.g., a red chili) and style differences (bold titles, colored texts) and cross-reference with any provided legend.
(3) If no legend is present, infer icon meanings using common conventions (e.g., a red chili icon indicates spiciness).
(4) Apply a rigid **[JSON schema]** to enforce consistent structure.
(5) Filter out extraneous elements like watermarks, disclaimers, and decorative texts.

Examples:

- **Input:** Image of a restaurant menu
- **Raw Text Output:** [Lunch Specials, Spicy Chicken Burger $8.99, Caesar Salad $6.99, . . .]
- **Reasoning Steps:** [Detected section header "Lunch Specials" ... parsed item "Spicy Chicken Burger" with price "$8.99" ... inferred red chili icon implies "spicy" for "Spicy Chicken Burger" ... filtered out decorative footer text ... structured data using the designated [JSON schema]]
- **Final Output:**

```
{
  "menu_items": [
    {
      "name": "Spicy Chicken Burger",
      "description": "Grilled chicken ...",
      "icons": ["spicy"],
      "price": "\$8.99"
    },
    ...
  ]
}
```

As shown above, the prompt comprises different components: (i) Menu snapshots; (ii) Task description; (iii) Sequence of reasoning steps; and (iv) Demonstrative examples. Notice how the reasoning

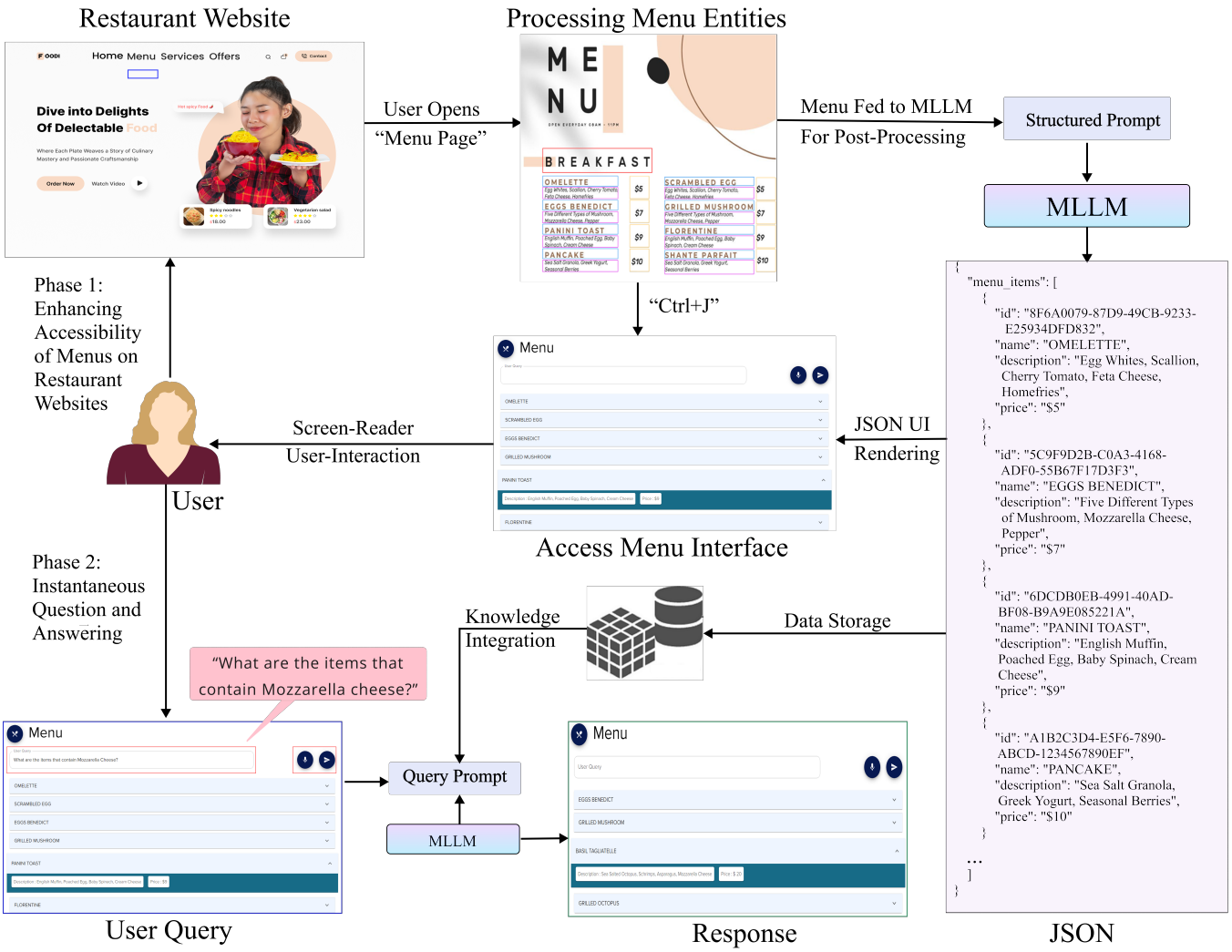

Figure 2: Architecture of AccessMenu consists of two phases: (a) Phase 1 - Information Extraction and (b) Phase 2 - Question Answering.

steps are designed to accommodate the unique aspects of restaurant menus such as spatial relationships, legends, icons, and symbols. As shown, the prompt also instructs the MLLM to structure the output, i.e., the extracted list of menu items, as a collection of JSON objects adhering to a pre-defined fixed schema for ensuring structural and processing consistency. Lastly, the prompt also ensures that the MLLM filters out extraneous elements such as watermarks, disclaimers, or promotional text. The final JSON output, which we henceforth refer to as 'menu model', is stored in the back-end server memory for the session, serving as the knowledge base for responding to subsequent user queries (refer to Section 4.3).

4.2.1 ***Evaluation.*** To evaluate the extraction performance of our approach, we collected a diverse dataset of 50 menus. The dataset was curated based on criteria such as cuisine type, menu format (e.g., à la carte, set menus), and geographical location to ensure diversity in both content and presentation styles. The dataset included menus of different layouts, with varied information structures, ranging from simple item listings to complex hierarchical representations featuring categories, subcategories, and additional legend information (e.g., spice level indicators, vegetarian legends). For each of the 50 selected menus, annotators manually created a ground truth dataset in JSON format.

We evaluated the extraction performance of three different MLLMs: GPT-4o-mini [2], Claude-3-5-Sonnet [1], and Llama 3.2-90B-Vision [3]. To measure the models' ability to capture the various components and relationships within the data presented in the menu, we used three main metrics: (i) ***Entity F1 Score (EF1):*** Captured the MLLMs' ability to extract individual menu elements, such as the names of dishes, their descriptions, prices, and any relevant legends (e.g., dietary symbols like vegan or gluten-free); (ii) ***Relationship F1 Score (RF1):*** Assessed the model's capacity to understand and extract associations between different entities. For example, it measured

how well the model linked menu items to their corresponding legends (such as indicating spice levels) or associated prices with the correct items; and (iii) ***Structural F1 Score (SF1):*** Evaluated the model's ability to maintain the hierarchical organization of the menu, such as distinguishing between main sections like appetizers, main courses, and desserts, and further recognizing subsections or groupings within each category. GPT-4o-mini outperformed the other models in extracting information from menu images, achieving an Entity F1 Score of 0.80, a Relationship F1 Score of 0.73, and a Structural F1 Score of 0.84. In comparison, Claude-3.5-Sonnet obtained an Entity F1 Score of 0.62, a Relationship F1 Score of 0.43, and a Structural F1 Score of 0.79, while Llama 3.2-90B-Vision achieved an Entity F1 Score of 0.79, a Relationship F1 Score of 0.61, and a Structural F1 Score of 0.78. We therefore integrated GPT-4o-mini in AccessMenu. Note however that AccessMenu follows a modular architecture, allowing individual components, including the MLLM, to be easily replaced with a better one if needed in the future.

4.3 Processing Contextual User Queries

To handle menu-related user queries, we again crafted a similar 'Chain-of-Thought' prompt [87] with few-shot examples, that instructed the LLM (GPT-4o-mini [2]) to comprehend and reason over the extracted 'Menu Model' (refer to Section 4.2) to generate the expected response. As in case of menu extraction, the prompt included different components: (i) Task description; (ii) User query; (iii) Menu model providing the context; (iv) Sequence of reasoning steps to be considered for generating the output; and (v) Few shot examples covering a variety of queries. The few shot examples also included 'negative' queries, i.e., queries unrelated to the menu content, to mitigate the impact of model hallucinations. The design of other menu-related few-shot examples were influenced by the participants' feedback in the earlier interview study. These few-shot queries ranged from simple filtering (e.g., *"List all vegetarian items."*) and single-hop reasoning (e.g., *"What are the desserts under $10?"*) to more complex multi-hop reasoning (e.g., *"What are the gluten-free appetizers with a drink under $20?"*), logical and arithmetic queries (e.g., *"Find me a combination of a main dish and a dessert for less than $30, with the main dish being vegetarian."*), and suggestive queries (e.g., *"What's a good vegan meal with a drink for under $25?"*). Lastly, the prompt instructed the MLLM to structure the output in JSON, which AccessMenu then parsed to render the response in its menu interface.

4.3.1 ***Evaluation.*** To evaluate the quality of responses generated by our method for user queries, we conducted a study using five restaurant menus wherein we invited 10 research volunteers to interact with the menus and pose various menu-related questions. Each volunteer was provided with 10 minutes per menu, allowing them to explore and query each of the five menus within a 50-minute study window. The MLLMs' responses to these questions were then evaluated against the ground truth using the F1 Score, with a final score of 0.83, indicating high similarity (generated vs. ground truth) and strong overall performance. The inaccuracies were primarily due to the model's difficulty in understanding relationships between items placed far apart in the menu. In some other cases, ambiguity in the user's phrasing contributed to errors in the responses. For example, when a user asked, "Give me the healthiest main course dishes," without specifying the criteria for "healthy" (e.g., low calorie or vegetarian), the system's response varied from what the user expected. In such instances, AccessMenu occasionally produced a response that did not fully align with the user's intent in the query.

4.4 User Interface

The AccessMenu's menu interface comprises a query form, a submit button and a voice-input button at the top followed by a list of extracted menu items displayed as an accordion. The accordion is made up of vertically stacked headers representing the names of items from the menu, which, upon activation (using the ENTER key), expand to reveal further details about each item (refer Figure 1). The interface was carefully crafted to enable easy navigation using simple TAB, ENTER, and ARROW shortcuts. Moreover, the web elements were optimized for accessibility, employing tab-index and ARIA (Accessible Rich Internet Applications) attributes [74]. The tab-index attribute specifies the sequence where elements would gain keyboard focus, whereas the aria label offers users additional information about the web element. By default, upon the interface activation, the initial focus is set to the 'Query' form, allowing users to smoothly transition between the form, the control buttons, and the accordion via TAB/SHIFT+TAB or ARROW shortcuts. When the user poses a query, the interface (if needed) simply refreshes the list of menu items in the accordion based on the MLLM output. Responses to factual or invalid queries (e.g., what is the price of omelette? Why is omelette so expensive?) on the other hand are simply voiced out.

4.5 Implementation Details

We implemented AccessMenu as a web browser extension, adhering to the open-source guidelines provided by Google for Chrome extensions[3]. When AccessMenu is activated, a service worker initializes and listens for specific browser events, such as the loading or closing of a page. Once a menu webpage is loaded, content scripts are dynamically injected into the page. These JavaScript files interact with the parent extension code and have access to the webpage's DOM, allowing AccessMenu to modify and enhance the page as needed. To capture menu snapshots, AccessMenu leverages the services of a Selenium driver [29]. These menu snapshots that serve as the preliminary input for subsequent extraction process are sent to the backend server via a POST request. The backend server was built using Django Rest Framework[4] and Python modules were used for all inter-module communication. Integration of the MLLM into AccessMenu was done using the LangChain framework[5], which is known to seamlessly orchestrate query processing and response generation. Additionally, the backend was containerized using Docker[6] to ensure a consistent environment across different systems, simplify dependency management, and enable seamless deployment.

[3] https://developer.chrome.com/docs/extensions/mv3/devguide/
[4] https://www.django-rest-framework.org/
[5] https://python.langchain.com/docs/get_started/introduction
[6] https://www.docker.com/

5 EVALUATION

We conducted an IRB-approved user study with 10 blind screen reader users to assess the effectiveness of AccessMenu and compare it with the status quo OCR-based assistive tool.

5.1 Participants

We enlisted 10 participants with visual impairments (6 female, 4 male), averaging 47.3 years old (Median = 47, SD = 12.7, Range = 23-66), recruited through email lists and snowball sampling. To preserve external validity, we ensured that there was no overlap between the participant groups in this study and the previous interview study. The inclusion criteria required the participants to be proficient in web browsing using the JAWS screen reader, as the study was conducted on the Windows OS platform with JAWS installed as the primary screen reader. Moreover, familiarity with the JAWS Convenient OCR feature [71] was essential, as this was the study baseline condition for assessing AccessMenu. All participants reported accessing online restaurant websites at least once a week. Participant demographics are detailed in Table 1. No participant reported having other difficulties (e.g., hearing, motor control) that could possibly affect their ability to perform study tasks.

5.2 Design

In a within-subject experimental setup, the participants were asked to freely explore the contents of a restaurant menu under the following two conditions:

- **OCR** – The participants were allowed to interact with the textual output generated by JAWS Convenient OCR [71] to access the menu content.
- **AccessMenu** – The participants were allowed to interact with the AccessMenu's alternative user interface to access the menu content.

We chose this free-form exploration task to emulate real-world scenarios where people typically start perusing menus freely without any specific focus. The participants were also asked to think-aloud during menu exploration. To minimize any learning effects, we ensured that menus from the same restaurant were not used more than once when performing the tasks under different conditions. Instead, we selected menus from two different restaurants for the two conditions. The assignment of restaurant menus to conditions and the ordering of conditions were counterbalanced across the study participants using the well-known Latin-square method [17]. A maximum of five minutes was allotted for each task.

5.3 Apparatus

The study was conducted using a Windows-based Lenovo ThinkPad laptop equipped with all the required software, including the Google Chrome browser, the AccessMenu Chrome extension, and the JAWS screen reader with JAWS Convenient OCR installed. An external QWERTY desktop keyboard was plugged in since all participants mentioned that they were familiar with the standard keyboard during the recruitment process.

5.4 Procedure

The experimenter began the study by obtaining the participant's informed consent and explaining the objectives of the study to the participant. The experimenter then allowed the participant to get familiar with AccessMenu and also configure the screen reader parameters according to their preferences. This was done to ensure that the participant's comfort level with the study apparatus was more-or-less similar to that with their own computers at home. The experimenter then asked the participant to complete the study tasks according to the predetermined counterbalanced order. After each task, the experimenter administered the SUS and NASA-TLX questionnaires [18, 32] to obtain feedback regarding the usability and task workload respectively for the corresponding study condition. All conversations were in English and the participants were compensated $25 for their time. Each study lasted about 45 minutes.

5.5 Data Collection and Analysis

Other than the SUS and NASA-TLX responses, we also recorded the participants' think-aloud utterances while doing the tasks as well as the number of items covered in each task. The experimenter also noted down any peculiar screen reader behavior from the participants while doing the tasks. We analyzed the SUS and TLX responses using standard descriptive and inferential statistical methods. Qualitative data was transcribed and analyzed using open coding and axial coding [69] to identify key insights and themes recurring in the data. We detail our findings next.

5.6 Results

5.6.1 ***Average number of items covered.*** All participants fully used the allotted 5 minutes exploring the menu in all the tasks. On average, the participants perused 14.6 items (Median = 15, Minimum = 7, Maximum = 28) under the OCR condition and 30.5 items (Median = 31.5, Minimum = 21, Maximum = 36) under the AccessMenu condition. This difference was found to be statistically significant (Wilcoxon signed rank test, Z = 2.76, W= 0, p = 0.005). Qualitative analysis of the participants' think aloud responses and experimenter's notes revealed the causes underlying this significant difference in the number of items covered between conditions. In the OCR condition, the participants were frequently complaining of getting confused by the screen reader output, and therefore they spent extra time going back-and-forth listening to the same content multiple times in order to not only comprehend it but also discover boundaries between the different menu items. Such an issue was not observed in the AccessMenu condition, where the participants went through the list one-by-one in a linear fashion. Also, in the OCR condition, the participants spent time searching for desired information regarding a group of items (e.g., "glutten-free options"), whereas in the AccessMenu condition, they avoided this overhead by simply asking the AccessMenu to filter the menu via a natural language command.

5.6.2 ***Accuracy of query responses.*** Overall, 108 queries were issued by the participants during the study, with an average of 10.8 commands (σ = 1.03) per participant. A manual inspection of the generated AccessMenu responses to these questions revealed a precision of approximately 0.71 and a recall of 0.85, resulting in an

ID	Age	Gender	Age of Vision Loss	Occupation	Preferred Screen Reader	Proficiency
P1	43	M	Cannot remember	Unemployed	JAWS	Intermediate
P2	66	F	Since birth	Self-employed	JAWS	Advanced
P3	38	F	Cannot remember	Student	JAWS	Beginner
P4	53	M	Age 3	Self-employed	JAWS	Intermediate
P5	37	F	Since birth	Social worker	JAWS	Intermediate
P6	63	M	Since birth	Corporate	JAWS	Expert
P7	59	F	Cannot remember	Teacher	NVDA	Advanced
P8	40	M	Age 5	Unemployed	JAWS	Intermediate
P9	51	F	Since birth	Corporate	JAWS	Expert
P10	23	F	Since birth	Student	NVDA	Beginner

Table 1: Demographics of blind participants in the evaluation study. All information was self-reported.

F1 score of 0.77. Error analysis revealed that the majority of inaccuracies (82.3%) were caused by ambiguities in user queries and issues in transcribing complex menu items from voice input. Ambiguous questions often led to filtering errors; for instance, when a user asked for "light snacks," the system struggled to interpret "light" as it could refer to either low-calorie items or small portions. Additionally, voice transcription errors occurred with complex menu item names (e.g., Wagyu with Béarnaise Sauce) leading to inaccuracies in response generation.

5.6.3 ***Usability and Task Workload.*** The System Usability Scale (SUS) questionnaire [18] asks participants to respond to alternating positive and negative Likert items on a scale from 1 to 5, where 1 indicating strong disagreement, 3 representing neutrality, and 5 representing strong agreement. These responses are combined into a single usability score between 0 to 100, with higher scores reflecting better usability. As shown in Figure 3a, the AccessMenu condition received significantly higher SUS ratings (Average (μ) = 69.25, Standard Deviation (σ) = 16.36) compared to the screen reader OCR condition (Average (μ) = 46.25, Standard Deviation (σ) = 11.25), as determined by a one-way ANOVA ($F = 12.08$, $p < 0.005$, $\eta^2 = 0.40$). The relatively high effect size suggests a strong influence of the condition on SUS ratings.

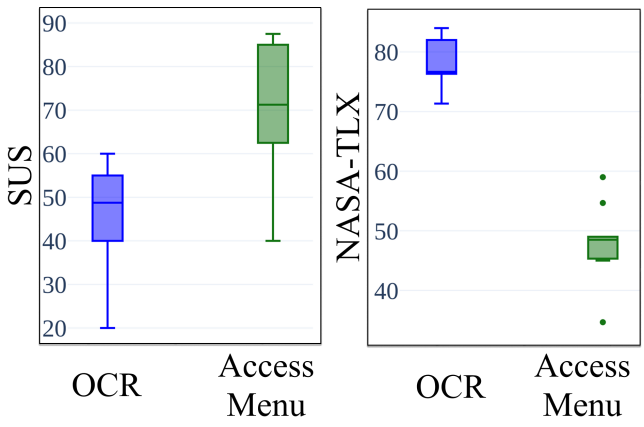

Figure 3: (a) System usability scale (SUS) and (b) NASA Task Load Index (NASA-TLX) for the two study conditions.

An in-depth examination of the System Usability Scale (SUS) responses illuminated the specific items that contributed more significantly to the observed variations in usability scores between the conditions. In particular, responses to statement 1 (I would like to use this system regularly), statement 3 (I found the system simple to use), statement 8 (I found the system unnecessarily complex), and statement 9 (I felt confident while using the system) displayed the most noticeable differences. The AccessMenu condition received consistently positive feedback on these items, while the screen reader OCR condition received unfavorable feedback. Although the responses to other SUS items followed a similar pattern, the differences were relatively less pronounced.

The NASA Task Load Index (NASA-TLX) questionnaire [32] is typically used to assess participants' perceived workload while doing the tasks. NASA-TLX scores also range from 0 to 100, but lower ratings indicate better performance, i.e., reduced taskload. We observed that there was a significant impact of the study conditions on the NASA-TLX scores (ANOVA test; $F = 161.26$, $p < 0.005$). Specifically, the TLX scores for the AccessMenu condition (Average (μ) = 48.03, Standard Deviation (σ) = 6.03) were significantly lower than those for the screen reader OCR condition (Average (μ) = 77.93, Standard Deviation (σ) = 3.67), suggesting a substantial reduction in perceived workload when using the proposed system (Figure 3b). A deeper inspection of the individual ratings revealed that responses to the Mental Demand, Effort, and Frustration subscales contributed relatively more to the difference in TLX scores between the conditions than those to the other subscales.

5.6.4 ***Qualitative Feedback.*** Qualitative analysis of the participants' feedback revealed the following notable themes:

Exploring menu items with screen reader OCR was cumbersome. All participants reported experiencing fatigue and frustration when interacting with restaurant menus using the default screen reader OCR feature. The primary challenges contributing to

this experience included *mentally linking scattered pieces of information*, *difficulty in locating specific content*, and a *need for memorization*. As P2 explained, "You have to listen back-and-forth and figure out on your own, what are all related to each other and what are not. The entire structure is complex without a proper order, so it is challenging to search for specific items I am interested in." Towards this, four other participants also expressed a preference for a system feature that will enable them to maintain a "favorite list", which the system can then use to automatically filter the items in the menu.

AccessMenu was perceived to be easy to learn and use. A majority of the participants (8) attributed their high usability ratings for AccessMenu to its simplicity and very short learning curve. They noted that memorizing a few shortcuts to access and navigate the AccessMenu's interface was a reasonable trade-off, given the substantial advantage of reduced navigation effort with the content.

Challenges in menu search and platform-wide filtering. Nearly half of the participants pointed out the lack of personalization in menu searches. Specifically, they asked if they had to go through the same search process of issuing the same queries when they accessed different menus while comparing restaurants. As P9 stated, "I want my previous search to carry over when I move to a different menu instead of typing it again." Additionally, a few participants emphasized the need for filtering at a platform-wide level rather than being limited to individual restaurant menus. They explained that while filtering menu items within a restaurant is helpful, the ability to search and filter across the menus of multiple restaurants, e.g., by relying on Google Maps platform, would be significantly more beneficial. Towards this, P3 noted, "It would be helpful if I could just search for a type of food or dietary preference across all available restaurants, rather than going through each one separately."

6 DISCUSSION

6.1 Limitations

A notable limitation of our evaluation study was that the selection of restaurant menus was confined to those with high extraction accuracy. While this strategy aimed to reduce confounding variables, it inadvertently restricted our capacity to assess the AccessMenu 'in-the-wild', i.e., in the presence of extraction inaccuracies. Future work should explore how blind participants respond and adapt to potential extraction errors, providing insights into the system's effectiveness under less ideal conditions.

Another limitation is that AccessMenu can presently support only restaurant menus in English. In real-world scenarios, menu languages often vary based on geographical location, reflecting local linguistic preferences. Extending our method to support multilingual restaurant menus is a promising direction for future research, allowing for broader applicability of our work.

The third limitation relates to the inherent latency associated with large language models such as GPT-4o-mini. While none of the participants reported any noticeable latency issues when using AccessMenu, this may not fully capture real-world scenarios where delays could potentially impact user experience. In future work, we aim to optimize the deployment process to mitigate any potential latency concerns, ensuring AccessMenu operates seamlessly and efficiently across various use cases.

Lastly, AccessMenu is currently designed exclusively for desktop environments. Given the widespread use of smartphones and the growing trend of mobile-based activities, enabling efficient non-visual web interactions with restaurant menus on smart mobile devices is essential, which is also in the scope of our future research.

6.2 Platform-Wide Menu Filtering

Our user study highlighted that while filtering items within a restaurant menu is helpful for blind users, they would immensely benefit from the ability to filter items from multiple menus across different restaurants, i.e., platform-wide level filtering by leveraging services such as Google Maps, Uber Eats, Grubhub, and DoorDash. Recognizing the increased adoption of LLM agents [52] in online platforms, we plan to develop a custom LLM agent that would enable blind users to issue filter queries at a platform-level, e.g., in Google Maps, and the agent would respond by providing an assimilated list of items extracted from multiple menus. The user would be able to therefore compare the items 'in-one-place' before deciding on the restaurant for ordering food.

6.3 Personalized Query-Based Menu Filtering

In our user study, we identified a need for personalization. Specifically, the participants wanted AccessMenu to carry over their prior search queries when navigating different restaurant menus. In the current system design, user queries are not stored, thereby requiring users to reissue the same query for each new menu, hindering efficient comparison. To address this, we plan to incorporate a personalization feature that will store user queries and automatically apply them (to the best extent possible) on other menus accessed in the same browsing session. In addition to queries, we also plan to store and apply user preferences such as allergen-related filters. For example, if a user requests to exclude items with specific allergens, the system will remember this preference and curate other subsequently accessed menus accordingly, providing a tailored, user-centric experience.

7 CONCLUSION

In this paper, we introduced AccessMenu, an intelligent browser extension designed to enhance the usability of online restaurant menus for blind and visually impaired (BVI) users who rely on screen readers. The design of AccessMenu was based on the findings of an interview study with 12 participants, which illuminated the various pain points and needs of blind screen reader users regarding online restaurant menus. AccessMenu provides an alternative screen reader-friendly interface to conveniently peruse information in the menus. The AccessMenu interface also supports natural language queries, enabling users to swiftly retrieve relevant information without the need to manually scan the entire menu. The findings from user evaluations showed that AccessMenu significantly improved usability, surpassing the capabilities of status quo OCR-based solutions.

ACKNOWLEDGMENTS

We sincerely thank Amanda Kelly Dcosta, Aditya Vishal, Sri Harshitha Gattu and Sushmitha Halli Sudhakara for their invaluable contributions to this work.

REFERENCES

[1] [n. d.]. claude. ([n. d.]). https://www.anthropic.com/news/claude-3-5-sonnet

[2] [n. d.]. gpt4. ([n. d.]). https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

[3] [n. d.]. llama. ([n. d.]). https://www.llama.com/

[4] [n. d.]. w3org. ([n. d.]). https://www.w3.org/TR/2006/WDaria-roadmap-20060926/

[5] Nov, 2008. WCAG. (Nov, 2008). https://www.w3.org/WAI/WCAG22/quickref/?versions=2.1#text-alternatives/

[6] Nov, 2008. webaim. (Nov, 2008). https://webaim.org/techniques/alttext/

[7] NV Access. 2018. NVDA screen-reader.

[8] Kriti Aggarwal, Aditi Khandelwal, Kumar Tanmay, Owais Khan Mohammed, Qiang Liu, Monojit Choudhury, Hardik Chauhan, Subhojit Som, Vishrav Chaudhary, and Saurabh Tiwary. 2023. DUBLIN: Visual Document Understanding By Language-Image Network. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*. 693–706.

[9] Patrizia Andronico, Marina Buzzi, Carlos Castillo, and Barbara Leporini. 2006. Improving search engine interfaces for blind users: a case study. *Universal Access in the Information Society* 5 (2006), 23–40.

[10] Inc Apple. 2023. *VoiceOver*. https://www.apple.com/voiceover/info/guide/_1121.html

[11] Vikas Ashok, Syed Masum Billah, Yevgen Borodin, and IV Ramakrishnan. 2019. Auto-suggesting browsing actions for personalized web screen reading. In *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization*. 252–260.

[12] Vikas Ashok, Yury Puzis, Yevgen Borodin, and IV Ramakrishnan. 2017. Web screen reading automation assistance using semantic abstraction. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*. 407–418.

[13] Shirley Ann Becker. 2009. Web Accessibility and Compliance Issues. In *Encyclopedia of Information Science and Technology, Second Edition*. IGI Global, 4047–4052.

[14] Syed Masum Billah, Vikas Ashok, Donald E Porter, and IV Ramakrishnan. 2017. Speed-dial: A surrogate mouse for non-visual web browsing. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*. 110–119.

[15] Syed Masum Billah, Vikas Ashok, Donald E Porter, and IV Ramakrishnan. 2018. SteeringWheel: a locality-preserving magnification interface for low vision web browsing. In *Proceedings of the 2018 CHI conference on human factors in computing systems*. 1–13.

[16] Yevgen Borodin, Jeffrey P Bigham, Glenn Dausch, and IV Ramakrishnan. 2010. More than meets the eye: a survey of screen-reader browsing strategies. In *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A)*. 1–10.

[17] James V Bradley. 1958. Complete counterbalancing of immediate sequential effects in a Latin square design. *J. Amer. Statist. Assoc.* 53, 282 (1958), 525–528.

[18] John Brooke. 1996. SUS: A 'quick and dirty'usability scale. Usability Evaluation in Industry. PW Jordan, B Thomas, BA Weerdmeester and AL McClelland.

[19] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[20] Giovanni Campagna, Silei Xu, Rakesh Ramesh, Michael Fischer, and Monica S Lam. 2018. Controlling fine-grain sharing in natural language with a virtual assistant. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 3 (2018), 1–28.

[21] Loredana Caruccio, Stefano Cirillo, Giuseppe Polese, Giandomenico Solimando, Shanmugam Sundaramurthy, and Genoveffa Tortora. 2024. Claude 2.0 large language model: Tackling a real-world classification problem with a new iterative prompt engineering approach. *Intelligent Systems with Applications* 21 (2024), 200336.

[22] Yihao Ding, Siwen Luo, Hyunsuk Chung, and Soyeon Caren Han. 2023. VQA: A new dataset for real-world VQA on PDF documents. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 585–601.

[23] DocuSign. 2025. Join More Than 1 Billion Users Who Trust DocuSign. https://www.docusign.com Accessed: [Your Access Date].

[24] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).

[25] Yansong Feng and Mirella Lapata. 2012. Automatic caption generation for news images. *IEEE transactions on pattern analysis and machine intelligence* 35, 4 (2012), 797–812.

[26] Javedul Ferdous, Hae-Na Lee, Sampath Jayarathna, and Vikas Ashok. 2022. InSupport: Proxy Interface for Enabling Efficient Non-Visual Interaction with Web Data Records. In *27th International Conference on Intelligent User Interfaces*. 49–62.

[27] Javedul Ferdous, Sami Uddin, and Vikas Ashok. 2021. Semantic table-of-contents for efficient web screen reading. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing*. 1941–1949.

[28] Fernando A Mikic Fonte, Martín Llamas Nistal, Martín Llamas Nistal, and Manuel Caeiro Rodríguez. 2016. NLAST: A natural language assistant for students. In *2016 IEEE global engineering education conference (EDUCON)*. IEEE, 709–713.

[29] Boni García, Mario Munoz-Organero, Carlos Alario-Hoyos, and Carlos Delgado Kloos. 2021. Automated driver management for selenium WebDriver. *Empirical Software Engineering* 26 (2021), 1–51.

[30] Cole Gleason, Amy Pavel, Emma McCamey, Christina Low, Patrick Carrington, Kris M Kitani, and Jeffrey P Bigham. 2020. Twitter A11y: A browser extension to make Twitter images accessible. In *Proceedings of the 2020 chi conference on human factors in computing systems*. 1–12.

[31] Darren Guinness, Edward Cutrell, and Meredith Ringel Morris. 2018. Caption crawler: Enabling reusable alternative text descriptions using reverse image search. In *Proceedings of the 2018 chi conference on human factors in computing systems*. 1–11.

[32] Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology*. Vol. 52. Elsevier, 139–183.

[33] Sayed Kamrul Hasan and Terje Gjøsæter. 2021. Screen Reader Accessibility Study of Interactive Maps. In *International Conference on human-computer interaction*. Springer, 232–249.

[34] Teakgyu Hong, Donghyun Kim, Mingi Ji, Wonseok Hwang, Daehyun Nam, and Sungrae Park. 2022. Bros: A pre-trained language model focusing on text and layout for better key information extraction from documents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 10767–10775.

[35] Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM International Conference on Multimedia*. 4083–4091.

[36] Mina Huh, Yi-Hao Peng, and Amy Pavel. 2023. GenAssist: Making image generation accessible. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–17.

[37] Wonseok Hwang, Seonghyeon Kim, Minjoon Seo, Jinyeong Yim, Seunghyun Park, Sungrae Park, Junyeop Lee, Bado Lee, and Hwalsuk Lee. 2019. Post-ocr parsing: building simple and robust parser via bio tagging. In *Workshop on Document Intelligence at NeurIPS 2019*.

[38] Md Farhan Ishmam, Md Sakib Hossain Shovon, Muhammad Firoz Mridha, and Nilanjan Dey. 2024. From image to language: A critical analysis of visual question answering (vqa) approaches, challenges, and opportunities. *Information Fusion* 106 (2024), 102270.

[39] Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. 2018. Dvqa: Understanding data visualizations via question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5648–5656.

[40] Kushal Kafle, Robik Shrestha, Scott Cohen, Brian Price, and Christopher Kanan. 2020. Answering questions about data visualizations using efficient bimodal fusion. In *Proceedings of the IEEE/CVF Winter conference on applications of computer vision*. 1498–1507.

[41] Muiz Ahmed Khan, Pias Paul, Mahmudur Rashid, Mainul Hossain, and Md Atiqur Rahman Ahad. 2020. An AI-based visual aid with integrated reading assistant for the completely blind. *IEEE Transactions on Human-Machine Systems* 50, 6 (2020), 507–517.

[42] Geewook Kim, Teakgyu Hong, Moonbin Yim, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2021. Donut: Document understanding transformer without ocr. *arXiv preprint arXiv:2111.15664* 7, 15 (2021), 2.

[43] Jonathan Lazar, Aaron Allen, Jason Kleinman, and Chris Malarkey. 2007. What frustrates screen reader users on the web: A study of 100 blind users. *International Journal of human-computer interaction* 22, 3 (2007), 247–269.

[44] Jonathan Lazar, Abiodun Olalere, and Brian Wentz. 2012. Investigating the accessibility and usability of job application web sites for blind users. *Journal of Usability Studies* 7, 2 (2012).

[45] Hae-Na Lee and Vikas Ashok. 2022. Customizable Tabular Access to Web Data Records for Convenient Low-vision Screen Magnifier Interaction. *ACM Transactions on Accessible Computing (TACCESS)* 15, 2 (2022), 1–22.

[46] Hae-Na Lee, Vikas Ashok, and IV Ramakrishnan. 2020. Rotate-and-Press: A Non-visual Alternative to Point-and-Click?. In *International Conference on Human-Computer Interaction*. Springer, 291–305.

[47] Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. 2023. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *International Conference on Machine Learning*. PMLR, 18893–18912.

[48] Maurizio Leotta, Fabrizio Mori, and Marina Ribaudo. 2023. Evaluating the effectiveness of automatic image captioning for web accessibility. *Universal access in the information society* 22, 4 (2023), 1293–1313.

[49] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*. PMLR, 19730–19742.

[50] Sheng Li, Zhiqiang Tao, Kang Li, and Yun Fu. 2019. Visual to text: Survey of image and video captioning. *IEEE Transactions on Emerging Topics in Computational Intelligence* 3, 4 (2019), 297–312.

[51] Xin Li, Yunfei Wu, Xinghua Jiang, Zhihao Guo, Mingming Gong, Haoyu Cao, Yinsong Liu, Deqiang Jiang, and Xing Sun. 2024. Enhancing visual document understanding with contrastive learning in large visual-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15546–15555.

[52] Yuanchun Li, Hao Wen, Weijun Wang, Xiangyu Li, Yizhen Yuan, Guohong Liu, Jiacheng Liu, Wenxing Xu, Xiang Wang, Yi Sun, et al. 2024. Personal llm agents: Insights and survey about the capability, efficiency and security. *arXiv preprint arXiv:2401.05459* (2024).

[53] Christos Liambas and Miltiadis Saratzidis. 2016. Autonomous OCR dictating system for blind people. In *2016 IEEE Global Humanitarian Technology Conference (GHTC)*. IEEE, 172–179.

[54] Kate Lister, Tim Coughlan, Francisco Iniesto, Nick Freear, and Peter Devine. 2020. Accessible conversational user interfaces: considerations for design. In *Proceedings of the 17th international web for all conference*. 1–11.

[55] Fangyu Liu, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Yasemin Altun, Nigel Collier, and Julian Martin Eisenschlos. 2022. Matcha: Enhancing visual language pretraining with math reasoning and chart derendering. *arXiv preprint arXiv:2212.09662* (2022).

[56] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Visual instruction tuning. *Advances in neural information processing systems* 36 (2024).

[57] Yuliang Liu, Zhang Li, Biao Yang, Chunyuan Li, Xucheng Yin, Cheng-lin Liu, Lianwen Jin, and Xiang Bai. 2023. On the hidden mystery of ocr in large multimodal models. *arXiv preprint arXiv:2305.07895* (2023).

[58] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2200–2209.

[59] Valentyn Melnyk, Vikas Ashok, Valentyn Melnyk, Yury Puzis, Yevgen Borodin, Andrii Soviak, and IV Ramakrishnan. 2015. Look ma, no aria: generic accessible interfaces for web widgets. In *Proceedings of the 12th International Web for All Conference*. 1–4.

[60] Yue Ming, Nannan Hu, Chunxiao Fan, Fan Feng, Jiangwan Zhou, and Hui Yu. 2022. Visuals to text: A comprehensive review on automatic image captioning. *IEEE/CAA Journal of Automatica Sinica* 9, 8 (2022), 1339–1365.

[61] Juan Nino, Sherezada Ochoa, Jocelyne Kiss, Geoffreyjen Edwards, Ernesto Morales, James Hutson, Frédérique Poncet, and Walter Wittich. 2024. Assistive Technologies for Internet Navigation: A Review of Screen Reader Solutions for the Blind and Visually Impaired. *International Journal of Recent Engineering Science* 11, 6 (2024).

[62] Vincent Perot, Kai Kang, Florian Luisier, Guolong Su, Xiaoyu Sun, Ramya Sree Boppana, Zilong Wang, Zifeng Wang, Jiaqi Mu, Hao Zhang, et al. 2023. Lmdx: Language model-based document information extraction and localization. *arXiv preprint arXiv:2309.10952* (2023).

[63] Yash Prakash, Akshay Kolgar Nayak, Shoaib Mohammed Alyaan, Pathan Aseef Khan, Hae-Na Lee, and Vikas Ashok. 2024. Improving Usability of Data Charts in Multimodal Documents for Low Vision Users. In *Proceedings of the 26th International Conference on Multimodal Interaction*. 498–507.

[64] Yash Prakash, Akshay Kolgar Nayak, Mohan Sunkara, Sampath Jayarathna, Hae-Na Lee, and Vikas Ashok. 2024. All in One Place: Ensuring Usable Access to Online Shopping Items for Blind Users. *Proceedings of the ACM on Human-Computer Interaction* 8, EICS (2024), 1–25.

[65] Yash Prakash, Mohan Sunkara, Hae-Na Lee, Sampath Jayarathna, and Vikas Ashok. 2023. AutoDesc: Facilitating Convenient Perusal of Web Data Items for Blind Users. In *Proceedings of the 28th International Conference on Intelligent User Interfaces*.

[66] Yury Puzis. 2012. Accessible web automation interface: a user study. In *Proceedings of the 14th international ACM SIGACCESS conference on Computers and accessibility*. 291–292.

[67] Yury Puzis, Yevgen Borodin, Faisal Ahmed, and IV Ramakrishnan. 2012. An intuitive accessible web automation user interface. In *Proceedings of the International Cross-Disciplinary Conference on Web Accessibility*. 1–4.

[68] Johnny Saldaña. 2015. *The coding manual for qualitative researchers*. Sage.

[69] Johnny Saldaña. 2021. The coding manual for qualitative researchers. (2021).

[70] Anastasia Schaadhardt, Alexis Hiniker, and Jacob O Wobbrock. 2021. Understanding blind screen-reader users' experiences of digital artboards. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–19.

[71] Freedom Scientific. 2020. JAWS OCR, What It Is and How It Works! https://www.freedomscientific.com/webinars/jaws-ocr-what-it-is-and-how-it-works/ Accessed: Mar. 7, 2025.

[72] Freedom Scientific. 2020. JAWS ® – Freedom Scientific. http://www.freedomscientific.com/products/software/jaws/.

[73] Woosuk Seo and Hyunggu Jung. 2022. Challenges and opportunities to improve the accessibility of YouTube for people with visual impairments as content creators. *Universal Access in the Information Society* 21, 3 (2022), 767–770.

[74] Weishi Shi, Heather Moses, Qi Yu, Samuel Malachowsky, and Daniel E Krutz. 2023. All: Supporting experiential accessibility education and inclusive software development. *ACM Transactions on Software Engineering and Methodology* 33, 2 (2023), 1–30.

[75] Shalini Sonth and Jagadish S Kallimani. 2017. OCR based facilitator for the visually challenged. In *2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*. IEEE, 1–7.

[76] Yash Srivastava, Vaishnav Murali, Shiv Ram Dubey, and Snehasis Mukherjee. 2021. Visual question answering using deep learning: A survey and performance analysis. In *Computer Vision and Image Processing: 5th International Conference, CVIP 2020, Prayagraj, India, December 4-6, 2020, Revised Selected Papers, Part II 5*. Springer, 75–86.

[77] Mohan Sunkara, Yash Prakash, Hae-Na Lee, Sampath Jayarathna, and Vikas Ashok. 2023. Enabling Customization of Discussion Forums for Blind Users. *Proceedings of the ACM on Human-Computer Interaction* 7, EICS (2023), 1–20.

[78] Zineng Tang, Ziyi Yang, Guoxin Wang, Yuwei Fang, Yang Liu, Chenguang Zhu, Michael Zeng, Cha Zhang, and Mohit Bansal. 2023. Unifying vision, text, and layout for universal document processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 19254–19264.

[79] Sachin Tanwar and PVM Rao. 2024. Inequality in User Experience: Can Mobile User Interfaces that Help Sighted Users Create Barriers for Visually Challenged People?. In *International Conference on Computers Helping People with Special Needs*. Springer, 19–30.

[80] Mary Frances Theofanos and Janice Redish. 2003. Bridging the gap: between accessibility and usability. *interactions* 10, 6 (2003), 36–51.

[81] Rubèn Tito, Minesh Mathew, CV Jawahar, Ernest Valveny, and Dimosthenis Karatzas. 2021. Icdar 2021 competition on document visual question answering. In *Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part IV 16*. Springer, 635–649.

[82] Tejal Tiwary and Rajendra Prasad Mahapatra. 2023. Enhancement in web accessibility for visually impaired people using hybrid deep belief network–bald eagle search. *Multimedia Tools and Applications* (2023), 1–22.

[83] Utku Uckun, Rohan Tumkur Suresh, Md Javedul Ferdous, Xiaojun Bi, IV Ramakrishnan, and Vikas Ashok. 2022. Taming User-Interface Heterogeneity with Uniform Overlays for Blind Users. (2022).

[84] Helmut Vieritz, Farzan Yazdi, Daniel Schilberg, Peter Göhner, and Sabina Jeschke. 2011. User-centered design of accessible web and automation systems. In *Information Quality in e-Health: 7th Conference of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society, USAB 2011, Graz, Austria, November 25-26, 2011. Proceedings 7*. Springer, 367–378.

[85] Peng Wang, Qi Wu, Chunhua Shen, and Anton Van den Hengel. 2017. The vqa-machine: Learning how to use existing vision algorithms to answer new questions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1173–1182.

[86] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652* (2021).

[87] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[88] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al. 2020. Layoutlmv2: Multimodal pre-training for visually-rich document understanding. *arXiv preprint arXiv:2012.14740* (2020).

[89] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2023. The dawn of lmms: Preliminary explorations with gpt-4v (vision). *arXiv preprint arXiv:2309.17421* 9, 1 (2023), 1.

[90] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. 2023. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178* (2023).

[91] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592* (2023).

[92] Wang Zhu, Alekh Agarwal, Mandar Joshi, Robin Jia, Jesse Thomason, and Kristina Toutanova. 2023. Efficient End-to-End Visual Document Understanding with Rationale Distillation. *arXiv preprint arXiv:2311.09612* (2023).