Conversational Web Browsing: Voice-Only Navigation

Daniele Farriciello and Jing Hua Ye
Department of Computer Science
Munster Technological University
Cork City, Republic of Ireland
e-mail: daniele.farriciello@mymtu.ie | jinghua.ye@mtu.ie

***Abstract*—Web browsing shapes how we work and connect, yet it still relies heavily on the mouse and keyboard. For people with physical disabilities or professionals in hands-busy environments, this dependence makes accessing the web frustrating or impossible. This study presents a solution, a VoiceNav system that allows users to browse websites entirely using voice commands. The goal is to make this system work automatically on any website, bridging the gap by enabling users to browse and control websites entirely through voice commands. By processing natural speech, analyzing the underlying structure of web pages, and mapping spoken instructions to precise actions, the system transforms traditional navigation into a more intuitive and conversational experience. A critical feature is its ability to resolve ambiguity. When faced with multiple similar elements, such as two buttons labeled "Upload," the system asks the user to clarify their intent, mirroring real-life communication and ensuring that interactions remain accurate and dependable. In its current prototype, the system focuses primarily on voice interaction and real-time understanding of the page. Spoken requests are converted into structured commands by a parser, while the page is broken down through Document Object Model (DOM) analysis to identify the most relevant interactive elements. This enables support for common actions, such as opening websites, clicking buttons and links, scrolling, navigating menus, and typing into fields, without requiring any site-specific setup. Initial testing shows strong performance on navigation-centric browsing, with an overall success rate of around 80%, driven primarily by robust execution of general navigation and clicking actions (e.g., opening links, selecting buttons, and scrolling). The main weaknesses appear in form-heavy scenarios, especially when entering emails or other structured inputs, where transcription and formatting errors can reduce precision. These results highlight both the practicality of voice-first browsing today and the areas where more intelligent input handling and stricter validation would further enhance reliability. By demonstrating the capabilities of voice-based web navigation, this project showcases how accessibility and modern technology can collaborate to enhance everyday life. It offers not just a practical tool, but also a glimpse of a future where interacting with the web feels more natural, inclusive, and human.**

Keywords-web browsing, voice-based, personalized, conversational, hands-free

I. Introduction

Web browsing remains one of the most common digital activities, yet interaction with websites still depends heavily on the mouse and keyboard. This dependency limits the adoption of more natural human–computer interaction and creates barriers in hands-busy scenarios, where users cannot conveniently switch between physical tasks and manual browser control.

These barriers are more severe for people with motor impairments (e.g., limited hand mobility or conditions affecting fine control), for individuals recovering from injuries, and for older adults experiencing reduced mobility. While accessibility standards, such as the Web Content Accessibility Guidelines (WCAG) encourage inclusive design, many real-world websites are still difficult to operate without conventional input devices, leaving a gap between technical capability and practical inclusion.

This study addresses that gap by presenting VoiceNav, a browser-based system that enables users to navigate and operate websites using natural voice commands, without requiring site-specific customization. Users can issue direct instructions such as opening a website, scrolling, clicking links or buttons, and entering text into input fields, allowing common browsing tasks to be completed hands-free. The system is designed to work on arbitrary websites by analyzing the page structure and mapping spoken intent to concrete browser actions.

A key challenge in voice-driven interaction is ambiguity: a page may contain multiple similar interactive elements (for example, two buttons labeled "Upload"). To maintain reliable control, the system incorporates a clarification step that prompts the user when multiple plausible targets exist, mirroring how ambiguity is resolved in human conversation. By combining speech recognition, intent parsing, DOM-based element discovery, and action execution in a modular workflow, the approach aims to make voice-first browsing more usable, dependable, and scalable for accessibility and everyday hands-busy use cases.

The general relevant background on this field is discussed in Section II. The design of the VoiceNav system and the evaluation methodology are presented in Section III. The prototype of this system is narrated in Section IV. The evaluation of the performance of the VoiceNav system is articulated in Section V. The final remarks and the future extension of this system are presented in Section VI.

II. Background

Recent advances in web technologies and interaction paradigms have significantly transformed how users access and navigate online content. These developments have improved inclusivity by enabling alternative interaction modalities that reduce reliance on traditional input devices, such as keyboards, mice, and touchscreens. Among these, hands-free, voice-driven web browsing has emerged as a promising approach for enhancing accessibility and usability, particularly for users with motor impairments or in contexts where manual interaction is impractical.

Hands-free web browsing enables users to navigate websites, retrieve information, and perform complex tasks using spoken language. At its core, this interaction paradigm relies on Automatic Speech Recognition (ASR) systems, which convert spoken audio into textual representations by analyzing phonetic patterns and linguistic structures [1]. While early ASR systems were constrained by limited vocabularies and sensitivity to noise and accents, modern deep learning–based approaches have substantially improved robustness and accuracy across diverse speakers and environments [2].

Once speech is transcribed, natural language processing (NLP) techniques are employed to interpret user intent and map spoken commands to executable browser actions. These commands range from simple navigational requests (e.g., "scroll down") to complex multi-step instructions (e.g., "search for running shoes under $100 and add the first result to the cart"). NLP models analyze syntax, semantics, and contextual cues to infer user goals and translate them into appropriate web interactions [3]. This capability is central to enabling natural, flexible voice-based browsing experiences.

Voice-driven browsing is closely related to the evolution of voice assistants, such as Siri, Alexa, and Google Assistant, but places a stronger emphasis on direct interaction with web page elements, including forms, buttons, multimedia content, and dynamically generated interfaces. This requires precise grounding of language commands to the Document Object Model (DOM) and continuous adaptation to changing web states [4]. Human–Computer Interaction (HCI) research plays a crucial role in shaping these systems by guiding interface design, feedback mechanisms, and error-recovery strategies to support natural and trustworthy interaction [4].

Key advantages of voice-driven browsing include greater accessibility for users with motor disabilities, more natural and hands-free interaction for multitasking scenarios (e.g., driving or cooking), and enhanced usability for users with limited literacy or language skills. Furthermore, real-time continuous speech recognition enables always-on listening modes, reducing the need for repeated activation phrases and providing smoother user experiences.

Despite its advantages, hands-free browsing presents several challenges. ASR systems must handle background noise, accents, homophones, and disfluencies, while NLP components must resolve ambiguous or underspecified commands. Furthermore, dynamic and visually complex web layouts complicate the mapping between language and actionable elements, necessitating robust error detection and correction mechanisms to maintain user confidence [3].

Recent progress in large language models (LLMs) has significantly advanced voice-based web interaction by enabling deeper contextual understanding, multi-step reasoning, and iterative error correction. Frameworks, such as DexAssist demonstrate how dual-LLM architectures can separate planning and execution monitoring, leading to improved task success rates in complex browsing scenarios [3]. Similarly, multimodal approaches like WebVoyager incorporate visual perception to interact with real-world websites more effectively [5]. These developments represent a shift from rigid command-based systems toward intelligent, adaptive web agents capable of natural language interaction.

Understanding the evolution and technical foundations of hands-free web browsing is essential for situating current research within the broader fields of computer science and HCI. This work builds upon advances in ASR, NLP, accessibility-driven design, and LLM-based reasoning to contribute to the development of inclusive, intelligent web interaction systems.

A. Speech Recognition and Natural Language Processing

At the foundational level, hands-free browsing relies heavily on ASR and NLP technologies [1][2]. Speech recognition serves as the primary input mechanism, converting acoustic signals into digital text representations. This process involves complex signal processing, phonetic modeling, and language modeling components that must operate reliably across diverse speakers, environments, and linguistic variations.

Modern speech recognition systems employ deep learning architectures, including recurrent neural networks, transformer models, and connectionist temporal classification (CTC) algorithms to achieve robust performance [2]. The challenge extends beyond simple transcription to include understanding speaker intent, handling disfluencies, and adapting to domain-specific vocabularies relevant to web browsing tasks.

NLP complements speech recognition by interpreting the semantic content of transcribed utterances. NLP techniques enable systems to parse complex instructions, resolve ambiguities, and map user intentions to executable browser actions [3]. This semantic understanding is crucial for handling the varied and often imprecise nature of spoken commands in real-world browsing scenarios.

B. Human-Computer Interaction and Accessibility

The accessibility dimension is particularly significant, as voice-driven browsing serves as an assistive technology for users with motor disabilities, visual impairments, or other conditions that limit traditional input methods. This positions the research within the broader context of universal design and inclusive technology development, aligning with established accessibility standards and guidelines.

C. Speech Recognition Technologies

Speech recognition technology has evolved significantly from early template-matching approaches to sophisticated deep learning systems. Traditional ASR systems relied on hidden Markov models and Gaussian mixture models, which required extensive training data and performed poorly with speaker variations [2].

Contemporary ASR systems employ end-to-end neural architectures that can learn directly from raw audio to text mappings. These systems demonstrate improved robustness to noise, accents, and speaking styles, making them more suitable for real-world browsing applications [1]. However, challenges remain in handling domain-specific terminology, proper nouns, and the informal language patterns common in voice commands.

Recent research has focused on accent-specific adaptation techniques that improve recognition accuracy for diverse user populations [2]. This work is particularly relevant for voice browsing systems that must serve global user bases with varying linguistic backgrounds.

III. System Architecture and Evaluation Methodology

A. System Architecture

The proposed architecture for VoiceNav integrates both voice and eye-based interaction to achieve a fully hands-free web navigation experience. The system follows a modular, service-oriented design where each module is responsible for a specific functionality but communicates seamlessly with others to ensure consistency, accuracy, and responsiveness.

At a high level, the system comprises six core components: Voice Input, Speech Recognition, Eye Tracking, AI Processing, Action Execution, and Browser Interface. Figure 1 illustrates the overall structure and data flow.

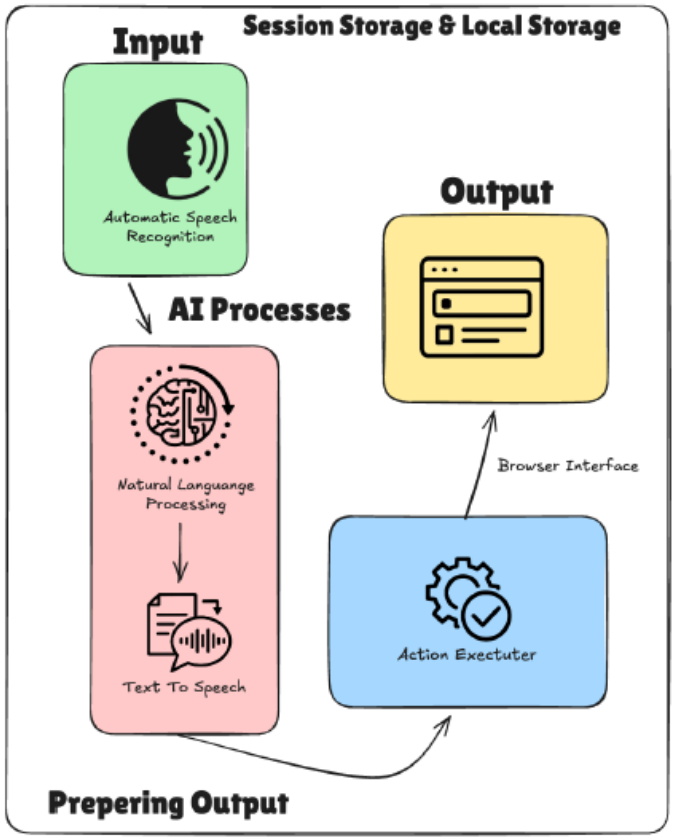

Figure 1. High-level architecture of the VoiceNav system showing the integration of voice

The voice input component captures the user's voice commands through the browser microphone interface. Using the Web Speech API, it continuously listens and converts spoken instructions into text. The system maintains a configurable confidence threshold (set to 0.3) to minimize misrecognition, automatically restarting recognition in noisy environments.

The AI processing module interprets both the text output from speech recognition. It uses natural language processing to determine user intent and to produce a specific action command. The module includes the following subcomponents:

- Intent Parser: Interprets command semantics (e.g., navigate, click, type).
- Context Resolver: Matches identified intent with elements or cross-checked elements in the DOM.
- Fallback Mechanism: Handles cases when voice input is unreliable, defaulting to the best available modality.

The action execution module receives the interpreted command and performs the corresponding operation directly on the web page. It interacts with the DOM to execute actions, such as clicking buttons, scrolling, typing input, or triggering events.

The interface feedback module provides visual or auditory feedback upon successful command recognition or action execution. Examples include highlighting the element being acted upon, confirming actions via voice response, or displaying temporary overlays.

B. Evaluation Methodology

The evaluation methodology for the VoiceNav system was designed to examine the effectiveness, robustness, and usability of hands-free, voice-first web navigation in realistic interaction scenarios. The methodology builds upon established evaluation practices in voice-driven browsing, multimodal interaction, and AI-powered web agents [3]–[5], emphasizing end-to-end task performance rather than isolated component accuracy.

1) Experimental Design: VoiceNav was evaluated using a task-based experimental design, in which the system was required to complete predefined browsing tasks on real, publicly accessible websites. This approach reflects common evaluation strategies used in prior studies on voice-controlled browsing systems and autonomous web agents [3][5], ensuring ecological validity and relevance to real-world use cases. The evaluation focused on voice-first interaction, meaning that all commands were issued verbally without reliance on traditional input devices, such as keyboards or mice. This constraint ensures that observed performance accurately represents hands-free usage conditions, which are central to the system's accessibility and HCI goals [4].

2) Task Selection and Websites: A set of representative browsing tasks was selected to cover a broad range of interaction types, including:

- Page navigation and scrolling
- Link and button selection
- Menu interaction
- Limited form interaction (e.g., entering text into input fields)

Tasks were executed across multiple websites with varying structural complexity, including content-oriented pages and form-heavy interfaces. This selection strategy allows performance comparison across different web layouts and interaction demands, a factor known to significantly influence voice-based system accuracy [3].

3) System Configuration: The VoiceNav system was deployed as a Chrome browser extension, leveraging the Web Speech API for speech recognition [1] and browser-level access to the DOM for action execution. Default system parameters were used throughout testing, including a fixed speech recognition confidence threshold and standard language settings. No user-specific calibration, adaptation, or personalization was applied, ensuring consistency across trials and enabling fair comparison across websites.

Speech input is processed in real time, and commands are interpreted using the system's AI processing module, which maps transcribed text to browser actions. This configuration mirrors the operational conditions of contemporary voice browsing systems and allows direct comparison with existing approaches described in the literature [3][4].

4) Evaluation Procedure: For each website, a sequence of tasks was executed sequentially. A task was considered successful if the system completed the intended action correctly without requiring repetition, clarification, or manual intervention. Failed tasks included misinterpreted commands, incorrect element selection, or incomplete execution.

To capture performance variability, task success rates were calculated per website, rather than aggregated across all tasks. This granular analysis enables identification of specific interaction contexts where performance degrades, particularly in scenarios involving structured data entry or complex page layouts [2][3].

5) Performance Metrics: The primary evaluation metric was task success rate, expressed as a percentage of correctly executed commands. This metric is widely used in evaluating voice-based and AI-driven web interaction systems, as it directly reflects practical usability [3]–[5].

In addition to the overall success rate, qualitative observations were recorded during testing to identify recurring error patterns, such as transcription errors, ambiguous command interpretation, and failures in form validation. These observations provide contextual insight into system limitations that are not fully captured by quantitative metrics alone [4].

6) Methodological Limitations: The methodology intentionally prioritizes system-level feasibility over large-scale statistical validation. As such, the evaluation does not include extensive user studies or long-term adaptation analysis. However, this approach is consistent with exploratory evaluations of emerging multimodal interaction systems and early-stage AI-assisted browsing frameworks [3]–[5].

Overall, this methodology provides a structured and realistic assessment of VoiceNav's current capabilities while clearly identifying directions for future refinement, including improved handling of structured input and enhanced error recovery strategies enabled by large language models [3].

IV. Prototype

In this section, the first version of the VoiceNav user interface is shown. These images (Figures 2, 3, 4, 5) give an idea of how users will interact with VoiceNav. This product is built as a Chrome extension. When loaded, the user sees the main bar showing that the project is ready (Figure 2). In the menu (Figure 3), users can switch languages (for example, English or Italian), mute the microphone, and choose whether the system is listening for commands. The green icon can be moved around the screen so users can place it where it is easiest to use. This makes the system flexible for everyone.

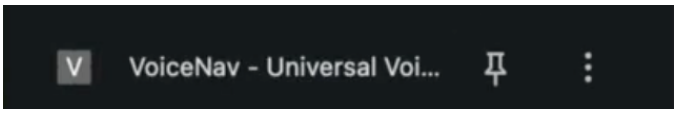

Figure 2. Extension loaded and ready to use (VoiceNav bar)

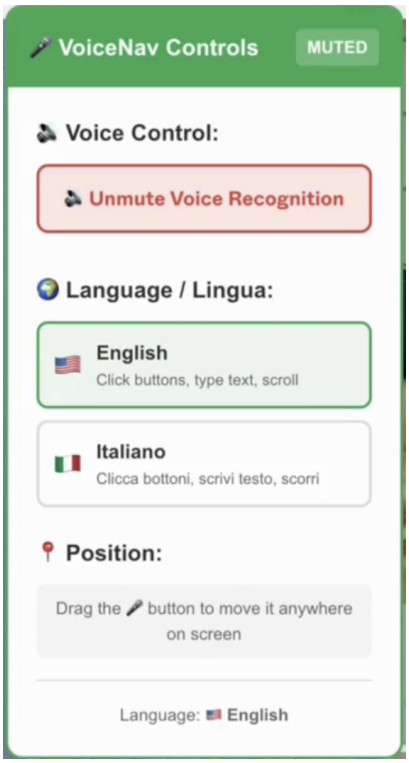

Figure 3. Menu interface lets users switch language and control the microphone. Icon can be moved anywhere

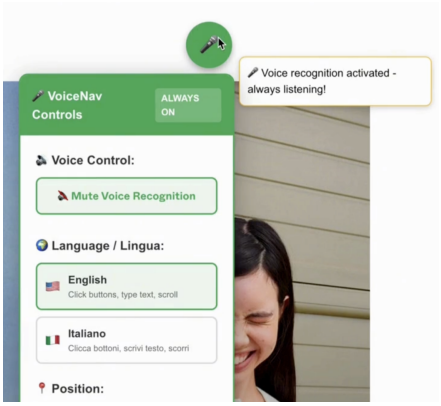

Figure 4. Voice recognition is always listening for commands and gives a quick response

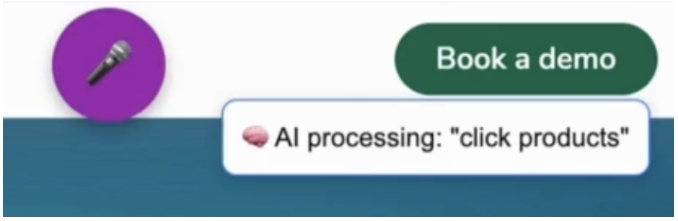

Figure 5. AI processes spoken commands and passes them as instructions to the command parser

When voice recognition is active, VoiceNav listens for spoken commands. A pop-up appears showing ”Voice recognition activated – always listening!”, letting users know they can speak and the system will react quickly (Figure 4). When the user says a command like ”click products,” an AI processes the instruction and shows a reminder that it is working on the user’s request (Figure 5). The spoken command is translated from speech to text, ensuring accuracy so actions match what users want.

V. Discussion | Evaluation

Scenario: Users want a drop-in paragraph added to their existing text that explicitly links the lower checkbox/radio accuracy to form-control ambiguity and UI implementation differences.

Initial testing shows strong performance on general navigation and clicking tasks, with an overall success rate of approximately 80% as illustrated in Figure 6. The figure summarizes command execution accuracy across the evaluated interaction categories and highlights the robustness of voice-driven navigation in typical browsing scenarios. The main weaknesses emerge in form-heavy interactions, particularly when entering email addresses or other structured inputs, where transcription and formatting errors reduce overall precision. These results demonstrate both the current practicality of voice-first browsing and the areas where smarter input handling and tighter validation mechanisms would further improve reliability.

Across navigation-centric websites, VoiceNav demonstrated consistent accuracy when executing commands, such as scrolling, opening links, navigating menus, and selecting visible interface elements, as reflected by the high success rates shown in Figure 6. These findings are consistent with prior research indicating that voice-driven interaction performs reliably when commands can be directly mapped to discrete DOM elements.

Performance degradation was observed on websites that require structured form input, particularly for email fields and text fields that enforce strict validation rules. As shown in Figure 6, these scenarios exhibit noticeably lower accuracy compared to navigation tasks. In such cases, minor transcription errors introduced by the speech recognition component resulted in failed submissions or incorrect input. This limitation reflects a broader challenge identified in voice-based interaction systems, where free-form speech must be translated into highly constrained input formats.

Form controls also contributed to reduced reliability, especially where the system had to resolve a spoken label to a specific option within a group. Figure 6 shows lower success rates for radio-button selection (60%) and checkbox selection (40%) compared to navigation tasks, indicating that VoiceNav is less accurate on state-based inputs than on discrete click targets. This reduction is partly explained by ambiguity in user phrasing (e.g., “enable/disable,” “tick/untick,” “the second option”), closely named alternatives, and inconsistent DOM/accessibility labeling on custom-styled inputs, all of which make correct grounding and state toggling harder than activating a single button or link.

Despite these limitations, the overall results indicate that voice-first browsing is already viable for a wide range of everyday web interactions. The contrast between high navigation accuracy and reduced performance on structured inputs, visible in Figure 6, underscores the importance of integrating intelligent input correction, validation-aware formatting, and iterative error-recovery mechanisms capabilities increasingly supported by large language model-based architectures.

VI. Conclusion and Future Work

This study demonstrates that voice-based hands-free web browsing is a practical and valuable step toward making everyday web access more inclusive for users who cannot reliably use a mouse and keyboard, as well as for hands-busy situations. The proposed approach emphasizes a modular, browser-native design in which continuous speech recognition, natural-language command parsing, DOM-based element discovery, and action execution operate as separate components that work together to deliver end-to-end voice control on arbitrary websites.

The main proposition supported by the prototype work is that voice control can be implemented effectively using standard browser technologies, provided the system includes robust intent parsing and reliable grounding of commands to page elements. This early voice-based system is evaluated using 40 test cases, with 10 per category across a single website. We also manually evaluated the system with 10 participants across mixed tasks, including tables, navigation, duplicate controls, and pop-up scenarios. Early prototype results indicate strong performance on common navigation operations (e.g., opening pages, clicking links/buttons, scrolling), while highlighting that form-heavy interactions and structured text entry remain the most error-prone areas due to transcription and formatting challenges. These findings suggest the system is already suitable for basic browsing assistance, with clear, well-scoped engineering work needed to improve reliability in noisy environments and to strengthen text-entry mechanisms (e.g., spelling modes, confirmation steps, and validation).

From a practical standpoint, the work implies that voice-first browsing can reduce friction for accessibility use cases without requiring site-specific configuration, which is critical for real-world adoption across diverse websites. To be dependable in daily use, the system must also prioritize user trust: clear feedback, safe error recovery, and transparent confirmations when actions are ambiguous are essential to prevent unintended clicks or submissions. Future development should focus on

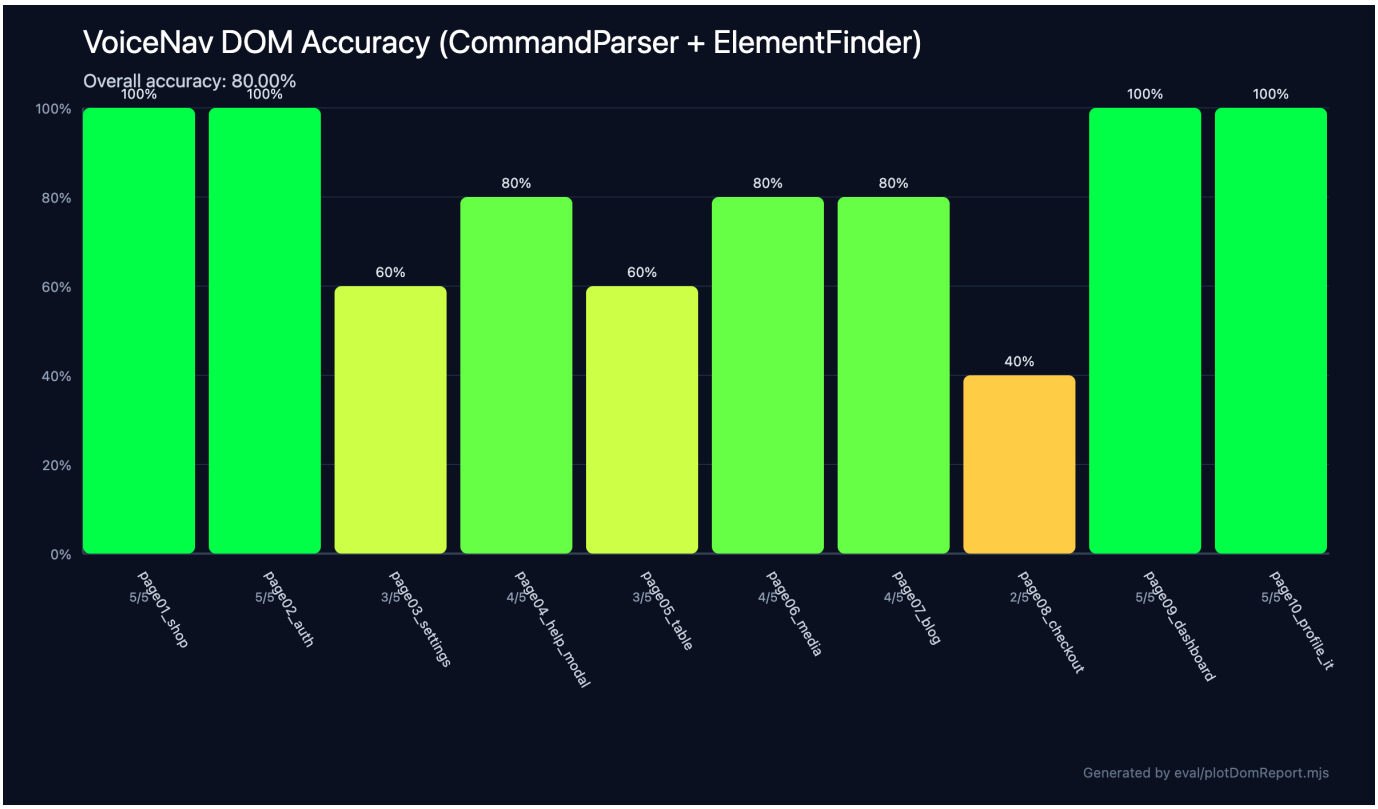

Figure 6. VoiceNav DOM command execution accuracy across evaluated interaction categories

expanding the command set, improving robustness and latency, and running structured user evaluations to quantify task success rates and usability across a wider range of websites and environmental conditions.

References

[1] J. Adorf, "Web Speech API", KTH Royal Institute of Technology, Technical Report, 2013.

[2] D. Prabhu, P. Jyothi, S. Ganapathy, and V. Unni, "Accented Speech Recognition With Accent-specific Codebooks", in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds., Singapore, Dec. 2023, pp.7175–7188. DOI: 10.18653/v1/2023.emnlp-main.444.

[3] S. Mehendale and A. Walishetti, "DexAssist: A Voice-Enabled Dual-LLM Framework for Accessible Web Navigation", in *Intelligent Human Computer Interaction: 16th International Conference, IHCI 2024*, D. Singh, J.-W. Van't Klooster, and U. S. Tiwary, Eds., Twente, The Netherlands: Springer-Verlag, May 2024, pp.171–177. DOI: 10.1007/978-3-031-88881-6_14.

[4] J. Cambre et al., "Firefox Voice: An Open and Extensible Voice Assistant Built Upon the Web", in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA: Association for Computing Machinery, 2021, pp.250:1–250:18. DOI: 10.1145/3411764.3445409.

[5] H. He et al., "WebVoyager: Building an End-to-End Web Agent with Large Multimodal Models", in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins, and V. Srikumar, Eds., Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp.6864–6890. DOI: 10.18653/v1/2024.acl-long.371.