A Preliminary Evaluation of Generative AI Tools for Blind Users: Usability and Screen Reader Interaction

Barbara Leporini
University of Pisa
Pisa, Italy
ISTI-CNR
Pisa, Italy
barbara.leporini@unipi.it

Marina Buzzi
IIT-CNR
Pisa, Italy
Marina.Buzzi@iit.cnr.it

Giuseppe Della Penna
Università dell'Aquila
L'Aquila, Italy
Giuseppe.DellaPenna@univaq.it

Abstract

The increasing use of Generative Artificial Intelligence (GAI) tools such as ChatGPT, Copilot, Perplexity and Gemini opens up new possible scenarios for supporting work and everyday activities. For people who are blind, the usability of such tools through screen readers is crucial to ensure their use of such AI-based technologies. In this study, we explore the accessibility and usability of the interfaces of four popular AI-based tools via screen readers through a combination of semi-automated evaluations and inspections conducted by both sighted and blind accessibility experts and screen readers with more than 20 years of experience. Navigation, labeling of control elements, feedback mechanisms, and prompt handling were considered in the study. The results point to usability difficulties in all tools, particularly in navigation structure, clarity of feedback and interactive elements. Although this work empirically explores the accessibility of AI-based tools it brings out the first critical issues that deserve further investigation. However, they are based on a small group of experts and thus should be considered preliminary and useful for future studies.

CCS Concepts

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

Keywords

Blind users, Generative AI, ChatGPT, Gemini, Copilot, Perplexity, Accessibility, screen reader interaction

ACM Reference Format:
Barbara Leporini, Marina Buzzi, and Giuseppe Della Penna. 2025. A Preliminary Evaluation of Generative AI Tools for Blind Users: Usability and Screen Reader Interaction. In *The PErvasive Technologies Related to Assistive Environments (PETRA '25), June 25–27, 2025, Corfu Island, Greece*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3733155.3737910

1 Introduction

In the era of the Internet society, technology has become an essential component for assisting people with disabilities in every field, including study, work, public life, social relations and leisure time. For people with visual impairments, especially blind individuals, having accessible information is a priority.

Graphical interfaces are mainly designed for interaction via mouse with point-and-click and drag-and-drop operations. However, a blind user usually has to interact with web applications using only the keyboard, i.e., with keyboard shortcuts and *tab* keys for navigation [13].

Another essential assistive technology for blind users is the screen reader, a software layer between the OS (operating system) and the application graphical user interface (GUI) that converts the screen content into voice or braille, with a sequential reading. In particular, when a web page is accessed, a screen reader tries to analyze its content in order to extract and announce active and structured elements such as links, text boxes, buttons, tables, headings, and so on [14]. This, however, implies an additional cognitive load and an additional execution time in performing tasks due to the complexity of the interaction [10], which increases when the web page is more complex and less accessible [5]. This issue becomes particularly critical in work environments, where efficiency and autonomy are essential and where inaccessible interfaces can significantly hinder participation and productivity.

Recently, the rapid evolution of AI, and in particular the progress of Large Language Models (LLM), has offered new possibilities for assisting vulnerable people in everyday activities and, in particular, to deliver support when they are studying or working. Generative AI (GAI) exploits models to produce original content combining different sources, infers knowledge in a process miming the brain connections and can deliver answers to users on different channels such as voice or text.

A big concern for people with disabilities is increasing their autonomy, being able to perform tasks autonomously, and having a job to be independent and fully integrated into society [13]. Actually, GAI systems (GAIs) are becoming essential also in many jobs to speed up activities. However, to be used at work by screen reading users, GAI tools must be easy to use and accessible to provide efficient, fast, and accurate answers that can be read, copied and modified according to the needs of the workers.

In this study, we investigate the accessibility and usability of four popular Generative AI tools: ChatGPT (https://chatgpt.com/), Copilot (https://copilot.microsoft.com/), Perplexity (https://www.perplexity.ai/) and Gemini (https://gemini.google.com/), when interacting via screen reader. Our goal is to assess whether these tools are effectively usable by blind users and to identify potential

PETRA '25, Corfu Island, Greece

ACM ISBN 979-8-4007-1402-3/2025/06
https://doi.org/10.1145/3733155.3737910

accessibility issues that may require improvements. Specifically, we aim to answer two research questions:

RQ1: To what extent are popular Generative AI tools (ChatGPT, Copilot, Perplexity, Gemini) accessible and usable with a screen reader?

RQ2: What are some of the main accessibility challenges that blind users may encounter when interacting with these tools, and is there a need for usability improvements?

Given the exploratory nature of this work, we intend to highlight interaction challenges that may hinder usability for screen reader users and to offer initial insights for further investigation. Results indicate that although these tools are potentially useful for supporting users in everyday tasks, accessibility is still limited in some key aspects across all four platforms.

The paper is organized as follows: after this introduction, Section 2 introduces and discusses the related work; the applied method is described in Section 3, while sections 4 and 5 present the study and its results, including the accessibility inspection and screen reader interaction. Lastly, the conclusions section ends the paper.

2 Related Work

The recent explosion of AI-based tools and applications has significantly enhanced the reasoning power of machines. Large Language Models (LLMs) apply deep learning techniques (transformer architecture) to learn and understand complex patterns and structures of language data. LLMs have the ability to process vast amounts of data, unstructured text, visual, audio, multi-modal data and infer semantic relationships between them [9].

Research focuses on an exponential trend in LLM potential, in every life sector, including accessibility for people with disabilities. Recent studies reveal the undiscovered potential of generative AI in the accessibility field. LLMs can improve accessibility in several ways: adding textual descriptions to images (captioning), assisting blind users in navigating web pages, and simplifying the textual content of web pages.

Patel et al. (2025) created a system that exploits the potential of LLM for image captioning in different fields that reach an accuracy of 88.7% by processing descriptions of scenes and applying sentiment analysis with high adaptability and contextual awareness [19].

López-Gil & Pereira (2024) investigated the use of LLMs for automating the web accessibility evaluation, and specifically for testing some WCAG success criteria that are currently checked manually. In this way, time and cost are reduced, with the advantage that practitioners, instead of experts, can also be able to carry out this kind of evaluation [17].

Ghosh et al. (2024) use the abilities of large language models to enhance interaction with user interfaces by enabling complex screen reader actions via natural language, in order to reduce the cognitive load and the interaction complexity and improve usability and efficiency. The proposed system automates the actions of the screen reader on control elements like buttons, text boxes, and drop-down menus via vocal commands, interpreting and performing the required action sequence on behalf of the user. Simplifying the interaction via screen reader, the system provides uniform access to different applications, thus reducing the load caused by the heterogeneity in the user interfaces (and the need to learn new user interfaces and web structures) [7].

However, despite the amount of research in GAIs, only a little attention has been devoted to the accessibility of user interfaces of AI based tools such as ChatGPT, Copilot, Gemini, and Perplexity. Only a few recent studies have addressed this topic. Adnin et al. (2024) interviewed 19 blind and visually impaired users, revealing insights on how participants interact with the GAI tools, about what mental models they have built on the AI functioning, and the experienced obstacles, showing that the accessibility in the text-based interaction is still unsatisfactory due to the lack of accessible keyboard navigation, unlabeled buttons and a poor design [2]. Moreover, Acosta-Vargas et al. (2024) investigated mental models of Generative AI tools from the perspectives of blind people and discovered the multiple ways in which blind people exploit GAI for content creation and information retrieval, often working around various accessibility and usability issues in GAI interfaces [1].

In this paper, starting from previous studies, we continue to investigate this very important research field to make designers aware of the importance of following accessibility principles from the early design phases, to deliver a satisfying user experience for all, including screen reader users. Specifically, we analyze and comment on existing guidelines to guide developers toward good design practices.

3 The Method

This study investigates the usability of generative AI tools for screen reader users performing an evaluation of the user interaction. First, we conducted a preliminary analysis of the accessibility of generative AI tool interfaces using semi-automated tools. However, our study is primarily focused on interaction aspects during actual GAI system usage. Therefore, we applied an inspection-based evaluation to assess key requirements and aspects relevant to screen reader navigation.

Blind users experience severe difficulties when interacting with web pages and spend much more time accessing information than sighted people [12]. A fundamental need for blind users is to quickly reach the information relevant to the task by filtering redundant and irrelevant content [8]. The W3C guidelines primarily focus on accessibility aspects, although more recent releases have started to incorporate usability considerations as well. However, we refer to studies that more specifically address usability in interaction with screen readers.

The literature presents numerous guidelines, principles, and criteria for designing user interfaces that are accessible and usable via screen readers. Among these, we selected those most applicable to GAI interfaces. Leporini et al. (2008) proposed nine guidelines for search engines to ensure their user interfaces are easily navigable with screen readers [15]. Additionally, Leporini and Paternò (2008) introduced a set of more general guidelines aimed at improving screen reader interaction in terms of effectiveness, efficiency, and user satisfaction. These guidelines are primarily focused on how to perceive the structure of the interface, how to easily reach the purpose or content of the page, and how to be efficient in doing so [16]. We have thus drawn inspiration from those that best fit or relate to our study. We also followed the methodology analyzed in

[18], which observed that "multiple evaluators, working independently, performed better than individuals. They are most reliable when reviewing sites using a screen reader. Additionally, we found that the use of a screen reader significantly increased the validity (reduced the number of false positives reported) of individual evaluators", by applying a synergic analysis by experts (manual and automated accessibility inspection) and blind persons interacting via screen reader.

The evaluation was conducted by the authors of this study, all experts in accessibility. One of them brings direct experience as a totally blind user. In addition to the blind co-author, two additional blind experts contributed to the evaluation. All blind participants have more than 20 years of experience using screen readers on Windows systems with JAWS for Windows [https://www.freedomscientific.com/products/software/jaws/]. One of the additional blind experts works as an accessibility consultant, while the other holds an official role as an accessibility expert within a public institution. Their long-standing expertise and practical knowledge of screen reader interaction provided valuable insights into the usability of GAI tools. While this expert-based approach offers qualitative findings, the results should not be considered representative of the broader population of blind users. Rather, this evaluation represents an exploratory contribution to detect interaction challenges and promote inclusive design in GAI tools.

Each expert performed the evaluation independently, after which the results were compared. The two sighted experts assessed the accessibility of the interfaces and analyzed the UI code to verify the implementation of the selected guidelines. The analysis was carried out by accessing the main page of each GAI tool as a registered and logged-in user. Each expert documented the success or failure of each guideline and recorded his/her impressions.

4 The Study

In this section, we describe the evaluation conducted in our study. First, we present the accessibility assessment performed using semi-automated tools and by analyzing the code by hand. Then, in the following section, we report on the study conducted using a screen reader.

4.1 The Accessibility Assessment

It is well known that assistive tools such as screen readers perform better on pages designed with accessibility in mind. This means that the page source code should be well-structured and follow as closely as possible the WAI-ARIA standard [29] and, in general, the Web Content Accessibility Guidelines (WCAG [21]).

Therefore, to assess the general accessibility of the GAI websites, before trying to access them with screen readers and other accessibility aids, we performed some preliminary analyses on their main pages, running both the Lighthouse [11] and the WAVE [4] tools and manually inspecting part of the pages' source code.

Unfortunately, despite their simple appearance, all the pages are heavily script-based, so the automatic tools are often fooled by the complex and dynamic page structure. Anyway, Lighthouse gave all the analyzed pages accessibility scores higher than 90%, so tagging them as highly accessible pages. Some minor issues were raised on each of them, like contrast or missing accessible names, but most of such warnings were clearly biased by the page structure (e.g., referred elements were visually hidden).

On the other hand, the WAVE tool identified missing form control labels on all the GAI pages except Gemini. ChatGPT and Perplexity also appear to have buttons with graphical (SVG) content and no textual alternatives. All the pages make a minimal use of the HTML5 structure and heading elements (so making the page overall organization difficult to deduce for many accessibility tools): some of them wrap the content in a *main* tag, but no other tags such as *<SECTION>*, <ARTICLE> or <NAV> are used. Title structure tags (*<H1>*...*<H6>*) are present in some cases, but used incorrectly (e.g., *<H2>* without a previous *<H1>*). Therefore, in general, the structure of the analyzed pages appears complex to understand from their source code, which may be an obstacle, e.g., also for screen readers. However, the WAVE tool confirmed a wide, correct use of the accessibility (ARIA) attributes on all the pages, which in many cases mitigates the absence of the above HTML5 elements and structures, e.g., by specifying the *role* on general purpose tags such as *div*.

A quick manual inspection of the source code analyzed by the tools confirmed most of the detected issues: ARIA attributes are widely used to add accessible information to the tags, which are mostly *<DIV>* containers. Gemini, being based on the Angular framework [6], also contains some custom markup. In some cases, there is clear evidence of screen reader-targeted structures and page elements, e.g., tags with class "*sr-only*" used in the ChatGPT, Copilot and Perplexity code.

Finally, for what concerns keyboard navigation, all the interface elements (menus, buttons, input fields) are reachable by *tab navigation* (also thanks to *tabindex* attributes correctly placed where needed) and activable using *enter* key, making the GAI pages usable also without a mouse.

4.2 Screen Reader Interaction

4.2.1 Interaction Objectives for GAI Tools. In our study, we have considered three interaction objectives in the context of screen reader usability, ensuring that users relying on assistive technologies can effectively achieve these goals:

1. Easily compose and edit the prompt

The input field for the prompt should be easily identifiable and positioned at the top of the page structure to ensure quick access. The layout should follow a logical arrangement of input fields, with clear labels and an intuitive placement of the text box and related options. Keyboard shortcuts should be assigned to enable users to quickly access and modify the prompt without unnecessary interactions.

2. Quickly access the response

The generated response should be clearly announced and highlighted within the interface, using headings or a dedicated section to differentiate it from other elements. The response should be structured in a readable and organized format, preferably in a numbered list or distinct sections to facilitate user scanning. Immediate feedback should be provided when the response is available or when no results are found, using visual or auditory indicators to notify the user.

3. Navigate and interact with the results efficiently

Table 1: Usability Guidelines

Guidelines1	Guidelines2
Place the most important elements of the interface at the top of the source file	Logical partition of interface elements
Navigating faster	Assignment of shortcuts
Easy location and labelling of edit field and search options	Proper form layout
Highlighting the search result	Importance levels of elements
Arranging the results in numbered lists	Specific sections
	Messages and dynamic data management

Navigation and help links should be strategically placed, such as shortcuts to quickly return to the input field or to navigate between multiple responses. Copying the response should be simple, with a visible copy button next to the response text or an easily accessible keyboard option.

By incorporating these objectives into our study, we aim to assess how well users with screen readers can successfully perform these tasks and navigate AI-generated content effectively.

4.2.2 Evaluation Design. Guidelines in literature and screen reader navigation strategies have guided the design of the evaluation. Works that indicate how the user interface should be organized to enhance usability, as well as insights into navigation strategies used by screen reader users, suggest a sort of framework for conducting the evaluation. For example, there are studies highlighting how the correct use of headings improves interaction with screen readers. Headings structure content hierarchically, allowing screen reader users to quickly navigate between sections of a page [3]. Of the nine guidelines suggested by Leporini et al. (2008), six were selected as they are most relevant to the type of interaction for GAI tools (Guidelines 1) and listed in the first column of Table 1. Analogously, column 2 lists the most relevant guidelines from those proposed by Leporini and Paternò (2008) (Guidelines2). In Table 1, the guidelines are paired based on their similarity.

Inspired by the proposed objectives in the framework of the above guidelines, the following tasks were defined for the GAIs evaluation:

1. Write a free prompt. This task is designed to assess whether the user can easily identify and access the editing field.

2. Read the response and request clarification or additional details. This task is designed to assess whether the user can effectively read the response and interact with it by requesting clarifications or additional details.

3. Navigate the responses and copy one of them for use in another document. This task is designed to assess whether the user can efficiently navigate the responses and copy to transfer one (or more) of them for use in another document.

All experts completed the three tasks and recorded:

(a) whether mechanisms at the code level were used to facilitate interaction with the interface to meet the principle;

(b) their own comments regarding that specific task and/or principle.

For example, in relation to point (a), the experts noted whether specific WAI-ARIA commands were implemented or if basic HTML tags such as <H1>, <H2>, and so on, were properly used. Regarding point (b), they documented screen reader behaviors, or any workarounds required (e.g., techniques used by experienced users to navigate more efficiently).

5 Results

This section presents and discusses the evaluation results organized according to the three interaction objectives.

5.1 Easily compose and edit the prompt

Regarding the first objective, the focus is primarily on correctly identifying the search field and being able to formulate the question or request to the tool, specifically by entering the prompt (task 1). To achieve this, users accessed the interface and searched for the input field using the tab key or specific screen reader commands. Additionally, they explored the availability of specific keyboard shortcuts to enable faster access to the input field. Evaluators also checked whether the search field was properly labeled and whether it was clear that this specific field was intended for entering the question.

ChatGPT. Upon page load, the focus is already positioned on the text input field, ready to be edited. The message ”Ask a question” is announced only if the user employs the ”Say line” SR command; otherwise, the screen reader simply announces ”Edit.”. No keyboard shortcuts are available to reach the input field with a single key combination. The button used to upload a file is detected twice in virtual cursor mode: first as an unlabeled graphical button and then as a properly labeled button. When the system cursor is on (i.e. the virtual cursor is disabled), navigating through the buttons, the first one is read as ”Menu,” while the second is read as ”File,” which can be disorienting for users. If the user leaves the input field to explore the rest of the interface, returning to edit a question requires knowledge of a specific screen reader command (e.g., pressing the letter ”e”). Overall, users find it manageable.

Microsoft Copilot. After activating Copilot via the shortcut, the focus is directly placed on the input field, and the label is announced as ”Send a message to Copilot.”. No quick access keys were detected. If the virtual cursor is disabled for page exploration, users need to rely on a screen reader command, as in the case of ChatGPT, to return to the input field. Overall, the tool is usable, though Copilot integrates with the current webpage as a side panel. This makes the interface more complex. It would be preferable if Copilot operated within a separate page or tab.

Google Gemini. Upon page load, the focus is correctly placed, and the message ”Enter a prompt here” is announced. However, the term ”prompt” may be unfamiliar to less experienced users. There are no keyboard shortcuts to quickly reach the input field, so using a screen reader command to navigate to the text box is necessary. Overall, the tool is usable, but if users move away from the text box to explore responses or navigate the interface, relocating the input field to enter additional questions can be somewhat challenging.

Perplexity. Upon page load, the focus is correctly positioned, and the message "Ask anything" is announced. No keyboard shortcuts appear to be available for quickly returning to the input field for further editing. As in previous cases, users need to use the screen reader command "e" to reach the input field again. After receiving an initial response, the focus remains on the input field, but the announced message changes to "Ask a follow-up," which may be unclear to users. Following the input field, the button to upload a resource is correctly detected. However, an unlabeled button is also identified by the screen reader but appears non-functional. Overall, the tool is usable, but, as with the other tools, relocating the input field after exploring the interface can be more challenging for less experienced users.

5.2 Quickly access the response

After entering a question (prompt), the next step involves evaluating how the screen reader interacts with the system's response. Specifically, this task focuses on the feedback provided to the user while the tool prepares an answer and the automatic or manual reading of the response.

ChatGPT. While the content is being generated, the screen reader announces a message ("ChatGPT is generating a response") to inform that the system is working; this is a very useful feedback for the user. Once the response is complete, it is automatically read by the screen reader without requiring any user commands. However, if the user wants to read again the response (either in block, line by line, or word by word), it is needed to manually navigate back to it via several keyboard commands using the screen reader's virtual cursor mode. In doing this, several links and buttons are encountered before reaching the text of the response, which can slow down navigation.

Microsoft Copilot. The response is automatically read by the screen reader as soon as it is generated, but no feedback is provided during the response generation process. So, if this requires a lot of time, the user is not aware if the system is working or not. Reaching the response to read it manually (line by line, word by word, etc.) requires several steps by exploring via screen reader commands. This can be too difficult or cumbersome for less skilled users.

Google Gemini. While generating the response Gemini gives feedback to the screen reader which announces "Gemini is typing". Unfortunately, the screen reader does not automatically read the complete generated response. Instead, it announces "Gemini has responded," which informs the user that the generation is finished, but this does not facilitate immediate reading. As a result, the user has to manually locate the answer using numerous navigation commands.

Perplexity. Perplexity occasionally provides feedback during response generation by giving the message "Pending," but this is not always consistent or reliable. The generated response is not read automatically, requiring the user to manually navigate through many steps via screen reader commands to read the content.

5.3 Navigate and interact with the results efficiently

After interacting with the GAIs through multiple prompts, this task evaluates how easily users can navigate between different questions-responses. The goal is to analyze whether conversations are accessible and structured in a way that allows smooth navigation especially via keyboard. Additionally, the task examines how easily a single response can be copied for use in another document.

ChatGPT. The interface allows users to jump between different questions-responses using heading levels (<h6>) applied to each entry in the conversational dialogue. These headings are labeled as "You said", "ChatGPT said", and so on. This structure makes navigation possible, but not particularly efficient. More meaningful and structured headings would improve usability. Regarding copying responses, each answer includes a "Copy" button. However, accessing this button requires sequentially navigating through the response text or using a specific screen reader command (e.g., pressing "b" to jump between buttons). Additionally, the "Copy" button lacks contextual information, making it unclear which response is being copied.

Microsoft Copilot. Copilot behaves similarly to ChatGPT, using headings such as "You said" and "Copilot said" to structure the conversation. The "Copy message" button also lacks contextual information, requiring users to explore surrounding content to determine which message they are copying.

Google Gemini. Gemini provides <h3> headings to navigate between different questions, with each heading directly linked to the question text. This makes jumping between questions easy and clear thanks to meaningful labeling. However, copying responses is even more cumbersome than in other tools. The "Copy" button is hidden inside the "More options" menu, requiring additional steps to locate and use it.

Perplexity. Perplexity does not implement any specific mechanisms to facilitate navigation between responses. Instead, each response is preceded by the word "Response", but without any structured element (such as a heading) to assist screen readers in identifying it. As a workaround, users must search for the presence of the word "Response" manually and repeat the search until they find the desired one. When positioned on a "Response" label, users can read the answer content by navigating with the arrow keys. This particular technique was suggested by one of the screen reader experts but is not intuitive for less skilled users. Furthermore, responses are followed or preceded by numerous source links, making navigation particularly cumbersome. Like other tools, Perplexity provides a "Copy" button, but it lacks contextual information, requiring additional effort to determine which response is being copied.

5.4 Discussion

Results of our analysis showed that the screen reader users encounter different levels of difficulty in detecting and interacting with the input fields of the analyzed AI-based systems. When uploading a page, the focus is well-positioned, but there are no keyboard shortcuts to quickly jump to the input fields, thus the users have to use specific screen reader commands to perform this action. In addition, unlabeled buttons as well as unclear text can create confusion in less experienced users. ChatGPT and Perplexity have duplicated or unclear button labels, while the side-panel integration of Copilot makes the navigation more complex. Google Gemini has clear labels but uses technical terminology ("prompt") not intuitive for any users. Across all systems, reaching the input field after navigating through responses remains a recurring difficulty, potentially impacting efficiency.

Overall, all the analyzed tools are generally usable, but improvements in keyboard accessibility, clearer labeling, and better navigation options would enhance the user experience for screen reader users.

Analyzing the feedback of the response generation, ChatGPT and Gemini provide clear status updates, Copilot lacks real-time feedback, and the Perplexity's feedback is inconsistent. ChatGPT and Copilot read responses immediately, while Gemini and Perplexity require users to detect and read them manually.

In all tools, navigating back to the response via screen reader requires many steps and keyboard commands to reach it, making the process difficult for inexperienced users. ChatGPT's interface presents additional navigation obstacles due to links and buttons encountered, navigating via Tab, before reaching the response.

Improving keyboard navigation with shortcuts and adding clear and context-independent labels would enhance the user experience of blind users interacting via screen reader. ChatGPT and Gemini provide clear feedback for status updates, Perplexity is inconsistent and Copilot do not provide feedback. More, ChatGPT and Copilot read responses immediately as soon they are generated, while Gemini and Perplexity require users to detect and read the generated text manually.

In all tools, navigating back to the response via screen reader requires many steps and keyboard commands to reach it, making the process difficult for inexperienced users. ChatGPT interface presents additional navigation obstacles due to links and buttons encountered navigating via Tab, before reaching the response.

Structuring text by appropriate heading levels (<H1>, <H2>, etc.) is a technique that makes navigation via screen reader faster. ChatGPT, Copilot, and Gemini apply <H2> headings to structure the conversations. Gemini directly links headings to the question text, making navigation clearer. Perplexity do not offer structured navigation, forcing users to rely on manual searches, which are inefficient and difficult for novice users.

To copy the responses, all tools provide a "Copy" button, but none of them offer a clear contextual label. Users must either navigate sequentially through the text or explore the surroundings to determine which response they are copying. Gemini makes copying challenging since the copy button is in a "More options" menu.

To improve interaction, GAIs should (a) enhance heading structures to provide more meaningful navigation; (b) ensure that "Copy" buttons include contextual labels to indicate which response is being copied; and (c) reduce unnecessary navigation steps (e.g., by avoiding excess links in Perplexity or simplifying the copy process in Gemini).

6 Conclusions

Generative AI systems, such as ChatGPT, Copilot, Gemini, and Perplexity are valuable tools for supporting people in several tasks, including work. For users who rely on screen readers, AI-based tools have the potential to significantly enhance autonomy and productivity. However, their effectiveness depends on the user experience, which may vary depending on assistive technology used. Thus, if not properly designed, these tools could become a barrier instead of being a valuable resource for the users.

This study evaluates the accessibility and usability via screen readers of four popular generative AI tools analyzing the user interaction for blind users. Ensuring that these tools are easy to use is crucial to making them a valuable, simple, and efficient resource. Our analysis aims to highlight potential issues and propose suggestions for improvements that can greatly enhance inclusivity and effectiveness.

Our analysis suggests that the accessibility and usability of popular Generative AI tools via screen reader can vary significantly, requiring a considerable effort to ensure an effective interaction (RQ1). While all four tools provide a basic level of accessibility, unclear labels, the lack of shortcuts for fast navigation, and difficult response retrieval can be challenging for blind users. Among the main accessibility barriers identified (RQ2), navigation inefficiencies, insufficient feedback mechanisms, and non-contextualized interactive elements are significant obstacles.

The analysis performed indicates possible improvements: enhancing user experience by structuring navigation, improving the labels for interactive elements (such as copy buttons), and integrating shortcuts to support the interaction. This study has some limitations. First of all, these results are based on a small number of expert evaluations and should be interpreted as exploratory. More, the evaluation is conducted by a few experts, including blind and sighted participants. Thus, results do not reflect the diversity of a broader population of blind users, with different expertise and experience. The empirical evaluation is only based on inspection and expert evaluation, without any quantitative performance measures or user testing. Thus, the findings should be considered exploratory and indicative. Future work will address these limitations by involving a wider sample of users.

References

[1] Acosta-Vargas, P., Salvador-Acosta, B., Novillo-Villegas, S., Sarantis, D., & Salvador-Ullauri, L. (2024). Generative Artificial Intelligence and Web Accessibility: Towards an Inclusive and Sustainable Future. Emerging Science Journal, 8(4), 1602-1621.

[2] Adnin, R., & Das, M. (2024). " I look at it as the king of knowledge": How Blind People Use and Understand Generative AI Tools. In Proceedings of the 26th International ACM SIGACCESS Conference on Computers and Accessibility (pp. 1-14).

[3] Ahmed, F., Borodin, Y., Soviak, A., Islam, M., Ramakrishnan, I. V., & Hedgpeth, T. (2012). Accessible skimming: faster screen reading of web pages. In Proceedings of the 25th annual ACM symposium on User interface software and technology (pp. 367-378).

[4] Bhatia, P. N., & Malek, S. (2024). A historical review of Web Accessibility using WAVE. In Proceedings of the 5th ACM/IEEE Workshop on Gender Equality, Diversity, and Inclusion in Software Engineering (pp. 55-62).

[5] Borodin, Y., Bigham, J. P., Dausch, G., & Ramakrishnan, I. V. (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) (pp. 1-10).

[6] Geetha, G., Mittal, M., Prasad, K. M., & Ponsam, J. G. (2022). Interpretation and analysis of angular framework. In 2022 International Conference on Power,

Energy, Control and Transmission Systems (ICPECTS) (pp. 1-6). IEEE.

[7] Ghosh, Anujay, Monalika Padma Reddy, Satwik Ram Kodandaram, Utku Uckun, Vikas Ashok, Xiaojun Bi, and I. V. Ramakrishnan. "Screen Reading Enabled by Large Language Models." In Proceedings of the 26th International ACM SIGACCESS *Conference on Computers and Accessibility*, pp. 1-5. 2024.

[8] Giraud, S., Thérouanne, P., & Steiner, D. D. (2018). Web accessibility: Filtering redundant and irrelevant information improves website usability for blind users. International Journal of Human-Computer Studies, 111, 23-35.

[9] Hadi, M. U., Al-Tashi, Q., Qureshi, R., Shah, A., Muneer, A., Irfan, M., ... & Shah12, M. (2025) LLMs: A Comprehensive Survey of Applications, Challenges, Datasets, Models, Limitations, and Future Prospects. TechRxiv. February 10, 2025. DOI: 10.36227/techrxiv.23589741.v8

[10] Harper, S., Michailidou, E., & Stevens, R. (2009). Toward a definition of visual complexity as an implicit measure of cognitive load. *ACM Transactions on Applied Perception (TAP), 6*(2), 1-18.

[11] Heričko, T., Šumak, B., & Brdnik, S. (2021). Towards representative web performance measurements with google lighthouse. In Proceedings of the 2021 7th student computer science research conference (p. 39-42).

[12] Lazar, J., Allen, A., Kleinman, J., & Malarkey, C. (2007). What frustrates screen reader users on the web: A study of 100 blind users. International Journal of human-computer interaction, 22(3), 247-269.

[13] Lee, H. N., Ashok, V., & Ramakrishnan, I. V. (2020). Repurposing visual input modalities for blind users: a case study of word processors. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 2714-2721). IEEE.

[14] Lee, S., Reddie, M., & Carroll, J. M. (2021). Designing for independence for people with visual impairments. *Proceedings of the ACM on Human-Computer Interaction*, *5*(CSCW1), 1-19.

[15] Leporini, B., Andronico, P., Buzzi, M., & Castillo, C. (2008). Evaluating a modified Google user interface via screen reader. *Universal access in the information society,* 7(3), 155-175.

[16] Leporini, B., & Paternò, F. (2008). Applying web usability criteria for vision-impaired users: does it really improve task performance?. *Intl. Journal of Human–Computer Interaction*, *24*(1), 17-47.

[17] López-Gil, J. M., & Pereira, J. (2024). Turning manual web accessibility success criteria into automatic: an LLM-based approach. *Universal Access in the Information Society*, 1-16.

[18] Mankoff, J., Fait, H., & Tran, T. (2005). Is your web page accessible? A comparative study of methods for assessing web page accessibility for the blind. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 41-50).

[19] Patel, V., Modi, A., Mistry, H., Mishra, A., Upadhyay, R., & Shah, A. (2025). From Alt-text to Real Context: Revolutionizing image captioning using the potential of LLM. International Journal of Scientific Research in Computer Science Engineering and Information Technology 11 (1): 379-387

[29] Pratiwie, N. (2023). Chatbots in the Workplace: An Interpretive Phenomenological Study of Workers on the Use of ChatGPT. *International Journal of Social Service and Research, 3*(9), 2296-2305.

[21] W3C. Accessible Rich Internet Applications (WAI-ARIA) 1.2 W3C Recommendation 06 June 2023, https://www.w3.org/TR/wai-aria-1.2/

[22] W3C. Web Content Accessibility Guidelines (WCAG) 2.2. W3C Recommendation 12 December 2024, https://www.w3.org/TR/WCAG22/