# How an LLM Can Improve Automatic Web Accessibility Validation ?

Fabio Paternò
CNR-ISTI, HIIS Laboratory
Pisa, Italy
fabio.paterno@isti.cnr.it

Manuela Vinci
CNR-ISTI, HIIS Laboratory
Pisa, Italy
m.vinci11@studenti.unipi.it

Marco Manca
CNR-ISTI, HIIS Laboratory
Pisa, Italy
marco.manca@isti.cnr.it

Nicola Iannuzzi
CNR-ISTI, HIIS Laboratory
Pisa, Italy
nicola.iannuzzi@isti.cnr.it

## Abstract

Digital accessibility is important since it allows all people, including those with disabilities, to interact and access the desired information available on the Web. The W3C WCAG guidelines provide a rich set of indications about how to obtain it. Over time, they have become very extensive in order to consider the many possible cases. Manual checking of all the corresponding techniques is impossible; thus, interest in the support provided by automatic tools is increasing. However, current validation tools sometimes have several limitations in their analysis, which still require considerable manual intervention to validate several accessibility techniques. Large Language Models (LLMs) present an opportunity to address such cases. In this paper, we report on an investigation that focused on exploiting the functionality made available by the GPT 4o APIs to address such cases. We report on the types of prompting techniques used for this purpose, how they have been exploited, for which accessibility techniques, and how they have been validated. The results provide useful indications for understanding the role of large language models for accessibility validation.

## CCS Concepts

• **Human-centered computing**; • **Accessibility systems and tools;**;

## Keywords

Accessibility validation, Automatic Tools, Large Language Models

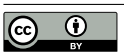
**ACM Reference Format:**
Fabio Paternò, Manuela Vinci, Marco Manca, and Nicola Iannuzzi. 2025. How an LLM Can Improve Automatic Web Accessibility Validation ?. In *CHItaly 2025: 16th Biannual Conference of the Italian SIGCHI Chapter (CHItaly 2025), October 06–10, 2025, Salerno, Italy*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3750069.3750310

## 1 Introduction

The importance of accessibility has been recognised by national and international legislations [3, 8, 9, 17], given the widespread use of digital services in daily life and the high number of people who suffer from permanent or transient disabilities. The W3C WCAG guidelines provide a solid and extended set of concrete indications to obtain accessible Web sites. They have evolved over the years to consider the evolution of interactive libraries and devices and better address the many possible types of disabilities. The resulting success criteria and techniques are rather numerous, and they need to be applied systematically to the enormous amount of content that is available over the Internet. Such validation work cannot be done manually and thus needs the support of automatic tools.

Over time, several types of automatic validation tools have been put forward [1, 4, 26], and they have had to evolve to address the challenges the evolving digital landscape has raised. Since accessibility evaluation is an area continuously evolving, several tools have not been able to keep up with such evolutions: they have become obsolete because they target old versions of the accessibility guidelines, or they are not able to address modern dynamic websites or they are limited in terms of scalability of the number of pages that they can validate. The W3C has recently updated the public list of accessibility validation tools, which, as of May 2025, is composed of 77 tools.

In general, it should be clear that their role is important to support scalability in accessibility validation, but they cannot provide a full, complete evaluation, and their use should be associated with direct user testing [24, 28]. There is also an issue concerning their transparency [14, 19] since often they are unclear about what techniques they can validate and how they validate them. The results of their evaluations are based on a set of algorithms associated with the various techniques, and there is still limited agreement on such algorithms. Indeed, the W3C ACT Rules initiative, which aims to inform accessibility testers on how to evaluate specific cases consistently, only covers a rather small portion of the WCAG guidelines. Consequently, several studies [2, 7, 23] comparing the results of various automatic validation tools have found differences in their results. Despite such problems, the area of automatic accessibility evaluation has evolved, and in recent years, several proposals have been put forward of tools able to address large-scale evaluations both at the commercial level (for example Siteimprove, Deque) and as research investigation [15, 16, 20, 21, 25] level. However, there are aspects that these tools cannot evaluate automatically. Indeed, while they are very efficient in identifying some types of problems, such as the absence of alt-text for non-textual content or labels




*CHItaly 2025, Salerno, Italy*

ACM ISBN 979-8-4007-2102-1/2025/10
https://doi.org/10.1145/3750069.3750310

for form fields, other aspects, such as the logical flow of elements on a page, its semantic content, or keyboard access, necessarily require manual review or real user interaction for their accessibility validation. Human judgment, therefore, for a full and accurate accessibility assessment, cannot yet be completely replaced. For this reason, the combination of different methods and tools, both automatic and non-automatic, and the involvement of both experts in the field and users with disabilities is recommended. The evolution of Large Language Models (LLMs) [6], thanks to their advanced language processing capabilities, can help overcome some of these limitations and automate some of those checks that, until now, only a human could perform, and this is the aspect that this study aims to investigate. For this purpose, we have identified a set of accessibility techniques that current tools usually cannot address and have investigated the possibility of addressing them through one of the widely used current LLMs (GPT 4o).

In the paper, we first review recent work on using LLMs for addressing accessibility validation, next we describe how we have designed prompts for GPT 4o for addressing the selected W3C techniques, we report on their applications on Web pages of widely used websites and the validation of such results. Lastly, we draw some conclusions and indications for future work.

## 2 RELATED WORK

Recently, some research studies have started to investigate the main advantages and limitations of the use of LLMs in the automatic validation of web accessibility. A study [11] explored the GPT-3.5 model's ability to find accessibility issues, analysing only HTML elements such as tables, forms and images, both accessible and non-accessible. The model evaluated their accessibility and suggested corrections; its responses were largely correct, albeit with some inaccuracies (particularly in assessing tables). However, since only textual inputs were provided to the model, it could not assess aspects such as the appropriateness of images' alt-text. Holmund's work [12] explored ChatGPT's ability to improve web accessibility for blind users. A website with intentional accessibility problems was developed, and prompts were designed to simulate real interactions and address common issues faced by visually impaired users, such as missing or inadequate alt-text, low colour contrast, or unintuitive navigation order. The model was 64.42% effective in summarising and explaining web content with accessibility problems and showed some limitations, such as its tendency to hallucinate and generate false information. The López-Gil and Pereira study [18] explored using LLMs to automate accessibility testing for three WCAG success criteria that require manual evaluation with current tools. To validate their approach, the authors selected the corresponding test cases from WCAG-ACT Rules to test these criteria and modified them by deliberately introducing errors. The authors developed scripts that use LLMs to assess these criteria automatically, tested them on the test cases, and compared the results with those obtained with some existing automatic evaluation tools. The LLM-based scripts were able to successfully identify the issues, with an overall detection rate of 87.18%, suggesting that LLMs can be integrated with existing automated accessibility tests, improving the ability to detect issues that automatic tools currently miss.

While the studies discussed in the previous section focus on LLMs' ability to detect accessibility violations, others have explored their potential to also correct these errors. A study [22] explored GPT's ability to fix accessibility problems of two websites: Qatar Airways and British Airways. Accessibility errors were identified using the WAVE tool and manual inspection, revealing significant violations. The websites' code with the detected errors was fed to GPT, which generated both suggestions to fix the errors and corrected HTML, and was able to fix a significant number of errors automatically but there were also several cases in which the results were not relevant. The authors note that its effectiveness depends on many variables: the training quality, the accuracy with which the WCAG are provided, and the website complexity. Another study [13] tested GPT 3.5 for automatically correcting WCAG 2.1 violations using prompt engineering techniques and direct DOM modification. A dataset of 25 websites was created and tested using the Playwright tool. The prompts included: the incorrect HTML code, error types, error descriptions, suggested changes, and desired answer format. GPT then generated corrections, and the corrected DOM was added to the dataset. The corrected DOMs were tested again with Playwright, showing a 51% error reduction. The results showed that, in this study, the LLM was more effective in finding text-based accessibility errors than ones related to the layout or visual aspects. A similar approach was developed in another study [10]: a web application for the automatic real-time correction of web accessibility problems. Four frequent web accessibility problems were selected: lack of ARIA landmarks to identify page regions, missing or incorrectly inserted titles, text with poor contrast with the background, and incorrect or absent labelling of form elements. The developed tool renders the page's DOM, with a "Page Scanner" collecting DOM data and screenshots before sending them to the server via an API. Then, an AI-based component, based on GPT-3.5 and trained with JSON-formatted accessibility data, analyses and corrects the errors. To validate the solution, some example web pages with accessibility errors were developed and tested with the system; then, the original page version and the modified one were manually compared, yielding positive results with a reduction of 51% of errors.

As shown by the analysis of previous studies, different approaches have been proposed to explore LLMs' capabilities in web accessibility assessment. They vary not only in the LLMs' role in the accessibility assessment but also in scope, input type for the LLMs and testing method used. Some studies have used LLMs to fix violations, by modifying the DOM or generating correct code. This approach reduces manual effort for developers but heavily depends on the accuracy of the model's analysis, risking introducing new errors. Therefore, adequate fine-tuning of the models is required, as well as a complex technical implementation. Other studies have instead focused on having models find and explain violations, a more flexible approach that allows developers to choose the best way to address issues contextually. In general, such first explorations had a limited scope, addressing rather limited issues. For example, the study carried out by Delnevo et al. [11] analysed only HTML forms and tables; López-Gil and Pereira [18] referred to three WCAG criteria, while Dash [10] examined four common violations. The inputs for the LLMs validation also varied across studies. Some have analysed the full web pages' DOM [10, 13, 22], providing a

comprehensive view of accessibility but requiring significant computational resources and effort for the analysis. Others used only code snippets [11] as input and, in one case, screenshots of the page elements [12]; these approaches facilitate the identification of individual problems and are easier to implement but provide more limited results.

## 3 The Proposed Approach

This work aims to investigate an aspect neglected by previous literature, namely the use of LLMs to automate the evaluation of web accessibility aspects that currently require manual intervention, as they involve a semantic interpretation of the content or context of the page. The proposed solution integrates GPT-4o functionalities to automate the checks required by a subset of WCAG techniques currently requiring manual verification. The only study that has used LLMs for a similar purpose [18] developed scripts to automate the checks of some ACT rules for three WCAG success criteria, accessing the LLM functionalities. Thus, their validation was related only to the simple cases that are considered by those ACT rules, while we aim to address more complex cases involving semantic information concerning the Web sites validated. For this purpose, we focus on specific WCAG techniques, which provide concrete and detailed guidance for meeting their validation. The proposed system addresses assessment for broader aspects not considered in the previous studies. In addition to assessing the descriptiveness of alternative texts for images and links and the correct identification of languages, semantic checks have been added, focusing on the descriptiveness of the page title, clarity of labels for interactive elements, correctness of markup usage for headings, and appropriate use of ARIA landmarks. We have also integrated the validation scripts into a web application that features an intuitive user interface and is usable for real website validation.

The W3C WCAG are based on success criteria that describe the requirements web content must meet to ensure accessibility. The validation of the success criteria depends on the results of the validation of the corresponding techniques. These techniques offer detailed guidance on addressing specific aspects of success criteria, which often deal with broader aspects. This study focuses on a subset of such techniques, which were selected among those not fully supported by the existing validation tool MAUVE++ [5, 12], the tool used by the national accessibility agency (Agid) for monitoring the level of accessibility of all Italian public organisations' Web sites (and in general they are not supported also by the other accessibility validation tools). The selected techniques are the following:

- Technique G94: Providing short text alternatives for non-text content that serves the same purpose and presents the same information as the non-text content;
- Technique G91: Providing link text that describes the purpose of a link;
- Technique G88: Providing descriptive titles for Web pages;
- Technique H58: Using language attributes to identify changes in the human language;
- Technique G131: Providing descriptive labels;
- Technique H42: Using h1-h6 to identify headings;
- Technique ARIA11: Using ARIA landmarks to identify regions of a page.

The proposed "TeVal" validation tool was implemented to support user validation of such techniques. The front-end was built with HTML5, CSS3 and JavaScript, utilising the Bootstrap 5 framework for a responsive user interface, and consists of three main parts. There is an input section, which allows the user to enter the URL of the page to be validated and specify how they prefer to receive the results: in real time or by email. This page also includes a list of supported techniques, each associated with a link to the official documentation of the technique. If the user has selected the option to receive the results by email, they will see a feedback page displaying a confirmation message that informs them that they will receive the results by email. This choice was motivated by the need to improve the user experience, given that the processing of the results may take some minutes depending on the complexity of the considered web page, resulting in an excessive wait for the user. There is also a page displaying the validation results, presented in a tabular format. The results include the date and time of the validation, the validated URL, and the total number of successes, failures and warnings found. The user can expand the results for each technique, containing the model responses for each element analysed, identified by its "xpath" (an example in Figure 1). The user can also select to display the results of all previous validations saved in the database, by clicking on the "See all validations" button. The backend part of the application is composed of a database, for storing validation results, a server, implemented using the Node.js Express framework, and a validation script, which is composed of two modules: a main validation function, which takes the URL to validate as input, executes the code related to the techniques and returns the results; some Javascript functions associated with the techniques to validate, which perform the accessibility checks on the page at the specified URL, through the integration with the GPT-4o model.

The Chat Completion API service offered by Microsoft Azure was used to integrate the GPT model features. Socket.IO, a Javascript library that enables real-time bidirectional communication between the server and the client and based on Web-Sockets, was used to send real-time messages about the validation status, which are displayed in a static modal window if the user chooses real-time processing. The Selenium library, specifically the Selenium WebDriver tool, was used for creating an instance of a headless browser to analyse the actual DOM after the page is loaded in such browser.

The validation of each technique is performed with the support of the GPT-4o model. Each function for the implemented techniques includes a specifically formulated prompt, with specific instructions for the model's evaluation. This prompt is sent as a message with the role "system", while the elements to be evaluated, extracted from the page, are sent as "user" messages.

In designing the prompts, a consistent structure was maintained for all techniques. The seven prompts are multifaceted, ranging from approximately 700 to 1100 tokens; this complexity is motivated by the need to provide all the necessary information to the model and guide it in generating the desired answers. In their formulation, OpenAI prompting guidelines[1] were considered as well. For example, to make the overall task more manageable, it was

[1]https://help.openai.com/en/articles/6654000-best-practices-for-prompt-engineering-with-the-openai-api

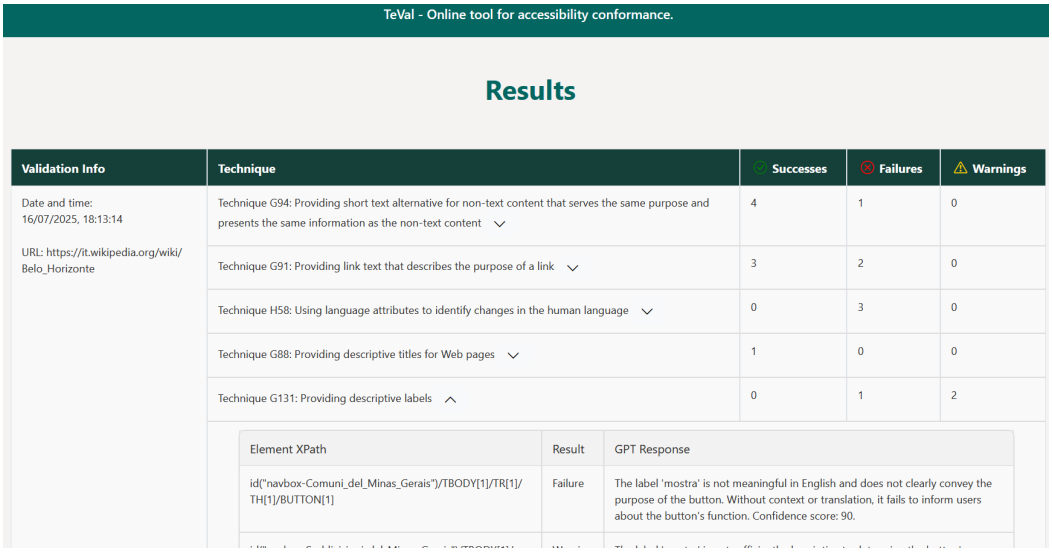



**Figure 1: An Example of Validation Results**

broken down into simpler steps, using a numbered list. All prompts contain the following elements, which apply several prompting strategies.

*Role of the model.* Each prompt begins with the sentence "You are a web accessibility evaluation tool": this is the application of the "Role Prompting" or "Persona Prompting" technique. It consists of asking the model to impersonate a specific role and adopting its skills and tone, in this case, a web accessibility evaluation tool [27].

*An indication of the task it is supposed to perform.* This is provided clearly, avoiding vague or generic wording. For example, the following are the instructions for the G94 technique: "Your task is to evaluate whether alternative texts for images on webpages are appropriate according to WCAG guidelines. The alt-text should serve the same purpose and present the same information as the image, and should be able to substitute for the non-text content. The page would still provide the same function and information if the non-text content were removed from the page and substituted with text. The text alternative would be brief but as informative as possible. When non-text content contains words that are important to understand the content, the alt text should include those words."

*An indication of the content from the web page under assessment that will be provided to perform the task.* For example, for the G94 technique: "You will be provided as input with the following: The image found on the webpage. The associated alternative text. When the alt-text is empty or absent, you will be explicitly informed. The surrounding context of the image. The page title, headings and the content of the "keywords" and "description" <meta> tag, if found."

*An explanation from the W3C documentation.* The purpose is to train the model by providing relevant context information, in this case, an explanation technique quoted directly from the official documentation. For example, this is the explanation for the G88 technique: "2. Use the following explanation to guide your evaluation: The intent is to help users find content and orient themselves within it by ensuring each Web page has a descriptive title. Titles identify the current location without requiring users to read or interpret page content. When titles appear in site maps or lists of search results, users can more quickly identify the content they need. User agents make the page title easily available to the user for identifying the page. For instance, a user agent may display the page title in the window title bar or as the tab's name containing the page. In cases where the page is a document or a web application, the name of the document or web application would be sufficient to describe the page's purpose. Note that it is not required to use the document's name or web application; other things may also describe the purpose or topic of the page. In cases such as Single Page Applications (SPAs), where several distinct pages/views are all nominally served from the same URI and the content of the page is changed dynamically, the title of the page should also be changed dynamically to reflect the content or topic of the current view."

*Some examples.* This is an application of the "few-shot prompting" technique [27], which consists of providing demonstrations of the execution of the task or the resolution of the problem; in this case, the examples are of correct applications of the techniques and are taken from the W3C documentation. This is useful to train the model to correctly classify the elements. The following excerpt reports the examples used in the prompt for the G88 technique, extracted from the text of the technique itself:

"Examples

Example 1: A title that lists the most important identifying information first

A Web page is published by a group within a larger organisation. The title of the Web page first identifies the topic of the page, then shows the group name followed by the name of the parent organisation.

<title>Working with us: The Small Group: The Big Organization</title>

Example 2: A synchronised media presentation with a descriptive title A synchro-nised media presentation about the 2004 South Asian tsunami is titled "The Tsunami of 2004."

Example 3: A Web page with a descriptive title in three parts

A Web page provides guidelines and suggestions for creating closed captions. The Web page is part of a "sub-site" within a larger site. The title is separated into three parts by dashes. The first part of the title identifies the organisation. The second part identifies the sub-site to which the Web page belongs. The third part identifies the Web page itself. (For a working example, see WGBH – Media Access Group – Captioning FAQ.)

Example 4: A newspaper Web page

A Web site that only permits viewing of the current edition titles its Web page "National News, Front Page".

A Web site that permits editions from different dates to be viewed titles its Web page, "National News, Front Page, Oct 17, 2005""

*Request to provide a "confidence score"*, i.e. a value between 0 and 100 that expresses the level of confidence in the evaluation. This technique is called "verbalised score" [27] and is useful for understanding how confident the model is in its response, even if its effectiveness is still debatable.

*Response length.* Given the tendency of the model to return excessively long responses, it was necessary to request to keep the length of the response within 100 words, with the instruction "Keep your response within 100 words.", to limit the total number of tokens in the conversations.

*Response format.* To extract the model's responses for each technique and obtain the responses in a consistent and homogeneous format, the last instruction of the prompt specifies the JSON format that the results must have, which is subsequently parsed; "assessment" will contain one of the three possible outcomes of the evaluation ("success", "failure" or "warning"); while "reasoning" will contain the reasoning followed in the evaluation:

"Here is the JSON format the response must have:

{ "Assessment" : "*your assessment*", "Reasoning" : "*your reasoning*" }"

## 4 Validation Test

This section describes the method used to validate the results of the proposed solution based on the use of GPT4o, including the criteria for forming the dataset of pages to be tested, an overview of the obtained results, followed by an analysis of the model's performance for the main aspects considered.

***Pages considered and results***. To test the application, 10 pages belonging to the most visited sites on the web (CNN, Microsoft, eBay, Wikipedia, Weather, Samsung, Instagram, Google, Twitch, Fandom) were selected in January 2025. This choice aimed to ensure that the solution is tested on a sample representative of the average user's online experience, as these sites receive millions of visits daily. The selected pages also span different sectors, creating a varied dataset that covers various cases and functionalities. After conducting the validation on each page, the results were manually verified by an accessibility expert (a paper author), referencing the official documentation of the techniques and considering the page context of the evaluated elements. During the test, 251 responses were obtained, which were then manually verified, specifically: 45 for images, 50 for links, 10 for titles, 28 for linguistic changes on the page, 48 for interactive elements, 45 for headings and 25 for landmark elements. Overall, the model identified a total of 185 successes (73.7%), 54 failures (21.5%), and 12 warnings (4.7%). The correct answers were 232, showing an effectiveness rate of 92.4%. These results demonstrate the model's positive performance in detecting the correct application or violations of the techniques. However, some "warnings" and incorrect responses also emerged, highlighting some limitations of the adopted approach.

***Meaningfulness of image alternative texts.*** The G94 technique concerns the association of a short text alternative for non-textual content that performs the same function and conveys the same information. This is important so that assistive technologies can render non-textual content for users who cannot perceive it. The G94 validation function identified 15 successes and 29 failures among the 45 images analysed, achieving correct answers in 11 of 15 successful cases and 28 of 29 failures, and one warning. The model effectively evaluated images by considering their type and the contextual information provided, namely: the page title, the content of the "description" and "key-words" attributes of the <meta> tag, and the page headers, which were useful for the model to identify the function of the page, the main topics, and the possible function of the image in the context. Analysing alt-text requires distinguishing image types, as different aspects must be evaluated. Many of the evaluated images were informational, conveying ideas that can be expressed in a short sentence and typically requiring a description of their essential visual elements. The evaluation of the model was mostly accurate. For instance, for the CNN article page about the German economy, it correctly assessed the main image, depicting Alexanderplatz in Berlin, as informative and relevant to the article's socioeconomic context, with the alt-text value "Shoppers in central Berlin in January 2025" essentially describing the content and conveying the same information and functionality. Similarly, for the Google homepage, the model accurately identified the main logo as a text image, noting that the alt text "Google" reflects the text contained within. This image serves an informative role by identifying the company, and the use of alt-text is appropriate due to the logo's prominence, but since it doesn't function as a link, the simple name without additional descriptions suffices for accessibility. The model also recognised several cases of violations. For instance, on the eBay product page, it correctly flagged the alternative text "Picture 1 of 3" as not descriptive. Similarly, on the Microsoft Create page, it identified "Alt text goes here" as a placeholder that was not replaced with a real alt-text. However, there were incorrect assessments as well. For the Samsung website page, it rated the alt text "Galaxy 24+ 512GB yours for € 1,049" as successful because it provides essential information about the product shown. However, the rating is incorrect; not only because it could better describe the product's visual aspects, but mostly because the alt-text does not describe the model shown (a tablet) but refers to another product (a smartphone). On the weather.com page, it considered two alt-texts "Window showing outdoor greenery" or "A basket with towel" for two images associated with tips about reducing pollen exposure, as successes, considering them informative; but the images play more of a decorative role, and do not add relevant information to the content.

***Meaningfulness of link texts.*** Regarding the G91 technique, which involves the use of descriptive texts for links that clarify their purpose. The model performed well, achieving 42 successes out of 50 (84%), with 7 failures (14%) and one case resulting in a "warning." Overall, it provided 45 correct answers out of 50 (90%). In successful cases, the model accurately assessed the link text's alignment with the main topics or functions of the landing page. For example, on the Google homepage, the "About us" link leading to "Google About" was deemed descriptive, as was "Forgot your password?" on the Instagram log-in page, which directs users to the "Reset password" page. A single warning arose from a link on the Microsoft page, with the text "Learn and expand". The model noted that while the page's focus on acquiring new skills is relevant, the link text itself does not reflect these topics.

***Meaningfulness of titles***. Moving on to the G88 technique, which focuses on using descriptive titles for web pages to identify and summarise their content clearly, the model effectively evaluated titles based on the provided elements (page headers and some <meta> tag attributes). This approach avoided overwhelming the model with excessive information. The results showed 6 successes out of 10, with 4 failures, but all responses were correct and justified, resulting in 100% effectiveness. The model was able to recognise highly descriptive titles, such as the CNN page title "German economy shrinks for second straight year | CNN Business", clearly identifying the main topic of the contained article. For the eBay page, the title "Xbox Series S - Certified Refurbished 889842651546 | eBay" was deemed successful since it reflects the content related to the sale of a specific product, including its name, condition, unique identifier, and the marketplace name. There were several cases in which the model deemed the titles not sufficiently descriptive. For example, on the Instagram page, which is a login page, but this is not indicated by the simple title "Instagram"; or in the case of "Google", for which the model suggested more complete titles such as "Google Services Overview" or "Explore Google Products and Solutions"; or even the cases of "Twitch" and "Fandom", which do not indicate the type of content that the user can expect to find. Despite the worldwide popularity of these websites, to ensure full WCAG compliance, it is not enough that the page title identifies the company or organisation that owns it, but it must also be descriptive of the content present within it or the offered services.

***Correct reporting of language changes***. The H58 technique focuses on correctly reporting linguistic changes on the page using the "lang" attribute and standards-compliant linguistic codes. The results were mostly positive: out of a total of 28 responses, 19 were considered successes (67%) and 9 were considered failures (32.1%). The correct responses were 26 out of 28 (92.8%).

***Labels meaningfulness***. Regarding the G131 technique, which focuses on using descriptive labels for interactive components that indicate their functionality or type of user input. The model correctly evaluated these elements, considering the function they had in the webpages. In total, it considered 43 elements as successes (89.5%), 3 as failures, and 2 as warnings, with a high accuracy rate of 97.9%. Different types of interactive components were analysed. Some were "submit" type buttons, used to submit form data. Those examined were mostly associated with search input fields; the model gave correct evaluations for most cases. For example, two buttons labelled "Search" on the Microsoft and Samsung pages were considered successes. Other button elements had the function of toggling the visibility of navigation menus. In the case of the eBay page, for example, the model correctly identified and evaluated the button labelled "Expand notifications" positively. Other evaluated buttons were used to perform actions and were all considered successes: for example, the labels "Share on Facebook" or "Share on x" associated with the icons of these social networks on the CNN article page. Only on the eBay page, elements of the <select> and <option> types were present. The respective labels "Select a category for search" and "All categories" were correctly judged as successes. Another common element was the text input fields, often used for search functionality. For example, the eBay page featured an input field with the descriptive label "Search for anything," which the model evaluated correctly. Similarly, the Google homepage's <textarea> labelled "Search," and the Weather page's input field labelled "Search city or postal code," were both evaluated correctly. However, the Twitch page raised some concerns; the model deemed the label "search input" sufficient, though it could be improved. The only cases of violations of the technique, correctly identified, were found on the Microsoft page, where the input field for the search has only a "Search models" placeholder but lacks an associated label; and the other one on the Weather page, where the label "[object Object]" for a button was considered correctly as inadequate.

***Semantic correctness in the use of headings***. The H42 technique involves the correct use of HTML markup for the page headers (from <h1> to <h6>). This includes ensuring not only proper indentation according to the hierarchical structure but also using the appropriate heading levels for each section. The approach chosen for this technique showed good results, but with some limitations. Out of a total of 45 responses, 36 were successes (80%), 2 failures (4.4%) and 7 "warnings" (15.5%). The correct responses represent 82.2% of the total (37 out of 45). For example, for the CNN article page, the model correctly evaluated the hierarchy of the headings, identifying as successes the use of <h1> for the title of the article, and <h3> for the titles of the sections relating to complementary content ("More from CNN"). For the Fandom page, it judged as correct the use of <h2> for main sections such as "Movies" and "Games", and <h3> for the subsections. On the eBay page, two violations were also found: a <h2> with "breadcrumb" text, semantically incorrect, and a <h2> element containing the text "Picture 1 of 3" (used to describe an image of a carousel). These cases not only violate the logical structure of the headings but are also used in a semantically incorrect way. Some evaluations were more uncertain. For instance, on the Instagram page, five <h5> headings received a "warning" because there were no higher-level headings they could logically belong to. However, these were the only headings present on the page. A more complete page structure might have led the model to be more accurate, highlighting the need for a broader contextual framework in evaluations.

***Semantic correctness of the use of ARIA landmarks***. The ARIA11 technique refers to the use of the "role" attribute to identify landmarks within the page, i.e. the main regions, facilitating navigation for users who use assistive technologies by allowing them to "jump" between sections without having to read all the content. Across the ten tested pages, elements for nearly all roles

(except for "form") were found, with the model performing well in evaluating their use. Of 25 responses, 24 were successful, and only one was a warning. Regarding the "main" role, which identifies the primary content of the page, successes were observed on the CNN article page (an <article> element) and the Instagram and Samsung pages (a  element). The use of these elements was correct as they contained the main content. The "region" role was successfully identified in several instances, including a  on the CNN page for the Cookie banner and a <section> element on the eBay page for an error message. However, warnings were issued for the Weather and CNN pages due to missing "aria-labels". The "navigation" role was used correctly on the eBay and Samsung pages within <nav> elements, and the Google page within a , all containing navigation links. The "search" role was appropriately identified in the CNN, Google, Microsoft, Samsung, and Weather pages for <form> elements with search functionalities, while in the Wikipedia page, these were within a . The "contentinfo" role, for sections containing information about the document (like privacy policies), was correctly identified in a  on the Google homepage and in a <footer> on the Instagram login page. Only the Weather page had a "banner" role, used correctly to identify the main title in a <header>. The "complementary" role was used on the Twitch page to indicate a sidebar with links to live streams, which was deemed appropriate since it supports the main content without being the primary focus. The only incorrect evaluations were on the CNN article page, regarding the "search" role in two <form> elements. Although the model recognised the roles as correct, it failed to note the absence of "aria-label" or "aria-labelledby" attributes to distinguish them, which is important for accessibility, indicating a need for clearer instructions for the model.

## 5 Conclusions and Future Work

This study aimed to investigate the effectiveness of using the LLM GPT-4o for automatically analysing semantic and contextual aspects of web accessibility, which are problematic for current automatic validators to analyse. The results obtained, with a 92.4% effectiveness of the model, are promising and demonstrate how an LLM-based solution can provide useful support for validation. The prompts developed are long and cannot fit in the paper but we can provide them to interested researchers.

One key advantage is the ability to analyse many semantic and contextual aspects of web pages in much shorter times than manual analysis, thus reducing the costs in terms of time and human effort required. The model identifies successes or failures and provides detailed explanations based on the specific context of each element. Another important advantage of this approach is its potential extendibility to other accessibility aspects beyond those considered in this study. This extendibility is enhanced by the model's capability to understand textual input and various media types, such as images, audio, and video, in multimodal contexts. One further positive point of the proposed solution concerns integrating the model's features in a simple-to-use web application, usable even by users who are not accessibility experts, thanks to the return of easy-to-understand answers focused on individual elements. However, some considerations are necessary regarding the limitations of the study and possible areas for future improvement. While the results are promising, the restricted set of web accessibility aspects analysed limits the generalizability of the findings. This targeted selection focused on specific semantic and contextual issues that are problematic for current tools, inevitably excluding other relevant areas of the WCAG guidelines. The dataset used for testing was limited; thus, further tests with larger datasets are necessary to fully validate the approach's effectiveness.

As for future developments, the system's functionality could be expanded. The system could also be integrated with current validation tools, combining the semantic capabilities of an LLM with the features already implemented, thus providing a more complete analysis. An important aspect regards the model's accuracy, which strictly depends on the prompt quality and the completeness of contextual information provided. From test results, it emerged that uncertain or incorrect responses are mainly due to the partiality of the context or to the interpretation of particular cases not covered by the examples provided in the prompt. Therefore, providing the model with a more complete context and a greater number and variety of examples could lead to more precise evaluations. However, it is also important to note that, despite the high effectiveness and continuous improvement of the algorithms underlying the models, it is not possible to rely completely on them; a manual check remains necessary to verify the correctness of the results. In fact, the models are not always consistent in their evaluations, and a different response can be obtained for similar elements or conditions. Their integration should therefore be understood as support, not a total replacement for human intervention.

Another important development concerns cost optimisation. Integrating LLMs, such as OpenAI models, often requires a subscription, leading to significant costs that can increase exponentially based on page length. In this study, analysing a single element required between 800 and 2,500 tokens (those for the prompt and in addition those for the additional web page content necessary to perform the evaluation), and with web pages containing numerous elements, costs can escalate. As the number of requests increases, response times also lengthen, affecting user experience. Possible solutions include exploring open-source models or lighter versions for simpler tasks and reserving advanced models for complex content analysis. Furthermore, implementing a feedback collection system from end users would be useful. They could express the clarity and usefulness of the answers using a numerical or qualitative scale. This could be used to improve the model's performance, making answers more accurate and aligned with user needs. In conclusion, this work demonstrates that integrating advanced linguistic models can represent a significant breakthrough in improving web accessibility, although further developments are needed to overcome the current limitations.

## References

[1] Julio Abascal, Myriam Arrue, and Xabier Valencia. 2019. Tools for Web Accessibility Evaluation. In Web Accessibility, Yeliz Yesilada and Simon Harper (eds.). Springer London, London, 479-503. https://doi.org/10.1007/978-1-4471-7440-0_26

[2] Siddikjon Gaibullojonovich Abduganiev. 2017. Towards Automated Web Accessibility Evaluation: A Comparative Study. IJITCS 9, 9 (September 2017), 18-44. https://doi.org/10.5815/ijitcs.2017.09.03

[3] Barbara Rita Barricelli, Pierlauro Sciarelli, Stefano Valtolina, and Alessandro Rizzi. 2018. Web accessibility legislation in Italy: a survey 10 years after the Stanca Act. Univ Access Inf Soc 17, 1 (March 2018), 211-222. https://doi.org/10.1007/s10209-017-0526-z

[4] Abdo Beirekdar, Marc Keita, Monique Noirhomme, Frédéric Randolet, Jean Vanderdonckt, and Céline Mariage. 2005. Flexible Reporting for Automated Usability and Accessibility Evaluation of Web Sites. In Human-Computer Interaction - INTERACT 2005, Maria Francesca Costabile and Fabio Paternò (eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 281-294. https://doi.org/10.1007/11555261_25

[5] Giovanna Broccia, Marco Manca, Fabio Paternò, and Francesca Pulina. 2020. Flexible Automatic Support for Web Accessibility Validation. Proc. ACM Hum.-Comput. Interact. 4, EICS (June 2020), 1-24. https://doi.org/10.1145/3397871

[6] Eric Brynjolfsson, D. Li e L. R. Raymond, Generative AI at Work (Working Paper No. 31161), National Bureau of Economic Research, pag. 5, 2023. https://doi.org/10.3386/w31161

[7] Andreas Burkard, Gottfried Zimmermann, and Bettina Schwarzer. 2021. Monitoring Systems for Checking Websites on Accessibility. Front. Comput. Sci. 3, (February 2021), 628770. https://doi.org/10.3389/fcomp.2021.628770

[8] Cazenave, E. and A. Bellantoni (2022), Accessible and inclusive public communication: Panorama of practices from OECD countries, OECD Working Papers on Public Governance, ?54, OECD Publishing, Paris, https://doi.org/10.1787/222b62d9-en

[9] EU Commission. 2016. Directive (EU) 2016/2102 of the European Parliament and of the Council. Retrieved from https://eur-lex.europa.eu/eli/dir/2016/2102/oj

[10] S. K. Dash, AI-powered real-time accessibility enhancement: A solution for web content accessibility issues, in: JOIN (Journal Online Informatika), 9(1), pp. 70-79, 2024. https://doi.org/10.15575/join.v9i1.1310

[11] Giovanni Delnevo; Manuel Andruccioli; Silvia Mirri, On the Interaction with Large Language Models for Web Accessibility: Implications and Challenges, 2024 IEEE 21st Consumer Communications & Networking Conference (CCNC), Las Vegas, pp. 1-6, 2024. https://doi.org/10.1109/CCNC51664.2024.10454680

[12] M. Holmlund, Evaluating ChatGPT's Effectiveness in Web Accessibility for the Visually Impaired, Università di Linnaeus, 2024. https://urn.kb.se/resolve?urn$=$urn:nbn:se:lnu:diva-130717

[13] Calista Huang, Alyssa Ma, Suchir Vyasamudri, Eugenie Puype, Sayem Kamal, Juan Belza Garcia, Salar Cheema, Michael Lutz, ACCESS: Prompt Engineering for Automated Web Accessibility Violation Corrections, 2024. https://doi.org/10.48550/arXiv.2401.16450

[14] Nicola Iannuzzi, Marco Manca, Fabio Paternò, and Carmen Santoro. 2022. Usability and transparency in the design of a tool for automatic support for web accessibility validation. Univ Access Inf Soc (November 2022). https://doi.org/10.1007/s10209-022-00948-x

[15] Nicola Iannuzzi, Marco Manca, Fabio Paternò, and Carmen Santoro. 2023. Large Scale Automatic Web Accessibility Validation. In ACM International. Conference on Information Technology for Social Good (GoodIT '23), September, 06-08, 2023, Lisbon, Portugal. ACM

[16] Nicola Iannuzzi, Marco Manca, Fabio Paternò, and Carmen Santoro. Combined accessibility validation and monitoring of web sites and PDF documents. Univ Access Inf Soc (2025).

[17] Jonathan Lazar, Julio Abascal, Simone Barbosa *et al.* , Human-computer interaction and international public policymaking: A framework for understanding and taking future actions, Human-computer interaction and international public policymaking: A framework for understanding and taking future actions, Foundations and Trends in Human-Computer Interaction, 9(2), pp. 69-149, 2015

[18] JM. López-Gil e J. Pereira, Turning manual web accessibility success criteria into automatic: an LLM-based approach, in: Universal Access in the Information Society 14, 2024. <https://doi.org/10.1007/s10209-024-01108-z>

[19] Marco Manca, Vanessa Palumbo, Fabio Paternò, and Carmen Santoro. 2023. The Transparency of Automatic Web Accessibility Evaluation Tools: Design Criteria, State of the Art, and User Perception. ACM Trans. Access. Comput. 16, 1 (March 2023), 1-36. https://doi.org/10.1145/3556979

[20] Beatriz Martins and Carlos Duarte. 2022. Large-scale study of web accessibility metrics. Univ Access Inf Soc (December 2022). https://doi.org/10.1007/s10209-022-00956-x

[21] Silvia Mirri, Ludovico Antonio Muratori, and Paola Salomoni. 2011. Monitoring accessibility: large scale evaluations at a Geo political level. In The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility, October 24, 2011. ACM, Dundee Scotland, UK, 163-170. https://doi.org/10.1145/2049536.2049566

[22] A. Othman *et al.*, Fostering websites accessibility: A case study on the use of the Large Language Models ChatGPT for automatic remediation, in: Proceedings of the 16th International Conference on Pervasive Technologies Related to Assistive Environments, (PETRA '23). Association for Computing Machinery, New York, pp. 707-713, 2023. <https://doi.org/10.1145/3594806.3596542>

[23] Marian Padure and Costin Pribeanu. 2019. Exploring the differences between five accessibility evaluation tools. 2019. .

[24] Christopher Power, André Freire, Helen Petrie, and David Swallow. 2012. Guidelines are only half of the story: accessibility problems encountered by blind users on the web. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, May 05, 2012. ACM, Austin Texas USA, 433-442. https://doi.org/10.1145/2207676.2207736

[25] Costin Pribeanu. 2019. Large-scale accessibility evaluation of Romanian municipal websites. International Journal of User-System Interaction 12(2) 2019, pp. 83-98.

[26] Antonio Giovanni Schiavone and Fabio Paternò. 2015. An extensible environment for guideline-based accessibility evaluation of dynamic Web applications. Univ Access Inf Soc 14, 1 (March 2015), 111-132. https://doi.org/10.1007/s10209-014-0399-3

[27] Sander Schulhoff *et al.*, The Prompt Report: A Systematic Survey of Prompting Techniques, arXiv, novembre 2023 [Online]. https://arxiv.org/html/2406.06608v1#Ch2.S2

[28] Markel Vigo, Justin Brown, and Vivienne Conway. 2013. Benchmarking web accessibility evaluation tools: measuring the harm of sole reliance on automated tests. In Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility, May 13, 2013. ACM, Rio de Janeiro Brazil, 1-10. https://doi.org/10.1145/2461121.2461124