Spring 2024

Evaluating Automated Accessibility Checker Tools

Thai Nguyen

Evaluating Automated Accessibility Checker Tools

Thai Nguyen

Abstract

The internet has grown into a tool that is essential for communication, education, and commerce. However, for those with disabilities, web content can often have significant barriers if they are not designed with accessibility in mind. Digital accessibility ensures that websites and applications are usable by everyone, regardless of ability. In order to make sure web content complies with accessibility standards, automated accessibility evaluation tools have grown in significance. This paper investigates the effectiveness of automated accessibility evaluation tools (AAETs) for identifying web accessibility issues. The research analyzed five prominent AAETs: Accessibility Insights, aXe DevTools, Lighthouse, SiteImprove, and WAVE and tested them against four popular news websites and one local news site. The paper introduced a weighted coverage error ratio (CER) metric which benefits from all issues found regardless of accuracy. After analyzing the results, WAVE was the most effective tool in the scope of this paper if only by a small margin. Ultimately, the best way to evaluate accessibility is to use a combination of the tools and incorporate manual testing.

Introduction

What is Web Accessibility

According to the World Wide Web Consortium (W3C), an international organization that works to develop Web standards, “web accessibility means that websites, tools, and technologies are designed and developed so that people with disabilities can use them” (Henry 2024) [1] . More specifically, individuals can: view, comprehend, navigate, and engage and contribute to the Web. Web accessibility considers a broad spectrum of disabilities, such as impairments in speech, cognition, motor function, hearing, and vision. With the need for a standard set of guidelines that people should follow, came the Web Content Accessibility Guidelines (WCAG). These guidelines are a byproduct of the broader disability rights movement and the legal compliance of the Americans with Disabilities Act. In 2013, the Department of Transportation (DoT) amended their rules to require that U.S. airlines made their websites accessible to individuals with disabilities, requiring that “Web sites must conform with the standard for accessibility contained in the widely accepted Web Content Accessibility Guidelines 2.0 and meet the Level AA Success criteria” (Dept of Transportation 2013) [4]. According to the DoT, the purpose of this amendment was to ensure that passengers with disabilities can access the same services and information about air travel via the airline websites and airport kiosks as able-bodied passengers. To the DoT, equal access means “passengers with disabilities can obtain the same information and services… as conveniently and independently as passengers without disabilities” (Dept of Transportation 2013) [4]. When web content is accessible, we ensure that everyone has equal access to information and functionality on the web. However, due to the constant evolution of technologies, accessibility is a moving target. It is also crucial to continuously assess and improve accessibility to ensure that we meet the evolving needs of web accessibility and provide equitable access for people with disabilities to the Web.

The Web Content Accessibility Guidelines was developed by W3C to provide a global standard for web content accessibility that “meets the needs of individuals, organizations, and

governments internationally" (Henry 2024) [1]. Web content refers to the information included in a web page or application, this can be natural information like text, images, and sounds or code/markup that defines structure or presentation. The WCAG was created to provide a technical standard for web developers, content creators, accessibility evaluation tool developers, and anyone who may want or need a standard for web accessibility. The WCAG standards are classified based on four main principles: perceivable, operable, understandable, and robust. These principles are often referred to as POUR. Table 1 summarizes these principles and gives some examples:

Table 1. Four Main Principles of WCAG

Principle	*Definition*	*Examples*
Perceivable	Information and user interface components must be presentable to users in ways they can perceive.	○ Captions for audio content ○ Description for video content ○ Color contrast ratio at least 4.5:1
Operable	User interface components and navigation must be operable.	○ Pages have proper titles and focus order ○ Content can be operated through a keyboard
Understandable	Information and the operation of the user interface must be understandable.	○ The language of the content is specified ○ Input fields have labels associated with them
Robust	Content must be robust enough that it can be interpreted reliably by a wide variety of user agents, including assistive technologies.	○ Elements have complete tags at the start and end and nested correctly

* The definitions and examples are from the Web Accessibility Initiative [19]

All web content must comply with these four principles otherwise, users with disabilities will not be able to access that content on the Web.

The WCAG does not outline specific actions that websites must take; however, it states what attributes accessible websites should do. The essence of the WCAG are testable requirements known as success criteria. In total, there are 78 criteria, and each is assigned to a level. The following are the levels of conformance:

- Level A (lowest): This is the bare minimum and typically considered unacceptable. There are 30 criteria, and these must be met in order to classify that website as accessible. Some requirements are that all non-text content like images and videos must have alternative text accompanying them, assistive technologies must be able to access content, and all content should be navigable with a keyboard.
- Level AA (mid-range): This level means that a website has met all the requirements of Level A and an additional 20 requirements. Most accessibility experts recommend this level of conformance. Examples of requirements include a minimum level of contrast of at least 4.5 to 1, content heading should follow a logical order, and navigation elements should be consistent across the website.
- Level AAA (highest): At this level, the site and all content must satisfy all Level A and Level AA, including an extra 28 requirements like the contrast ratio between background and text is at least 1 to 7 and any video or audio content must have sign language translation.

Related Works

There have been several studies that evaluate and assess the performance of these tools, employing diverse methodologies and metrics to gauge their effectiveness. Among these efforts, Alsaeedi (2020) undertook a comparative study of two prominent automated accessibility evaluation tools, SiteImprove and WAVE, focusing on their capacity to detect accessibility issues conforming to the WCAG 2.1. Their exploration introduced the concept of the coverage error ratio (CER) as a means to quantify tool performance. The ratio takes into account two variables, the number of errors detected by a given tool divided by the total number of errors

detected by all tools. By comparing the CER of each tool, the one with the highest CER can be considered the best performing tool. It is important to note that this metric "is focused on false negatives and benefits from all issues reported by the tools", meaning higher CERs are a result of the tools ability to detect a larger number of issues (Alsaeedi 2020) [2].

Similarly, Frazão and Duarte's (2020) [6] paper comparing accessibility evaluation plug-ins, advocating for a meticulous testing environment to mitigate external influences on tool performance. In their study, they had all the tools evaluate the same web page within a few seconds of each other to ensure that there were no differences between them. They also performed all the testing on the Chrome browser to avoid any extra errors being produced due to the browser. Their methodology highlights the importance of consistency and control in experimental setups, emphasizing the need for standardized testing protocols to ensure reliable results.

Despite the continuous evolution of these automated tools, a consensus emerges from the papers I reviewed. While these tools are valuable, they should not be the sole determinant of digital accessibility. Instead, a holistic approach that integrates multiple evaluation methods, including using a combination of automated tools or manual testing, is advocated to ensure the comprehensiveness of accessibility evaluations. In the next part, this paper will go over the methodology and criteria used to select the set of tools and websites. I will also explain the criteria I used to evaluate each tool and their usability.

Methodology

My paper aims to evaluate five automated accessibility evaluation tools (AAETs) for the Microsoft Edge browser: SiteImprove, Microsoft Accessibility Insights, Google Lighthouse, aXe DevTools, and WAVE browser extension. These were the top tools recommended by Harvard University for automated testing. Along with being free or open source, all these tools also have browser extension, which was an initial criteria for selection. I will be looking at many aspects of each tool like what success criteria they use, what kind of tests are offered, and the difference in the results of the reports generated. As a result, I will have assessed each tool and identified their strengths and limitations based on their results and also factor in their usability. With the data

collected, I hope to give potential enhancements to the automated accessibility evaluation process and contribute to bettering this type of tool.

Automated Tool Selection

The criteria I used to select these automated tools was that they needed to be free and available as an extension for Microsoft Edge. Due to this resulting in a very large list of tools, further criteria were that these tools needed to check for WCAG 2.1 conformance. In the end, I chose a set of five tools based on Harvard University's recommendation for automated testing tools and their popularity.

Google Lighthouse is an open source accessibility auditing tool built into the Chrome DevTools that helps identify accessibility issues on a webpage. Lighthouse has a feature that automates many of the WCAG 2.0 checks that are required by the Americans with Disabilities Act. It runs a comprehensive test that identifies warnings, errors, and gives potential fixes. Lighthouse also provides a score between 0-100 to represent that overall accessibility of the page. This is a useful benchmark, but it is still important to review each audit for specific issues.

WAVE, developed by WebAIM, is an online accessibility evaluation tool that can detect WCAG errors. The tool creates a report of its findings and categorizes the errors into six categories: alerts, errors, contrast errors, features, structural features, and ARIAs. WAVE offers many ways to check for accessibility like entering a web address, uploading a local file, or pasting HTML code. WAVE also has visual indicators that highlight accessibility issues directly on the webpage. Icons and overlays indicate problems, making them easier to identify.

The SiteImprove accessibility checker extension is a free extension available on most browsers. It checks any web pages for accessibility issues and provides an immediate overview of a website's accessibility along with explanations of how the issues may impact different users and recommendations on how to fix them, including links to relevant WCAG pages. SiteImprove also shows the exact location of where the errors occur on the page and the analysis is done within the browser, making it able to evaluate password-protected sites and dynamic content.

Accessibility Insights is another tool that helps developers find and fix accessibility issues in their websites and applications. It can be integrated with the development process

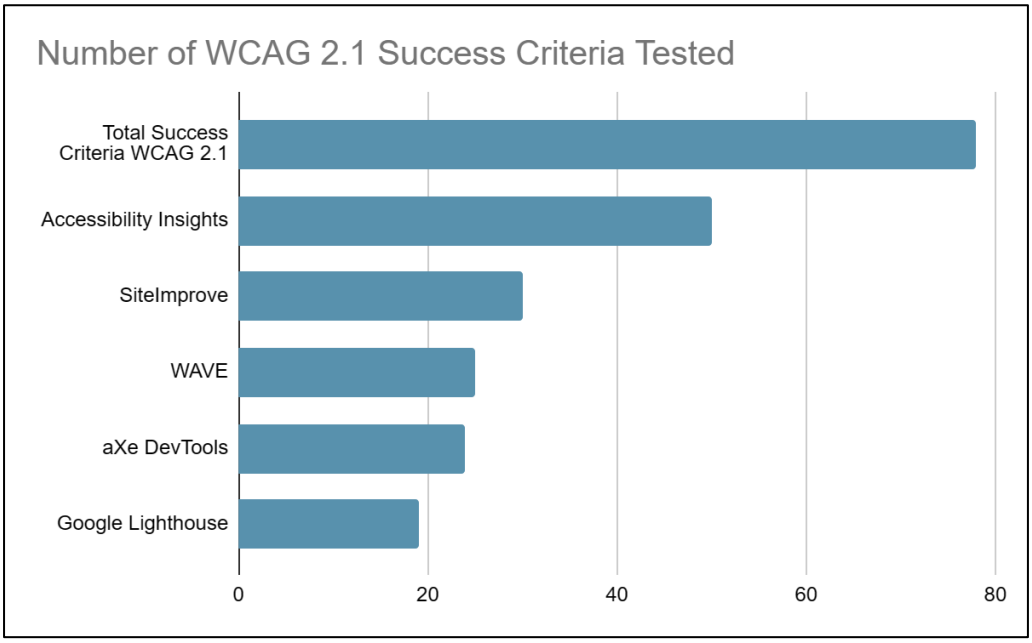
through its browser extension for web apps and its desktop app for Windows software. The tool has a process called FastPass to identify common high-impact accessibility issues within minutes. The tool also has a combination of tests that incorporate automated checks with assisted manual checks for more thorough testing.

Lastly, Axe DevTools by Deque Systems, is a popular accessibility testing tool that can only be used in a browser's "Inspect" mode. According to Axe, "you can find on average 57% of WCAG issues automatically" and it will return components as incomplete if the tool can't be certain and there needs to be manual review (DequeLabs 2024) [5]. It leverages the powerful Axe-core engine, known for its accuracy and reliability, to help developers and testers identify and fix accessibility issues in websites. For each identified issue, Axe provides a clear explanation of the problem, how it impacts users, and offers recommendations on how to fix it.

Table 2. Features of evaluation tools

Tool	Compliance	Version	Features
aXe DevTools	WCAG 2.1 Section 508 EN 301 549 ACA, AODA, AMA	4.80.1	Integration with workflow Generating reports Component-level testing
Google Lighthouse	WCAG 2.1 Section 508	100.0.0.4	Detailed reporting Multiple usage options Integration with dev tools
Microsoft Accessibility Insights	WCAG 2.1	2.41.0	Live inspect Generating reports Guided assessments
WAVE	WCAG 2.1 Section 508	3.2.5.3	Generating reports Visual indicators Prioritized issues
SiteImprove	WCAG 2.1 Section 508 Section 752	2.0	Remediation guidance Generating reports Content quality assurance

Figure 2. Number of WCAG criteria tested by each evaluation tool



Website Selection

To assess the coverage of the different automated accessibility evaluation tools, I selected four of the top news websites according to YouGov, an international online research data and analytics group. NBC News, NPR, Fox News, and the New York Times (NYT) were chosen to represent a diverse range of content and audience demographic. This allows for a broader evaluation of how the tools handle different content types and potential accessibility issues. Additionally, The Bellingham Herald, a local news website, was included to represent a different content structure and audience focus compared to the national news sites.

Evaluation Criteria

To assess the effectiveness of the automated accessibility evaluation tools, I focused on two key factors: a weighted coverage error ratio (CER) and overall usability of the tool. I utilized the CER metric proposed by Alsadee (2020) which gave a ratio for each tool by dividing the number of accessibility issues identified by one tool by the number of accessibility issues identified by

all the tools. The higher the CER the better the tools are at identifying potential issues. One modification I made to this metric to help give a more nuanced view of effectiveness was to assign weights based on WCAG conformance level (A, AA, AAA).

WCAG level AAA represents the highest standard of accessibility, but the least immediate issues, therefore it received the highest weight of 1. Level AA issues have a medium weight of 2 and level A issues, which are the bare minimum and essential for every website, have the weight of 3. By multiplying the number of identified issues at each WCAG level by their corresponding weight and summing the results, I obtained a weighted CER score for each tool. It is important to consider that assigning weights is subjective and may need adjustments based on the specific context and user needs. Additionally, the weighted CER like the original CER "focuses on false negatives and benefits from all issues reported by the tools" (Alsadee 2020). This means that tools that will be considered more effective if it can detect more issues.

To analyze strengths and weaknesses of each tool from a usability standpoint, I used insights and experiences gathered while I was using each tool. Some factors were how easy was it to install and set up the tool, does the tool offer clear documentation, whether or not the tools provided solutions for issues, and is the interface intuitive and easy to navigate.

Testing Environment

In order to have a controlled and comparable testing environment, I used the Edge web browser developed by Microsoft. Specifically, version 125.0.2535.51(64-bit) was used. This choice eliminated potential variations that might come up from using different browsers. Additionally, all testing tools will be launched within a few seconds of each other. This tight timeframe minimized the change of external factors, like web page updates or server load fluctuations, influencing the results. By having maintained consistency in the browser and launch timing, I aimed to isolate the specific variables being tested and obtain the most accurate data possible.

Research Results

Usability Analysis

In this section, I analyzed the usability of each tool, mainly their strengths and weaknesses. There was no formal evaluation procedure as this analysis was based on my insights and experience using the tools during my study. However, for each tool, I looked at factors like ease of use, functionality, and support resources available.

I. Accessibility Insights

Accessibility Insights'(AI) strength is that it was very easy to use and with a good user interface. Aside from its large coverage of WCAG success criteria, AI has a lot of different assessment options to select from. Each one has fast automated checks along with guided manual checks. Another strength of this tool is that it shows visually where the errors are on each page. In addition to the browser extension, there is also a Windows desktop version of AI to evaluate the accessibility of desktop applications. Aside from having a lot of strengths, AI also has limitations, the biggest being that it is only available on the Chrome and Edge browsers. This severely limits the tools reach, excluding testing on browsers like Firefox, Safari, and Opera which are used by millions of people.

II. aXe DevTools

This tool's strength is that it is integrated with the developer tools in most modern browsers like Chrome and Firefox. It is quite easy for developers to use and has an intuitive user interface. aXe gives a score at the end of their automated checks giving users a general idea of a website's overall accessibility. There is a plethora of documentation and tutorials available as well. As for the weaknesses, non-technical users may have trouble getting to the tools if they do not have experience with the browser developer tools. Some features like scanning specific parts of a page are also locked behind a paywall. There were also times where the tools crashed during testing and the browser had to be refreshed to start again.

III. Lighthouse

This tool is similar to aXe with both being built into the developer tool of the browser and gives a score at the end of automated checks. Lighthouse gives users easy access to success criteria documentation from the WCAG and generates a report of all the tests that were failed as well as the ones that passed. Lighthouse's interface is quite lackluster and the process of beginning an accessibility evaluation is quite difficult to figure out for non-technical users. Lighthouse isn't a tool meant solely for checking accessibility conformance but also SEO and performance. There are a lot of options and fields that may cause confusion. Additionally, this tool takes significantly longer to scan a page than the rest in the study, there were several times throughout testing that the tool states that the results were incomplete due the scanning process timing out.

IV. SiteImprove

This tool excels at identifying and explaining accessibility issues on a given page. It provides detailed reports on the problems including the priority of the issues and their locations. It has a very user-friendly dashboard and users can customize the dashboard to suit their needs. SiteImprove not only provides solutions for WCAG issues, but it also helps users implement ARIA and industry best practices. As for limitations, there were several times during testing where the tool removed all the CSS of a page and did not have an option to reapply it. Also, the downside to SiteImprove checking for additional criteria like WAI-ARIA and SI best practices, there is often an overwhelming amount of results in the report that may cause confusion for beginner users.

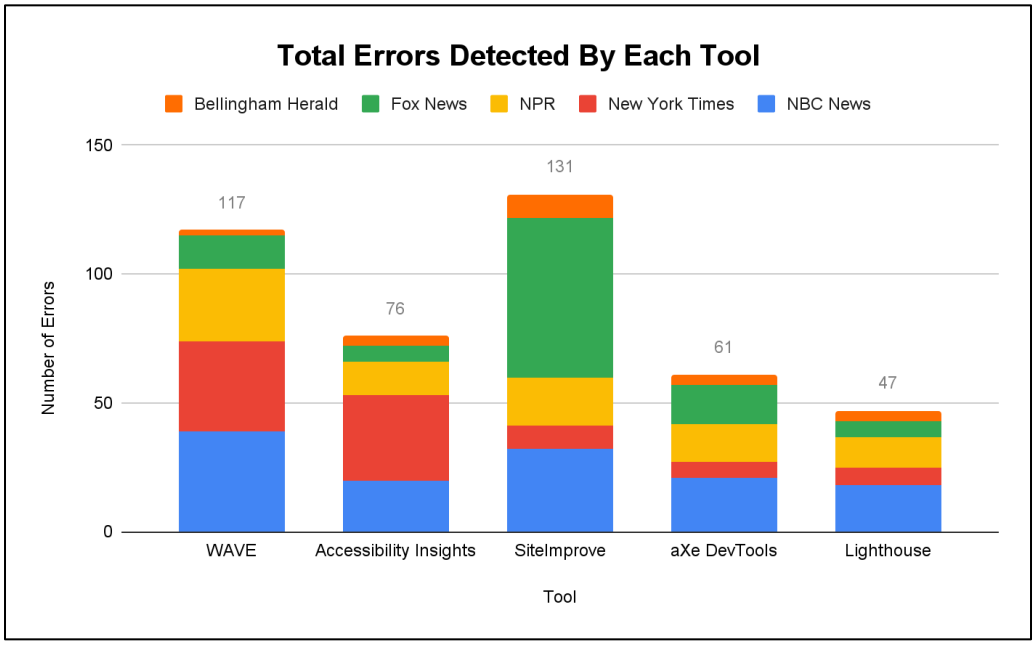
V. WAVE

WAVE gives users direct access to success criteria documentation and even offers solutions to fix the identified problems. It includes features like turning off the CSS of a page, the ability to look at the HTML of the page, and a color contrast checker directly in the tool. All errors, warnings, and structural elements are signified by over 100 icons on the screen making it easy to identify where something resides on a page and identify what it is. However, the problem with this is that there are often too many icons on the screen, making the tool overwhelming for new users.

Result Analysis

I. Total Errors

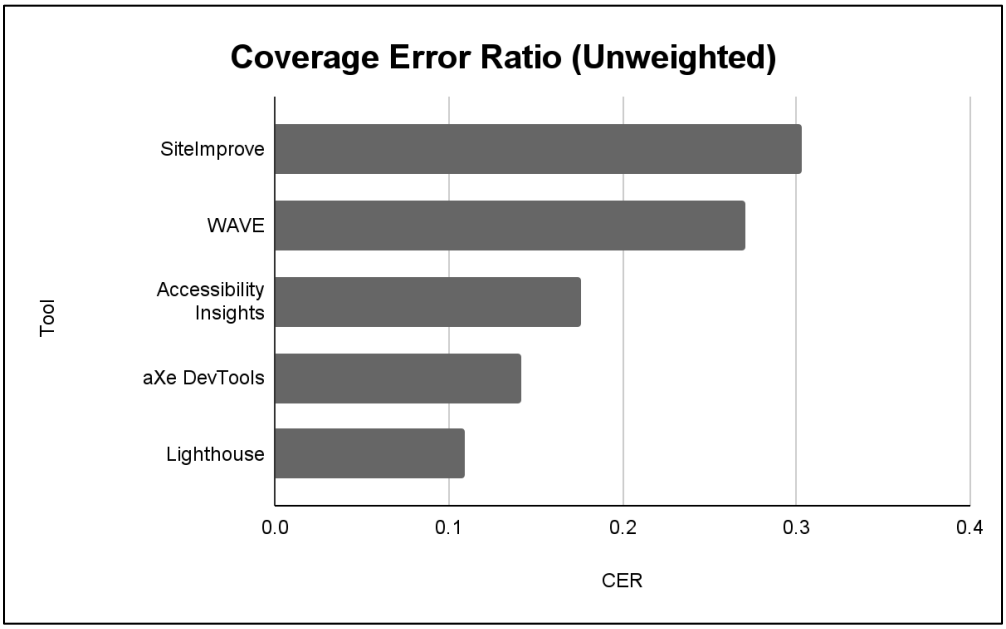
Figure 3. Total Errors Detected By Each Tool



This stacked column chart shows the total number of accessibility issues detected by each tool. Lighthouse and aXe DevTools were found the least and second least amount respectively, this could have been due to their limited WCAG coverage as shown in Figure [2]. Accessibility Insights, despite having the highest coverage in terms of WCAG success criteria, produced the third most issues. It is clear that SiteImprove was able to identify the most amount of issues with the majority of the issues coming from Fox News. This was due to SiteImprove's coverage of criteria including level AAA criteria thus resulting in more issues that may not be as severe.

II. Coverage Error Ratio

Figure 4. Coverage Error Ratio of Each Tool (Unweighted)



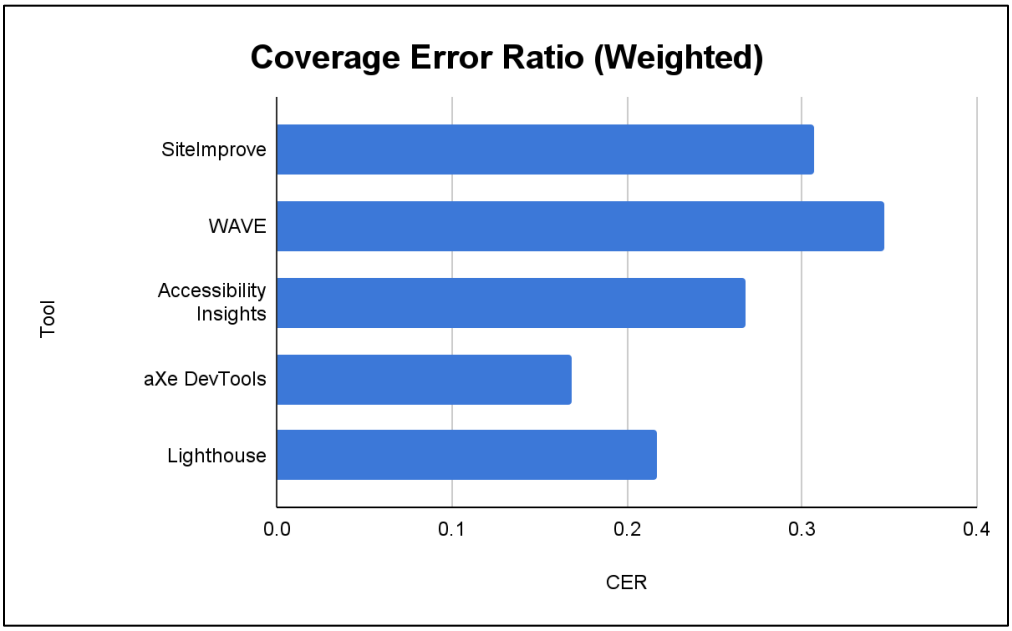
Utilizing Alsaeedi's proposed formula for the coverage error ratio (CER):

$$CER = \frac{Number\ of Errors\ Detected\ By\ a\ Given\ Tool}{Total\ Number\ of\ Errors\ Detect\ By\ All\ Tools}$$

I calculated the CER for each of the tools with the data I gathered during my research and consolidated the ratios into Figure [4]. Some tools are much better at detecting errors than others. SiteImprove is the most effective at detecting issues amongst the ones analyzed with a ratio of 0.303, with WAVE following close behind with a ratio of 0.271. Something of significance to note is that Accessibility Insights' CER of 0.176 is very middle of the pack despite its wide coverage of the WCAG criteria. aXe DevTools and Lighthouse's low ratio of 0.141 and 0.109 respectively was in line with their low coverage of the criteria. There was quite a big gap between the highest and lowest CER reflecting the discrepancy in WCAG criteria coverage. This revealed a connection between the coverage of WCAG success criteria and their CER, with Accessibility Insights being an outlier to the pattern. The CER offered a baseline for comparing different tools, however it treated all issues the same regardless of severity. In the next section, I applied a weight to issues with greater impact on accessibility which provided a more nuanced view of the data.

III. Coverage Error Ratio (Weighted)

Figure 5. Coverage Error Ratio of Each Tool (Weighted)



Building upon the foundation laid by Alsaeedi's (2020) coverage error ratio metric, I introduced a weight to each accessibility issue. As a result, there were some significant changes in the CER of three out of the five tools. The tool with the biggest change in CER was Lighthouse, increasing 0.108. This showed that although Lighthouse wasn't able to identify as many issues as the other tools, the ones that it did find were quite critical, increasing its overall effectiveness. WAVE and aXe DevTools' ratio also increased by a notable amount, showing that these tools prioritized critical issues in level A over a wide coverage. SiteImprove, surprisingly, changed very little, only increasing 0.004. This revealed that SiteImproved prioritized finding a broad range of accessibility issues regardless of conformance level. When there was weight attached to each issue, WAVE became the most effective tool when it comes to identifying accessibility issues with a new CER of 0.347 and aXe dropped to the least effective tool with a score of 0.168.

Discussion

In this paper, I contributed to the methods of evaluating automated accessibility evaluation tools (AAETs) by expanding upon the coverage error ratio (CER). The weighted CER offered a more

nuanced view of effectiveness, prioritizing tools that excel at identifying WCAG level A conformance issues that hinder user experience. This approach acknowledged the varying severity of accessibility issues based on WCAG conformance levels (A, AA, AAA). Based on the findings in my research of the five AAETs, there was a definite tradeoff between coverage and prioritization. Tools like SiteImprove identified a broad range of issues but may not prioritize the most severe ones as highly. Conversely, WAVE with a higher weighted CER focused on critical issues but missed out on less severe ones like enhanced contrast and visual presentation. The ideal balance between coverage and prioritization depends on the context.

While the weight CER offered valuable insights, it did come with limitations. The assigned weights (1 for AAA, 2 for AA, 2 for A) are subjective. The importance of specific WCG levels could be debated on the context and target audience of the website. The weighting also considers WCAG level and not the specific severity of the issue within that level. A minor A issue might be less critical than a major WCAG AAA issue. Another limitation of this paper was that the usability analysis that I conducted, although it provided valuable feedback, was subjective and didn't follow any formal evaluation procedures.

After I examined all these five AAETs, it was still apparent that these tools are still quite limited in what they can do. These tools can't test for everything because many accessibility guidelines rely on human judgment and can't be automatically tested. The metric that was utilized rewards more issues being identified and doesn't care about accuray. These tools also can't understand the context of the content and might not be able to tell if images need alternate text to convey information or if the order of elements on a page makes sense. While these tools are valuable, they should still be used in conjunction with manual testing to get a complete picture of a page's accessibility.

Best Practice

Manual accessibility testing is done by human testers to assess websites and applications for accessibility issues. Often these testers are experts in accessibility guidelines such as the Web Content Accessibility Guidelines. They navigate every element on every page to test functionalities, ensuring that it can be easily accessed and utilized by all users. Additionally,

manual testers consider the broader context of the product and its overall accessibility consistency.

While it is thorough, manual testing has its limitations. It demands significant time and resources, especially for large or frequently updated products. The testing also relies on each individual testers' expertise and can influence the consistency of results. Additionally, human testers are susceptible to errors, which can lead to inaccurate assessments and potentially overlook crucial accessibility issues.

Oftentimes, the optimal approach to accessibility testing involves a combination of both methods. Automated testing can provide a quick and comprehensive initial scan, flagging significant issues early in the development process. Meanwhile, manual testing offers in-depth analysis, focusing on complex issues, user experience, and edge cases that automated tools may overlook.

A good approach for the accessibility testing process is to utilize automated testing during the early "development process to identify and address significant accessibility issues quickly" (CX Score 2023) [21]. Since they can be run quickly and repeatedly, this provides developers with immediate feedback, allowing them to fix issues early on before they become too complex and expensive to address later. Incorporating manual reviews to automated testing in the middle stages of development is beneficial to test for usability and uncovering edge cases. For the final stages of development, automated testing with in-depth manual review is crucial for polishing the user experience and ensuring quality content.

Figure 6. Accessibility Testing Process

Conclusion

Accessibility is growing more and more important as technology becomes a bigger part of everyday life. Making web content accessible isn't just becoming a regulatory requirement, it's also smart business. Accessible content benefits users with and without disabilities, and allows businesses to reach a wider audience. The W3C and WCAG are continuing to evolve everyday and help maintain a standard that developers can utilize to make more accessible websites. While these tools are valuable assets, they shouldn't be the sole method for accessibility evaluation. They still lack the ability to assess user context, complex interactions, and the overall user experience. Future research could explore a more nuanced weighting system for the CER metric that considers the specific impact of each WCAG issue. Additionally, a more formal evaluation procedure for usability can provide more insights on each tool's user experience. In the end, employing a combination of automated and manual testing is the best way to ensure that the Web is accessible to everyone and remains an inclusive space.

References

[1] Alim, Sophia. “Web Accessibility of the Top Research-Intensive Universities in the UK.” *SAGE Open*, vol. 11, no. 4, Oct. 2021, p. 215824402110566. *DOI.org (Crossref)*, https://doi.org/10.1177/21582440211056614.

[2] Alsaeedi, Abdullah. “Comparing Web Accessibility Evaluation Tools and Evaluating the Accessibility of Webpages: Proposed Frameworks.” *Information*, vol. 11, no. 1, Jan. 2020, p. 40. *DOI.org (Crossref)*, https://doi.org/10.3390/info11010040.

[3] Dias, João, et al. “Automated Evaluation Tools for Web and Mobile Accessibility: Proposal of a New Adaptive Interface Tool.” *Procedia Computer Science*, vol. 204, 2022, pp. 297–304. *DOI.org (Crossref)*, https://doi.org/10.1016/j.procs.2022.08.036.

[4] Department of Transportation. 2013. https://www.federalregister.gov/documents/2013/11/12/2013-26749/nondiscrimination-on-the-basis-of-disability-in-air-travel-accessibility-of-web-sites-and-automated

[5] DequeLabs. 2024. https://github.com/dequelabs/axe-core

[6] Frazão, Tânia, and Carlos Duarte. “Comparing Accessibility Evaluation Plug-Ins.” *Proceedings of the 17th International Web for All Conference*, ACM, 2020, pp. 1–11. *DOI.org (Crossref)*, https://doi.org/10.1145/3371300.3383346.

[7] Henry, Shawn. 2024. https://www.w3.org/WAI/standards-guidelines/wcag/

[8] Kumar, Shashank, et al. “Comparing Ten WCAG Tools for Accessibility Evaluation of Websites.” *Technology and Disability*, vol. 33, no. 3, Aug. 2021, pp. 163–85. *DOI.org (Crossref)*, https://doi.org/10.3233/TAD-210329.

[9] Lecturer, Khujand Polytechnic Institute of Tajik Technical University named after academician M.S. Osimi, 735700 Khujand, Tajikistan, and Siddikjon Gaibullojonovich Abduganiev. “Towards Automated Web Accessibility Evaluation: A Comparative Study.” *International Journal of Information Technology and Computer Science*, vol. 9, no. 9, Sept. 2017, pp. 18–44. *DOI.org (Crossref)*, https://doi.org/10.5815/ijitcs.2017.09.03.

[10] Macakoglu, Sevval Seray, and Serhat Peker. “Web Accessibility Performance Analysis Using Web Content Accessibility Guidelines and Automated Tools: A Systematic Literature Review.” *2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, IEEE, 2022, pp. 1–8. *DOI.org (Crossref)*, https://doi.org/10.1109/HORA55278.2022.9799981.

[11] Manca, Marco, et al. “The Transparency of Automatic Web Accessibility Evaluation Tools: Design Criteria, State of the Art, and User Perception.” *ACM Transactions on Accessible Computing*, vol. 16, no. 1, 1, Mar. 2023, pp. 1–36. *DOI.org (Crossref)*, https://doi.org/10.1145/3556979.

[12] Microsoft. 2024. https://accessibilityinsights.io/docs/web/overview/

[13] Pool, Jonathan Robert. “Accessibility Metatesting: Comparing Nine Testing Tools.” *20th International Web for All Conference*, 2023, pp. 1–4. *arXiv.org*, https://doi.org/10.1145/3587281.3587282.

[14] Sarita, Kumari, et al. “Accessibility of Healthcare Sites: Evaluation by Automated Tools.” *Proceedings of International Conference on Data Science and Applications*, edited by Mukesh Saraswat et al., vol. 287, Springer Singapore, 2022, pp. 625–36. *DOI.org (Crossref)*, https://doi.org/10.1007/978-981-16-5348-3_50.

[15] SiteImprove. 2024.

https://help.siteimprove.com/support/solutions/articles/80000448514-a-guide-to-the-siteimprove-accessibility-checks

[16] Vigo, Markel, et al. “Benchmarking Web Accessibility Evaluation Tools: Measuring the Harm of Sole Reliance on Automated Tests.” *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*, ACM, 2013, pp. 1–10. *DOI.org (Crossref)*, https://doi.org/10.1145/2461121.2461124.

[17] W3C. 2023. https://webaim.org/standards/wcag/checklist

[18] W3C. 2019. https://www.w3.org/WAI/GL/wiki/WCAG_2.x_Priority_levels_discussion

[19] W3C. 2019. https://www.w3.org/TR/UNDERSTANDING-WCAG20/intro.html#:~:text=Understandable%20%2D%20Information%20and%20the%20operation%20of%20user%20interface%20must%20be%20understandable.&text=Robust%20%2D%20Content%20must%20be%20robust,user%20agents%2C%20including%20assistive%20technologies.&text=If%20any%20of%20these%20are,able%20to%20use%20the%20Web.

[20] YouGov. 2024. https://today.yougov.com/ratings/entertainment/popularity/news-websites/all

[21] CX Score. 2023. https://cxscore.ai/a-comprehensive-exploration-of-automated-vs-manual-accessibility-testing/#:~:text=Limitations%20of%20Manual%20Accessibility%20Testing,large%20or%20frequently%20updated%20websites.