Comparing automatic accessibility testing tools

Aki Lempola, Timo Poranen and Zheying Zhang

Tampere University, Finland

Abstract

Accessibility aims to provide services to users with disabilities. With 15% of the world's population living with some form of disability and the increasing aging population, web accessibility is increasingly critical. Recent legislation reinforces this need, requiring accessible websites for all. Web accessibility evaluation ensures that the website conforms to legal requirements and the needs of disabled users. Automatic testing tools play an important role in this process. Previous studies have shown that tools detect a different number of issues. In this paper, we compared three automatic accessibility testing tools: IBM Equal access accessibility checker, LERA, and WAVE. We measured their coverage of WCAG success criteria, scanning speed, and the number of issues detected. Finnish e-commerce sites and a test site with known accessibility issues were used for evaluation. This study highlights the strengths and weaknesses of selected automatic accessibility testing tools. WAVE was the fastest tool to scan pages. IBM Accessibility Checker covered the most WCAG success criteria. The number of detected issues varied depending on the page and the type of accessibility issues present on the page. In five out of six tested pages, IBM Equal Access Accessibility Checker identified the most issues, and WAVE identified the most issues on one of the six pages.

Keywords

web accessibility, WCAG, tool comparison, automatic accessibility testing

1. Introduction

Web accessibility ensures everyone, regardless of ability, can access and use web content. While non-disabled people may easily read, navigate, watch, and listen to media content, disabled people may not access the content in the same way as others. Inaccessible websites exclude individuals from information and services increasingly delivered online.

World Health Organization [1] estimates that about 15% of the world's population lives with some form of disability. Accessibility benefits everyone, not just individuals with disabilities [2]. Aging people may experience deterioration of cognitive and or physical skills and senses, making accessibility design important [3]. Also, proper accessibility design can improve user experience, especially in challenging situations such as noisy environments, bright sunlight, or small screens [4].

Web accessibility evaluation can ensure that the website meets the needs of disabled users and complies with legal requirements. Automatic testing tools play an important role in identifying potential accessibility issues. However, studies have shown variation in detecting these issues [5].

This paper is based on a master's thesis [6] of the first author. In this research, we compare three different automatic accessibility testing tools. This study aims to answer the following research questions:

- RQ 1 What success criteria do automatic accessibility testing tools test?
- RQ 2 Is there a difference between selected tool features?
- RQ 3 Do the tools detect different issues?

The research questions are addressed by conducting a document analysis to identify how the tools communicate the WCAG success criteria they test and by comparing them using Finnish e-commerce sites and a test site with known accessibility issues. This comparison measures their coverage of WCAG success criteria, scanning speed, and the number of issues detected.

TKTP 2024: Annual Doctoral Symposium of Computer Science, 10.-11.6.2024 Vaasa, Finland

akilempola@gmail.com (A. Lempola); timo.poranen@tuni.fi (T. Poranen); zheying.zhang@tuni.fi (Z. Zhang)

0000-0002-4638-0243 (T. Poranen); 0000-0002-6205-4210 (Z. Zhang)

© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

The rest of this article is organized as follows. Section 2 covers Web accessibility and accessibility evaluation. In Section 3, we describe the research method, evaluation tool selection, test website selection, sampling page selection, test process, and comparison metrics. Results are elaborated in Section 4. Then, Section 5 discusses the findings and Section 6 concludes the research.

2. Web accessibility

2.1. Accessibility

The definition of web accessibility is widely discussed in the research, and there are many different definitions with different scopes and natures. Our research adheres to the definition by WAI [7]: "Web accessibility means that websites, tools, and technologies are designed and developed so that people with disabilities can use them. More specifically, people can: perceive, understand, navigate, interact with the Web, and contribute to the Web". Web accessibility is closely related to usability and inclusion when developing a Web that works for everyone.

The World Wide Web Consortium (W3C), an international community that develops web standards to ensure the long-term growth of the Web, launched the Web Accessibility Initiative (WAI). This initiative developed the widely adopted Web Content Accessibility guidelines (WCAG) [8]. The previous version 2.1 was released in 2018, and the latest version 2.2 was released in October 2023. WCAG 2.2 extends the older 2.1 version, and content that conforms to the newer version 2.2 also conforms to the 2.1 version. Thus WCAG 2.x versions are backward compatible [9]. Web accessibility guidelines, checklists, and standards such as Web Content Accessibility Guidelines (WCAG) are used to evaluate accessibility. They are also used in some countries' legislation. For example, the European Union uses WCAG 2.1 conformance levels A and AA as standards for web accessibility [10]. WCAG 2.0 is also ISO (International Organization for Standardization) standard ISO/IEC 40500.

Figure 1 shows the structure of WCAG 2.1. At the top level, WCAG 2.1 is divided into four principles that make the Web accessible. Under each principle, there is a list of guidelines that set basic goals that the authors should follow to make the content accessible. WCAG 2.1 comprises

CEUR-WS.org/Vol-3776/paper05.pdf

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

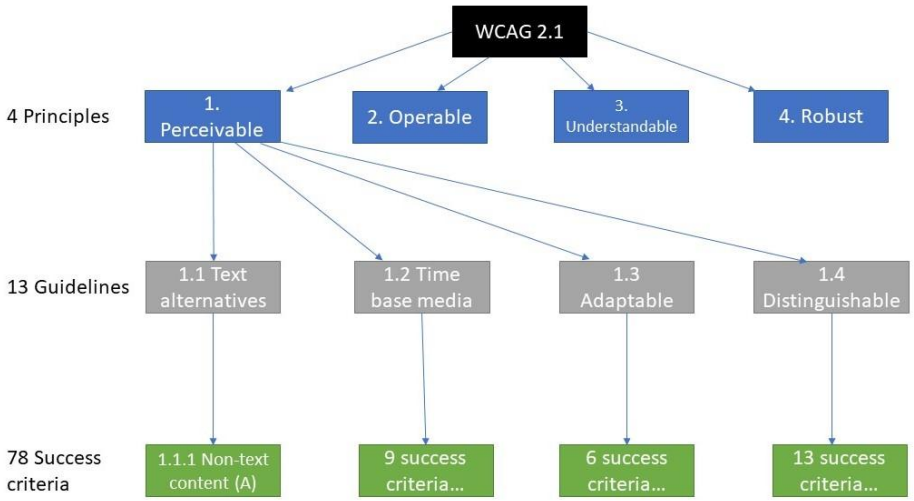

Figure 1: Structure of the WCAG 2.1.

13 guidelines, each of which includes a set of testable success criteria. WCAG 2.1 has 78 success criteria (30 level A, 20 level AA, and 28 level AAA). Each success criterion belongs to one of three conformance levels A (lowest), AA, and AAA (highest). To meet a certain conformance level of WCAG 2.1 website needs to satisfy all success criteria of that level and all levels below it. That means to meet conformance level AA, the site must satisfy all success criteria of levels AA and A. For each of the guidelines and success criteria in the WCAG 2.0, the document provides techniques that are either sufficient to meet the success criteria or advisory that go beyond what is needed to pass the success criteria. Advisory techniques may address accessibility issues that are not covered by any of the success criteria. [9]

The four principles of WCAG 2.1 are: perceivable, operable, understandable, and robust. Under the perceivable principle, there is a total of four guidelines and 29 success criteria. Perceivable means that the content and user interface components must be presented in a way the users perceive them.

The operable principle includes a total of five guidelines and 29 success criteria. Operable means that all user interface components and navigation must be reachable and usable. The understandable principle contains a total of three guidelines and 17 success criteria. Understandable means that the content on the site should be comprehensible for users from different backgrounds, education, and language skills. The robust principle includes one guideline and three success criteria. Robust means that the web pages should be robust enough to work on various user agents.

In Finland, the act on the provision of digital services [11] put in place the accessibility requirements for public service websites and mobile applications. The main target of the act is public sector websites and mobile applications, such as schools and authorities, but also a part of the private sectors such as banks, insurance companies, etc.) are subject to the law.

2.2. Evaluating web accessibility

Web accessibility evaluation is a process that evaluates how well users with disabilities can use the Web. This process aims to find accessibility problems and possibly assess the level of accessibility [12].

Evaluating web accessibility involves assessing its content in two parts: technical content and natural information content [13]. Technical content consists of the markup and code that describes how the content is displayed and how the user interface functions. Natural information content includes the information contained on web pages, text, multimedia, images, etc. Some success criteria for technical content are easy to evaluate automatically with software. For example, WCAG [9] success criterion 1.4.3 for minimum contrast sets minimum requirements for contrast between foreground text and the background. The evaluation shares similarities with software quality assurance, where specific test cases verify the behavior of software in a controlled environment. However, the same rule is not trivial to evaluate when evaluating text in images. It is hard to differentiate between foreground text and background in image data, and often human input is needed. Evaluating natural information content for accessibility is equally important as technical content, even though it is often neglected [13]. Web content is subject to change frequently, while the software is often released in discrete versions that don't change much over time [13].

Accessibility testing tools can be evaluated in at least two ways: using a test suite or selecting a representative sample of websites [14]. Test suites comprise a set of tests where each success criterion is designed to check if a tool detects an intentionally made error. On the other hand, selecting a representative sample of websites allows assessment of the tool's performance across diverse real-world scenarios.

W3C [15] lists 139 automatic tools for evaluating against WCAG 2.0 guidelines and 85 tools for WCAG 2.1 guidelines.

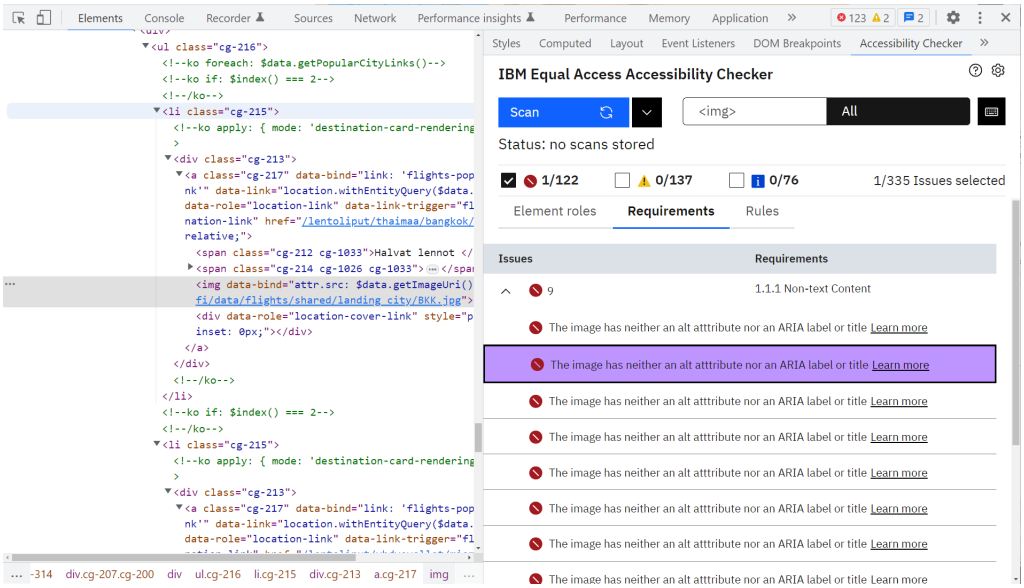

Figure 2: Accessibility test result using IBM Accessibility Checker [18].

The list allows filtering the tools by language, tool type, supported formats, assistive technologies, scope of evaluation, and licenses.

Errors reported by automatic testing tools can greatly differ when testing the same website [5]. Some tools may report the same error multiple times, thus inflating the number of errors [16]. Previous studies [16, 5, 17] recommended to use multiple automatic tools to increase confidence in results.

3. Comparison of accessibility testing tools

To answer RQ 1 and RQ 2, we investigated the tool documentation if the tools are transparent about what success criteria they test. Success criteria coverage was collected from the available tool documentation. We categorized coverage as the tool mapping at least one issue to a success criterion, not necessarily guaranteeing detection of all potential issues within that criterion. In addition, one issue might map to multiple success criteria. To answer RQ 3, tools were tested on three different websites, with a focus on in only automatically detected issues. This excludes warnings and potential accessibility issues that need a human review, recommendations, and best practices not related to WCAG 2.1 success criteria. The study also assumes that all the accessibility issues reported by the tools were real issues.

3.1. Selecting tools

Automatic accessibility evaluation tools were selected from the WAIs [15] web accessibility evaluation tools list page. The vendors and others provide information about the tools on the page. W3C does not endorse specific tools listed on the page. The page can assist in selecting evaluation tools by allowing users to filter according to a wanted tool feature. [15] When we selected the tools, firstly, the list was filtered by selecting the tools that check WCAG 2.1 guidelines. Then the type of the tool was set to browser plugins, and supported formats were set to CSS, HTML, and images. The list of tools was further filtered down by selecting tools that generate evaluation results reports, and the license was set to free software. These filters resulted in a list of five tools. From the result list, three tools were selected for this study. They are IBM Equal access accessibility checker [18], LERA [19], and WAVE [21].

All these accessibility testing tools are Chrome extensions. We used Google Chrome Version 111.0.5563.65 (Official Build) (64-bit) [22], and for IBM Equal Access Accessibility checker Chrome extension version 3.1.46.9999, for LERA we used version 0.5.2, and for WAVE we used the Chrome extension 3.2.3.

The IBM accessibility checker reports accessibility errors in violations, needs review, and recommendations categories. Figure 2 shows a result of an accessibility scan. Violations are errors detected by the tool automatically, needs review are possible violations that need manual review, and recommendations are opportunities to apply best practices.

Figure 3 shows the LERA dashboard. The dashboard shows the number of issues found on the page and the distribution of issues by severity. The automated issues tab shows the issues in a list. After clicking an issue, LERA shows details of the issue, including code snippet, impact, issue tags that show references to guidelines, and recommendations on how to fix the issue. Clicking the eye icon highlights the issue location on the page.

WAVE presents the page with embedded icons and indications. These icons and indicators present some information about the accessibility of the page. Figure 4 shows an example scan result with a missing text alternative issue selected. The WAVE side panel provides a summary of the scan results. Accessibility issues are reported as errors and alerts. WAVE also reports features, structure elements, and ARIA labels detected, that way evaluator can manually review that the features are implemented correctly.

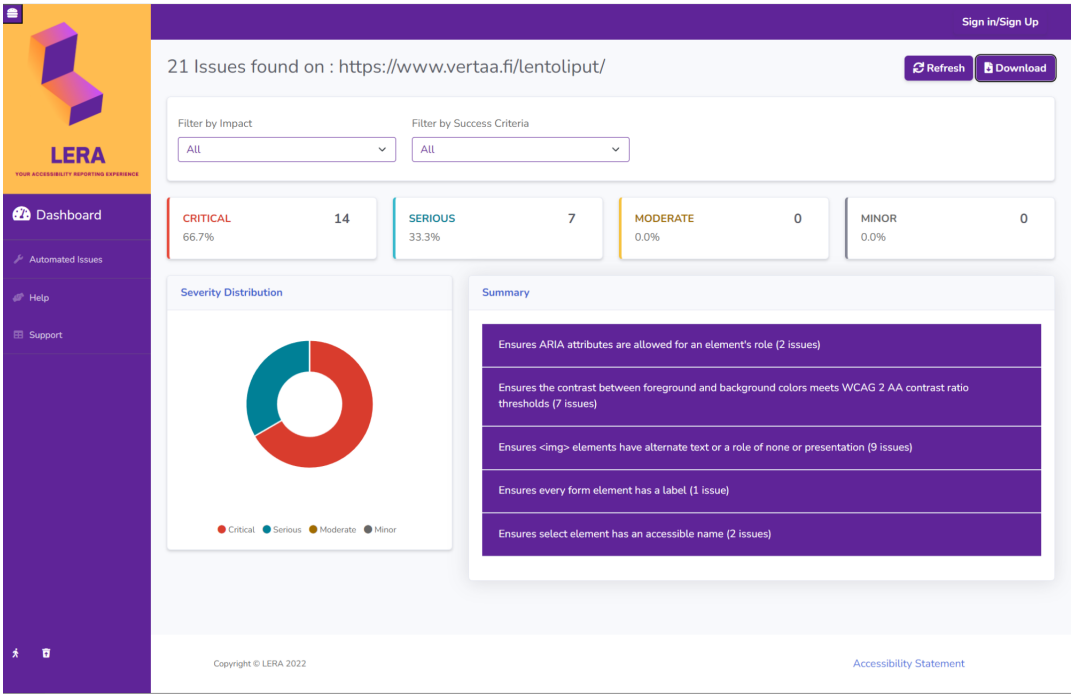

Figure 3: LERA dashboard [19].

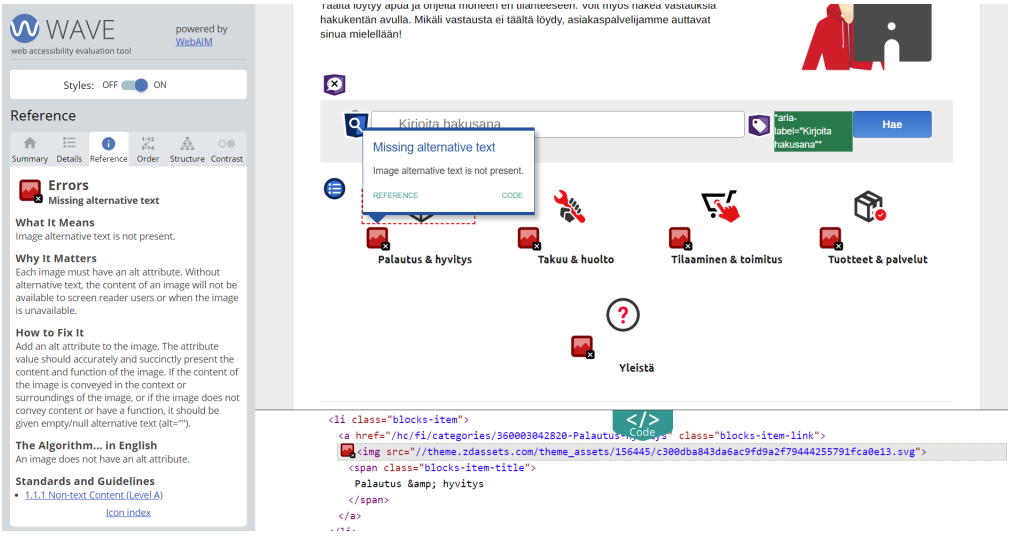

Figure 4: WAVE detecting an image with a missing alternative text [20].

3.2. Selecting websites

To compare the accessibility testing tools objectively, we decided to use two e-commerce websites Vertaa.fi and Verkkokauppa.com, for their diverse content coverage, including images, text, videos, input fields, and forms. In addition, these e-commerce sites do not have specific legal accessibility requirements yet. Additionally, we selected an open-source test suite.

Vertaa.fi is the most popular price comparison site in Finland [23]. It offers a service to compare prices of products but doesn't sell any products itself. The service collects product price information from 243 stores and directs users to search for products and find a store that offers the lowest price.

Verkkokauppa.com is the most visited and well-known retail e-commerce store in Finland [24]. They sell computers, electronics, toys, games, sports products, etc. Verkkokauppa was founded in 1992, and in addition to their e-commerce store, they have 4 retail stores.

To compare the tools' capabilities, we also selected an open-source test suite developed by the Government digital services [25]. The test site has 142 accessibility issues. However, we excluded two tests from our evaluation, because the embedded YouTube video for the flashing content test is no longer available, and the alternative text for an audio file is missing.

3.3. Selecting sampling pages

To ensure diverse scenarios and capture a wider range of potential accessibility issues, we strategically selected pages beyond just the landing pages of each website. A landing page is the first page for most users to visit the site and navigate to the different pages of the site. We assume that developers paid the most attention to minimize errors and accessibility issues. On Verkkokauppa.com we included in the test the landing page, account creation page, and customer service page. On the Vertaa.fi site we tested the landing page and the cheap flights category (halvat lennot) page. On the test suite, most of the tests were on the main page, but some of the tests were linked to different pages. We tested both the main page and test pages directed by links. All the pages were tested on 6.4.2023.

3.4. How the pages were tested

We tested each page with each tool one by one. First, we waited for the entire page to load, and scrolled to the bottom of the page before initiating any tests. Then we tested the same page in the same browser window, with each tool one by one. This minimizes the probability of dynamic content changing between the test of different tools. Lastly, we repeated the steps one more time, to see if the accessibility testing tools produced consistent results, and to collect the data of scan speed of the tools. All the tools reported identical results on both scans on every page we tested. Tools report the detected accessibility issues in different ways. Each tool checks accessibility rules that are connected with one or more WCAG 2.1 success criteria. The study focused on how well issues detected by the automatic testing tools conform to WCAG. Therefore, we collected the total number of accessibility issues and the number of issues in each WCAG success criterion. In some cases, the number of accessibility issues may be less than the sum of WCAG violations, because an issue detected by the tool may be mapped to multiple success criteria by the tool.

3.5. Comparison metrics

We compared the tools on the aspects of efficiency, completeness, and the number of detected issues. Efficiency was measured by scan time, reflecting the time taken to analyze each page. Fast scan time is important for efficiently evaluating large websites with hundreds of pages. Completeness defines how completely the tool covers the WCAG success criteria. The tool is considered to cover a success criterion if the tool performs at least one test on that success criterion. The number of detected issues is the number of automatically detected issues on a page. Issues that needed a human review and recommendations were discarded because we were interested in how well the tools detect issues automatically. While not exhaustive, we also compared a list of features found particularly useful during testing. This comparison helps users identify tools suited to their specific needs.

4. Results

4.1. Success criteria coverage

To evaluate the transparency of each tool, we analyzed their documentation to identify how clearly they communicate the WCAG success criteria they test. This helps us understand how comprehensively users can assess the tool's capabilities. Figure 5 shows the success criteria covered by the tools according to the documentation [26, 18, 20]. In Figure 5, F stands for failure, A for an alert, N stands for needs review, and X means that the tool didn't specify if the tests produce errors or alerts of possible errors.

Our analysis reveals that the selected tools collectively offer tests for issues and warnings across 37 of the 78 WCAG success criteria, which represents 47% of the success criteria coverage. While the success criteria are covered, it is crucial to understand that the coverage does not mean complete testing. It simply means that the tool can detect at least one kind of issue mapped to the success criterion. Tools also map some detected issues to multiple success criteria.

IBM Equal Access Accessibility Checker did not mention if the automatic test for specific WCAG success criteria produces issues or alerts of possible errors. They only reported that the success criteria were automatically tested. But when comparing the coverage of issues and alerts, IBM Accessibility Checker covers the most WCAG 2.1 success criteria 31 out of 78, while WAVE and LERA both cover 22 success criteria when considering both automatically detected issues and alerts of possible issues. According to the tool's documentation, LERA covers the most success criteria with a fully automatic test with 20 out of 78 success criteria, while WAVE detects automatic issues for 13 out of 78 success criteria. The Union of the success criteria covered by the tools shows that the selected tools cover different success criteria. Thus complementing each other. The Union of success criteria covered by issues and alerts is 37, while the single tool with the widest coverage covered 31 success criteria.

4.2. Detected accessibility issues

In this section, we go over the accessibility issues detected by the tools. We found that every tool we tested detected accessibility issues on every tested page. The performance of the tools seems to depend on the page and the type of accessibility issues present on the page. One tool may find the greatest number of issues on one page but the least number of issues on another page.

Figure 6 shows the total number of issues for each success criterion on all tested pages. IBM Accessibility Checker reported the most issues for success criteria 4.1.2 (name, role value), 2.4.1 (bypass block), 2.1.1 (keyboard), and 1.3.1 (info and relationships). WAVE reported the greatest number of issues for success criteria 1.4.3 (contrast), 1.1.1 (non-text content), 2.4.4 (link purpose in context), 2.4.6 (headings and labels), and 3.3.2 (labels or instructions).

4.2.1. Verkkokauppa.com

Figures 7, 8, and 9 show the accessibility issues detected on the tested pages on verkkokauppa.com. Every tool found accessibility issues on all these pages. WAVE found a total of 25 issues, IBM 69 issues, and LERA 17 issues.

Figure 7 shows the accessibility issues detected on the verkkokauppa.com landing page. IBM Accessibility Checker found the most errors, that is 41 accessibility issues were detected, while WAVE detected 12 issues and LERA 3 issues.

IBM Accessibility Checker also found issues in the greatest number of success criteria, finding issues in 5 different success criteria. While IBM found the most issues on the

Success Criteria	IBM	WAVE	LERA (axe-core 4.3.5)
WCAG 1.1.1	X	F, A	F, N
WCAG 1.2.1	X	A	N
WCAG 1.2.2	X	A	N
WCAG 1.2.3		A	
WCAG 1.2.4	X		
WCAG 1.2.5	X	A	
WCAG 1.3.1	X	F, A	F, N
WCAG 1.3.2	X	A	
WCAG 1.3.3	X		
WCAG 1.3.5	X		F
WCAG 1.4.1	X		F, N
WCAG 1.4.2	X	A	F, N
WCAG 1.4.3	X	F	F, N
WCAG 1.4.4	X		
WCAG 1.4.12			F
WCAG 2.1.1	X	F, A	F, N
WCAG 2.1.2	X	A	
WCAG 2.2.1	X	F	F
WCAG 2.2.2	X	F	F
WCAG 2.2.4			F
WCAG 2.4.1	X	F, A	F, N
WCAG 2.4.2	X	F	F
WCAG 2.4.3		A	
WCAG 2.4.4	X	F, A	F, N
WCAG 2.4.6	X	F, A	
WCAG 2.4.7	X		
WCAG 2.4.9			N
WCAG 2.5.3	X		F
WCAG 3.1.1	X	F, A	F
WCAG 3.1.2	X		F
WCAG 3.2.1	X		
WCAG 3.2.2	X	A	
WCAG 3.2.5			F
WCAG 3.3.1	X		
WCAG 3.3.2	X	F, A	N
WCAG 4.1.1	X		F
WCAG 4.1.2	X	F	F, N
Union of covered success criteria: 37	31	22	22
Union of Issues: 22		13	20
Union of Warnings: 22		17	13

Figure 5: WCAG Success criteria covered by the tools according to the documentation [26, 18, 20].

landing page, every tool found issues in the success criterion 1.4.3 low contrast, WAVE found the greatest number of issues for this success criteria 12, IBM 11, and LERA 1 issue.

Accessibility issues identified on the account creation page are presented in Figure 8. Again, IBM Accessibility Checker identified the most issues. In detail, IBM Accessibility Checker found 17 issues, and WAVE and LERA both found 3 accessibility issues. On this page, WAVE found the most issues violating success criteria 1.4.3 for low contrast. WAVE also maps the empty form label rule to 4 different WCAG success criteria, success criteria 1.1.1, success criteria 1.3.1, success criteria 2.4.6, and success criteria 3.3.2, while IBM Accessibility Checker maps this issue to success criteria 4.1.2. LERA and IBM Accessibility Checker found the same number of issues for success criteria 1.3.5 and 1.4.3. Again, WAVE found the greatest number of contrast issues 2, while IBM and LERA found 1. Figure 9 shows the accessibility issues found on the verkkokauppa.com customer service page. On this page, the tools produced the most similar results. All three tools found the same number of violations for success criteria 1.1.1 non-text content, each tool found 10 issues. In addition, LERA and IBM Accessibility Checker produced an identical report of errors on the customer service page. Both tools found 11 issues, and both tools mapped the found to the same success criteria. Every tool scanned the customer service page under a second, because the page was the smallest page of the three tested pages.

4.2.2. Vertaa.fi

Figures 10 and 11 show the results of pages tested on vertaa.fi. The total number of errors on the tested pages were: WAVE 205, IBM Accessibility Checker 234, and LERA 139 accessibility issues between the two tested pages. Table 6 shows the accessibility issues detected with each tool on the vertaa.fi landing page. WAVE detected the greatest number of issues on the landing page with a total of 139 issues were detected, IBM Accessibility Checker detected 112 issues, and LERA 118 issues.

Figure 10 also shows that the WAVE detected the greatest number of accessibility violations in the success criteria

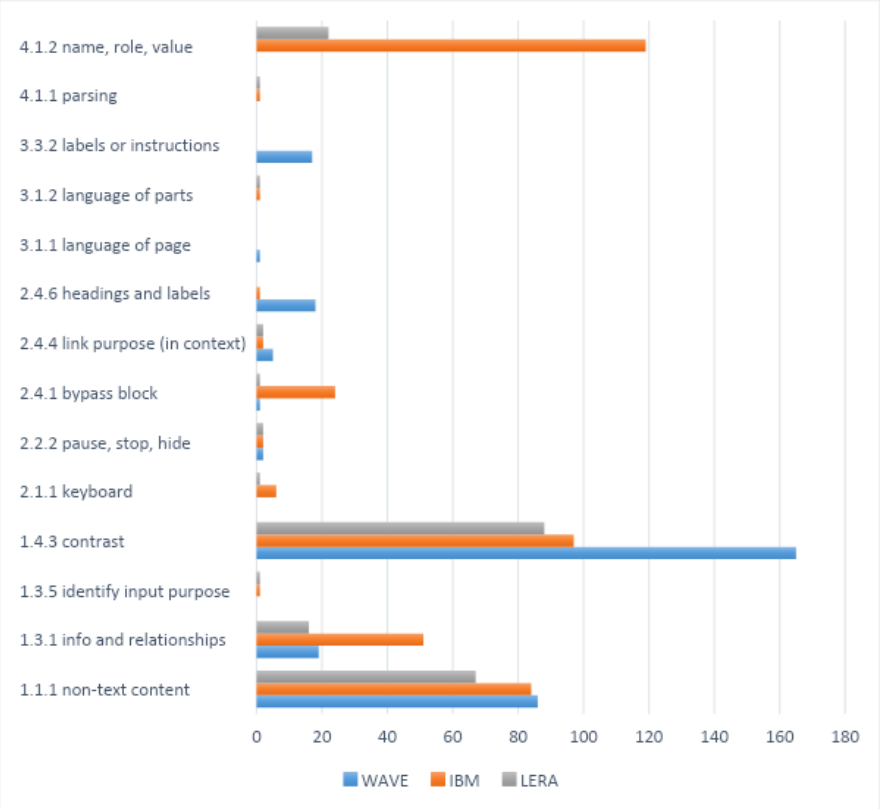

Figure 6: The total number of issues for each success criterion on all tested pages.

Success criteria	WAVE	IBM	LERA
1.1.1 non-text content		14	
1.4.3 contrast	12	11	1
2.1.1 keyboard		2	
2.4.1 bypass blocks		10	
4.1.2 name, role, value		4	2
total issues	12	41	3
Scan time	< 1 sec	21 sec	8 sec

Figure 7: Accessibility issues detected on the verkkokauppa.com landing page.

1.4.3 contrast. Figure 7 shows some contrast issues detected on the vertaa.fi landing page, only the dark blue text has sufficient contrast. IBM Accessibility Checker was the only tool to detect any violations of success criteria 2.4.1. Every tool found 44 violations for the success criteria 1.1.1, but WAVE also maps missing labels to this category, for that reason WAVE reported more success criteria 1.1.1 violations than the two other tools.

Figure 11 shows the accessibility issues detected on the vertaa.fi flight search page. IBM Accessibility Checker found the greatest number of errors on the flight search page, with a total of 122 accessibility issues found, WAVE found 66 issues, and LERA 21. Similarly, to the vertaa.fi landing page, all the tools found the same number of success criteria 1.1.1 violations, but WAVE mapped additionally 3 missing label issues to this category. Hence the larger number of issues for success criteria 1.1.1. WAVE detected the greatest number of success criteria 1.4.3 violations. IBM Accessibility Checker was the only tool to detect issues for success criteria 2.1.1 and 2.4.1. IBM Accessibility checker also detected the greatest number of violations for the success criteria 4.1.2: IBM 97, LERA 5, WAVE 0.

4.2.3. Test suite

Accessibility issues detected on the test are shown in Figure 12. IBM Accessibility Checker detected the greatest number of issues in the test suite, most of the issues were in the success criteria 1.3.1 and these issues were about data Table cells missing header or scope, IBM Accessibility Checker was the only tool that detected these issues.

All the tools found the same number of issues for success criteria 1.4.3 and 2.2.2. LERA and IBM Accessibility Checker found the same number of issues for 7 out of 11 success

Success criteria	WAVE	IBM	LERA
1.1.1 non-text content	1		
1.3.1 info and relationships	1		
1.3.5 identify input purpose		1	1
1.4.3 contrast	2	1	1
2.1.1 keyboard		1	
2.4.1 bypass block		10	
2.4.6 headings and labels	1		
3.3.2 labels or instructions	1		
4.1.2 name, role, value		4	1
total issues	**3**	**17**	**3**
Scan time	**< 1 sec**	**4 sec**	**2 sec**

Figure 8: Accessibility issues detected on the verkkokauppa.com account creation page.

Success criteria	WAVE	IBM	LERA
1.1.1 non-text content	10	10	10
1.3.1 info and relationships		1	1
4.1.2 name, role, value		1	1
total issues	**10**	**11**	**11**
Scan time	**< 1sec**	**< 1sec**	**< 1sec**

Figure 9: Accessibility issues detected on the verkkokauppa.com customer service page.

criteria. For the test of usage of lang attribute for change of language with an invalid value, IBM accessibility checker and LERA mapped the issue to success criteria 3.1.2, while WAVE mapped this issue to success criteria 3.1.1.

4.3. Summary of test results

WAVE reported the most issues for the success criterion 1.4.3 low contrast on every page we tested. While testing the test suite, all the tools detected all the contrast tests in the test suite. This implies that the use of test suites doesn't necessarily imitate the real issues on real pages. Across the 6 pages we tested, IBM Accessibility Checker detected the most accessibility issues overall. However, WAVE surpassed it in finding issues on one specific page, demonstrating the value of considering multiple tools for diverse scenarios.

IBM Accessibility Checker was the only tool to report issues for success criteria 2.1.1 and 2.4.1 on the real pages, while LERA and IBM Accessibility Checker detected issues for these categories in the test suite, and WAVE detected issues for the 2.4.1 in the test site.

While all tools detected the mentioned accessibility issues, there are inconsistencies in how they mapped these issues to specific WCAG success criteria. For example, WAVE maps

Success criteria	WAVE	IBM	LERA
1.1.1 non-text content	47	44	44
1.3.1 info and relationships	3		
1.4.3 contrast	92	67	74
2.4.1 bypass blocks		1	
2.4.6 headings and labels	3		
3.3.2 labels or instructions	3		
total issues	**139**	**112**	**118**
Scan time	**< 1sec**	**7 sec**	**6 sec**

Figure 10: Accessibility issues detected on the vertaa.fi landing page.

Success criteria	WAVE	IBM	LERA
1.1.1 non-text content	12	9	9
1.3.1 info and relationships	3		3
1.4.3 contrast	54	13	7
2.1.1 keyboard		2	
2.4.1 bypass block		1	
2.4.6 headings and labels	3		
3.3.2 labels or instructions	3		
4.1.2 name, role, value		97	5
total issues	**66**	**122**	**21**
Scan time	**< 1 sec**	**5 sec**	**3 sec**

Figure 11: Accessibility issues detected on the vertaa.fi flight search page.

Success criteria	WAVE	IBM	LERA
1.1.1 non-text content	16	7	4
1.3.1 info and relationships	12	50	12
1.4.3 contrast	5	5	5
2.1.1 keyboard		1	1
2.2.2 pause, stop, hide	2	2	2
2.4.1 bypass block	1	2	1
2.4.4 link purpose (in context)	5	2	2
2.4.6 headings and labels	11	1	
3.1.1 language of page	1		
3.1.2 language of parts		1	1
3.3.2 labels or instructions	10		
4.1.1 parsing		1	1
4.1.2 name, role, value		13	13
total issues	**26**	**85**	**31**

Figure 12: Accessibility issues detected on the test suite.

an issue of an empty or missing form label to four success criteria 1.1.1 non-text content, 1.3.1 info and relationships, 2.4.6 headings and labels, and 3.3.2 labels and instructions. IBM Accessibility Checker maps the same issue to a success criterion 4.1.2 name, role, and value. Additionally, LERA maps the missing form label to two success criteria 1.3.1 info and relationships and 4.1.2 name, role, value. Another example is about an image link with no alternative text. WAVE maps this to criteria 1.1.1 non-text content and 2.4.4 link purpose, but IBM Accessibility Checker just maps this criterion 2.4.4 link purpose. In addition, LERA maps this issue to two criteria 2.4.4 link purpose and 4.1.2 name, role, value. While consistent issue detection is crucial, these discrepancies in WCAG mappings can be confusing for users, especially those relying on the tools for compliance guidance.This highlights the importance of considering not only the number of issues detected but also how tools interpret and categorize them. Users should be aware of potential mapping inconsistencies and may need to consult additional resources for definitive WCAG compliance assessments.

The number of accessibility issues detected by each tool depends significantly on the selected pages and the type of issue present. One tool might detect more accessibility issues on one page than another and fewer issues on another page, depending on the type of accessibility issues on the page. Out of the selected tools WAVE appears to be best at detecting issues for success criterion 1.4.3 low contrast, while IBM accessibility Checker appears to detect most issues for success criteria 4.1.2 name, role, value, 2.4.1 bypass block, and 2.1.1 keyboard on the tested pages. If one tool is better at detecting one type of accessibility issue than other tools, and then if this type of issue is prominent on the page, then that tool is going to detect more issues on the page. As can be seen in Figures 10 and 11 vertaa.fi pages, WAVE detected more issues on the landing page and IBM on the flight search page. For that reason, using multiple automatic testing tools is recommended.

As for the scan time, WAVE was the fastest tool, LERA was the second fastest, and IBM Accessibility Checker was the slowest of the selected tools. Regarding the average scan time per tested pages, IBM Accessibility checker was over 7 times slower than WAVE and LERA was 4 times slower than WAVE.

Feature	WAVE	IBM	LERA
Show navigation order	yes, shows what screen reader reads	yes	no
Toggle styles	yes	no	no
Change rule set	no	yes	no
Map issues to WCAG 2.1 standards	yes	yes	yes
Highlight issues on the page	yes	yes	yes
Highlight issue in the code	yes	yes, in dev tools	code snippet
Instructions how to fix the issue	yes	yes	yes
Show issues on selected part of the page	no	yes	no
Semiautomatic checks	yes	yes	no
Last update	17.3.2023	5.4.2023	11.5.2022

Figure 13: Tool features.

4.4. Tool features

Tool features were gathered while using the tools to scan pages. Features listed in Figure 13, are not a comprehensive list of all the features of the tools, rather they are the ones we identified useful while using the tools. The tools showing navigation order can be a useful feature. Especially the way WAVE implemented this feature. WAVE shows what the screen reader says. This can be useful for understanding the functionality of screen readers, without the need to install and learn to use a screen reader. Another useful feature implemented by WAVE is to toggle the styles, this feature can help to find accessibility issues hidden with styles. IBM Accessibility Checker is the only tool to allow changing between rulesets. IBM Accessibility Checker is the only tool that allows one to select an element on the page and show the issues on the selected element. This feature can be useful if the page has a large number of issues. It may be easier to select a part of the page and fix issues that way, instead of going over an overwhelming number of issues.

All three selected tools map the detected issues to the WCAG 2.1 guidelines, this allows the user to seek more information about the issue. All the selected tools highlight issues on the page. All the tools also provide instructions on how to fix accessibility issues.

5. Discussion

In this research we compared three automatic accessibility evaluation plugins for Google Chrome in terms of efficiency, WCAG success criteria covered, and issues detected. The selected tools were WAVE, IBM Equal Access Accessibility Checker, and LERA. With the comparison of the tools, we deepened our understanding of the automatic accessibility evaluation tools, what these tools can test in terms of WCAG, what issues they found in Finnish e-commerce sites, and whether are there differences among the selected tools.

Regarding WCAG success criteria covered, we found that the combination of the tools covered 37 success criteria out of 78. This is more than any single tool, alone IBM Accessibility Checker covered 31 success criteria, and WAVE and LERA each covered 22 success criteria. From this, we can see that the tools cover not only a different number of success criteria but also different success criteria. Thus, the tools complement each other.

In terms of the number of issues detected, results depend on the scanned page. More precisely the number of issues detected by the tool depends on the types of accessibility issues present on the page. Most of the issues are in perceivable principle. Out of the scanned pages, success criterion 1.4.3 low contrast issues have the greatest impact on the results. IBM Accessibility Checker detected the greatest number of accessibility issues on five out of six scanned pages, while WAVE detected the most issues on one out of the six scanned pages. IBM Accessibility Checker detected the greatest number of issues on the most of tested pages, it also detected the least number of issues on one tested page.

As to types of accessibility issues detected, WAVE detected the greatest number of issues for four success criteria and IBM Accessibility Checker for four success criteria. Although the accessibility issues detected for each success criterion may not be a reliable metric for measuring the tool performance, the tools seem to map detected issues to the WCAG differently. WAVE tends to map an issue from one to four success criteria, while IBM Accessibility Checker maps these issues to one success criterion, and LERA maps issue one to two success criteria.

The results align with the previous studies of automatic accessibility evaluation tools [16, 5, 17]. The tools selected for this research cover different success criteria and complement each other. The usage of a combination of different automatic accessibility testing tools yields better results than using a single tool. Thus, it is recommended to use more than one automatic accessibility evaluation tool. It is

also important to keep in mind, that the automatic accessibility testing tools cannot detect all accessibility issues. Many accessibility requirements need human interpretation. It's not possible to determine conformance to the guidelines with automatic tools alone.

The method of this research has its limitations. Firstly, we used only automatic accessibility evaluation tools, these tools can only detect a part of accessibility issues present on a page. And in this research, we were only interested in the automatically detected issues. Further limiting the number of the issues these tools can detect, as we discarded all issues that needed manual review. Secondly, we assumed that all the issues reported by the tools are true positives. These limitations may reward a tool that reports more issues with a cost of accuracy and penalizes tools that are more conservative and attempt to report only real accessibility issues.

6. Conclusion

In this study, we covered three automatic accessibility evaluation tools. We tested them on two different websites and a test site. Future studies could expand the number of tools and the number of tested pages and include more different types of websites, to gain more confidence in the results. Different types of automatic accessibility tools could be included.

Future studies could also analyse alerts of potential issues, to find out if there are differences between tools. Does one tool report an accessibility issue as a detected issue, and do other tools then report the same issue as an issue that needs manual review? Future studies could also manually analyse the automatically detected issues to compare the accuracy of the tools. A comparison against a manual conformance review of the page could also be made to analyse how well the automatic tools detect issues compared to an expert evaluator.

References

[1] World Health Organization, World report on disability summary, https://www.who.int/publications/i/item/WHO-NMH-VIP-11.01, Accessed: 2024-5-31.

[2] S. Schmutz, A. Sonderegger, J. Sauer, Implementing recommendations from web accessibility guidelines: would they also provide benefits to nondisabled users, Human factors 58 (2016) 611–629.

[3] J. T. Richards, V. L. Hanson, Web accessibility: a broader view, in: Proceedings of the 13th international conference on World Wide Web, 2004, pp. 72–79.

[4] WAI, Accessibility, Usability, and Inclusion, https://www.w3.org/WAI/funda-mentals/accessibility-usability-inclusion/, Accessed: 2023-5-8.

[5] R. Ismailova, Y. Inal, Comparison of online accessibility evaluation tools: an analysis of tool effectiveness, IEEE Access 10 (2022) 58233–58239.

[6] A. Lempola, Comparing automatic accessibility testing tools, Master's thesis, Tampere University, 42 pages, available at: https://trepo.tuni.fi/handle/10024/148622 (2023).

[7] WAI, Accessibility intro, https://www.w3.org/WAI/fundamentals/accessibility-in-tro/, Accessed: 2023-6-9.

[8] W3c accessibility standards overview, https://www.w3.org/WAI/standards-guidelines/wcag/, Accessed: 2023-5-31.

[9] Web content accessibility guidelines (WCAG) 2.1, https://www.w3.org/TR/WCAG21/, Accessed: 2023-7-15.

[10] European Commission, Directive (EU) 2016/2102 of the European Parliament and of the Council of 26 October 2016 on the accessibility of the websites and mobile applications of public sector bodies, https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016L2102, Accessed: 2024-5-31.

[11] Laki digitaalisten palveluiden tarjoamisesta. Valtiovarainministeriö, https://www.finlex.fi/fi/laki/smur/2019/20190306, Accessed: 2024-1-31.

[12] G. Brajnik, Y. Yesilada, S. Harper, The expertise effect on web accessibility evaluation methods, Human–Computer Interaction 26 (2011) 246–283.

[13] S. Aboy-Zahra, Web accessibility and guidelines, in: S. Harper, Y. Yesilada (Eds.), Web Accessibility a Foundation for Research, Springer Science & Business media, 2008, pp. 79–106.

[14] M. Vigo, J. Brown, V. Conway, 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, 2013, pp. 1–10.

[15] W3C, Web accessibility evaluation tools list, https://www.w3.org/WAI/ER/tools/, Accessed: 2023-7-15.

[16] M. Padure, C. Pribeanu, Comparing six free accessibility evaluation tools, Informatica Economica 24 (2020) 15–25.

[17] T. Frazão, C. Duarte, Comparing accessibility evaluation plug-ins, in: Proceedings of the 17th International Web for All Conference, 2020, pp. 1–11.

[18] IBM Equal Access Accessibility Checker, https://www.ibm.com/able/toolkit/tools/, Accessed: 2024-1-31.

[19] LERA - Website Accessibility Testing & Reporting Tool, https://advancedbytez.com/lera/, Accessed: 2024-1-31.

[20] Keyboard accessibility, https://webaim.org/techniques/keyboard/, Accessed: 2023-7-15.

[21] WAVE web accessibility evaluation tools, https://wave.webaim.org/, Accessed: 2024-1-31.

[22] Google, Google Chrome., https://www.google.com/chrome/, Accessed: 2023-3-21.

[23] Vertaa, Vertaa.fi, https://www.vertaa.fi/info/info/, Accessed: 2023-4-3.

[24] Verkkokauppa, Verkkokauppa.com. Yritystiedot., https://www.verkkokauppa.com/fi/yritystiedot, Accessed: 2023-4-3.

[25] Government Digital Services. Accessibility tool audit., https://alphagov.github.io/accessibility-tool-audit/test-cases.html, Accessed: 2023-4-3.

[26] Deque, Deque Labs. 2023c. Rule descriptions., https://github.com/dequelabs/axe-core/blob/4937bfa4f8d689f81fb89c71d6a292fcbdba767b/doc/rule-descriptions.md, Accessed: 2023-3-21.