Benchmarking web accessibility evaluation tools: Measuring the harm of sole reliance on automated tests

Conference Paper · May 2013

DOI: 10.1145/2461121.2461124

Markel Vigo
The University of Manchester

Vivienne Conway
Web Key IT Pty Ltd

Benchmarking Web Accessibility Evaluation Tools: Measuring the Harm of Sole Reliance on Automated Tests

Markel Vigo*
School of Computer Science
University of Manchester
Manchester (UK)
markel.vigo@manchester.ac.uk

Justin Brown
School of Computer and Security Science
Edith Cowan University
Perth (Australia)
j.brown@ecu.edu.au

Vivienne Conway
School of Computer and Security Science
Edith Cowan University
Perth (Australia)
v.conway@ecu.edu.au

ABSTRACT

The use of web accessibility evaluation tools is a widespread practice. Evaluation tools are heavily employed as they help in reducing the burden of identifying accessibility barriers. However, an over-reliance on automated tests often leads to setting aside further testing that entails expert evaluation and user tests. In this paper we empirically show the capabilities of current automated evaluation tools. To do so, we investigate the effectiveness of 6 state-of-the-art tools by analysing their coverage, completeness and correctness with regard to WCAG 2.0 conformance. We corroborate that relying on automated tests alone has negative effects and can have undesirable consequences. Coverage is very narrow as, at most, 50% of the success criteria are covered. Similarly, completeness ranges between 14% and 38%; however, some of the tools that exhibit higher completeness scores produce lower correctness scores (66-71%) due to the fact that catching as many violations as possible can lead to an increase in false positives. Therefore, relying on just automated tests entails that 1 of 2 success criteria will not even be analysed and among those analysed, only 4 out of 10 will be caught at the further risk of generating false positives.

Categories and Subject Descriptors

H.5.2 [**User Interfaces**]: Benchmarking, Evaluation / methodology; H.5.4 [**Hypertext/Hypermedia**]: User issues; D.2.2 [**Design Tools and Techniques**]: User interfaces

General Terms

Human Factors, Verification

Keywords

Accessibility, evaluation, tools, WCAG, testing, benchmark

1. INTRODUCTION

A number of studies indicate that the World Wide Web is not accessible [16]. One could blame on webmaster amateurism or their lack of familiarity with accessibility practices. However, a study of webmasters' perceptions revealed that the majority of them are familiar with accessibility guidelines [15]. The fact that webmasters put compliance logos on non-compliant websites [12] may suggest that some step is skipped in the development process of accessible websites. We hypothesise that the existence of a large amount of pages with low accessibility levels, some of them pretending to be accessible, may indicate an over-reliance on automated tests. This may imply that even if it is widely known that automated evaluation does not substitute human evaluation, perhaps this knowledge is not *that* widely spread. A lack of awareness on the negative effects of tool over-reliance may also influence the current situation. In order to alleviate this lack of awareness, this paper examines the role and reliability of automated tests and to what level tool output can be considered an accurate representation of a websites' actual level of accessibility. To do so, the role of automated evaluation versus expert manual evaluation is examined. Further, we look at a number of currently available automated evaluation tools and compare their output to that of a team of expert evaluators.

Web Content Accessibility Guidelines 2.0 (WCAG 2.0) form the basis of most international legislation[1] and organisations are increasingly being expected to state their accessibility conformance levels. Yet, the limitations of accessibility guidelines regarding their validity and testability are well known [6, 7]. It also is acknowledged that context has to be taken into consideration to ensure that guidelines meet accessibility expectations [22]. Additionally, relying on guidelines alone leaves out a number of issues as it was found that guidelines do not cover the whole spectrum of accessibility problems, as they only cover around 50% of accessibility problems encountered by users [19].

With the advent of WCAG 2.0 the use of automated evaluation tools has become even more prevalent. This is mainly due to the fact that WCAG 2.0 was designed to be more testable than its predecessor WCAG 1.0. In the absence of expert evaluators, organisations increasingly rely on automated tools as the primary indicator of their stated

*Informatika Fakultatea, University of the Basque Country UPV/EHU, 20018 Donostia, Spain.

[1] Policies Relating to Web Accessibility. Available at `http://www.w3.org/WAI/Policy/`

W4A2013 - Technical, May 13-15, 2013, Rio de Janeiro, Brazil.
Co-Located with the 22nd International World Wide Web Conference.
Copyright 2013 ACM 978-1-4503-1844-0 ...$15.00.

level. When it comes to the use of tools to evaluate web pages against accessibility guidelines it is highlighted that conformance review "*...is not likely to distinguish reliably between important and non-important accessibility problems*" [7]. Due to the highly interpretive nature of WCAG 2.0, automated evaluation tools can lack the discerning nature of a human evaluator, who can look subjectively at WCAG 2.0 and decide if a particular guideline has been satisfied. Typically, automated software tools mostly indicate a negative or positive result against a guideline with no contextualized interpretation of the guideline and its severity impact on the user. That is why many success criteria (henceforth SC) require human evaluation and therefore fall outside the scope of fully automated evaluation.

Most accessibility assessment takes a summative form whereby the evaluation is conducted at the end of the development process or on a site already in operation. The aim of a summative assessment is to determine the current level of conformance with guidelines for an existing web page or site. On the other hand, formative assessment can be conducted on an ongoing basis during the development. Optimally, the formative assessment process should be ongoing from the initiation of a project, and continue throughout the operational lifespan of the web page or site [6, 21]. Both types of assessment can be accomplished by expert human evaluators, automated evaluation tools or, ideally, a combination of both, and can be integrated into the software development and quality assurance lifecycle. At each relevant iteration of the lifecycle, automated tools are understood to be used in the initial stages of the accessibility assessment process (formative or summative) where most salient problems are caught. Then, the resulting prototype should be pipelined into the following stages of the assessment process where experts and end-users focus on more subtle issues.

There are a number of scenarios where accessibility evaluation is not only used to build accessible sites. To name a few, accessibility evaluation is also employed in quality assurance of web applications, accessibility observatories, information retrieval and user-adapted interaction scenarios [25]. These scenarios do not focus on repairing accessibility flaws, but they provide a diagnosis (normally in terms of conformance) about the accessibility of web pages. In these situations, tools are employed to bypass the burden of manually evaluating the accessibility of a unmanageable number of pages, each containing a few thousand lines of code. Moreover, tools are indispensable when the evaluation of a few pages must be carried out in real-time.

A comprehensive assessment of pages requires the involvement of human testers and end-users. In light of the unmanageable amount of pages or the time constraints that the above scenarios deal with, sampling techniques help to select a representative number of pages from a site. The importance of sampling in large web presence is emphasised by a number of different sampling techniques for accessibility testing [8, 14]. To the authors' knowledge there are no tools implementing different sampling approaches – not even the retrieval of a random subset of pages –, entailing that tool developers seem to omit this crucial aspect. Other approaches to error identification and correction come in the form of crowdsourcing [20]. In this approach, users can identify a page that contains accessibility errors and report it to an accessibility service. This service provides a community driven resource whereby members can read the error, make the page correction and submit the changes. The user is then notified of the change and directed to the corrected page. From the perspective of the end user, the crowdsourcing approach uses a self-sampling approach in that the user identifies the problems and requests appropriate, contextualized corrections.

As mentioned above, the purpose of web accessibility evaluation goes beyond building web applications. This is especially relevant when medium-large scale evaluations have to be performed or when evaluation reports are needed in real time. As a result, there is the risk of relying on automated tools alone, which is inevitable for large-scale evaluation. However, this risk is also extended as far as website development is concerned. This is quite problematic as the works described in section 2 confirm that, compared to other evaluation methods, tools alone perform poorly in terms of coverage and completeness. Therefore organisations should not rely on automated tests alone. Even if such a statement is obvious for some, data supporting such an assertion is outdated, scarce and reported in a unstructured way. What is more, we do not know to what extent, in quantitative terms, automated evaluation can be harmful.

The goal of this paper is to measure such harm in terms of (lack of) coverage, completeness and correctness. If we are able to quantify the limitations of relying on automated tests alone, we can measure how much is missed and we are able to infer the effort required to completely evaluate accessibility. Based on the current state of the Web, we suspect that there is an over-reliance on automated tests, leaving out further evaluation. The measurement of the problems caused by relying on tools alone will help to raise awareness about the problems generated due to poor evaluation practices. As a result, the outcomes of this paper inform organisations, practitioners, and web developers. We also target researchers who have to employ evaluation tools in their projects by showing them the error rates they are introducing if they rely on automated tests alone.

Inspired by previous work [5], this paper aims at providing empirical evidence on how well evaluation tools, and specifically automated tests, behave with respect to WCAG 2.0 conformance. Following sections describe how the performance of 6 state-of-the-art evaluation tools compares to accessibility evaluation carried out by experts.

2. RELATED WORK

There are several works that analyse how automated guideline review performs with respect to some other techniques. In one of the earliest studies to explore multiple methods of evaluation and to discuss levels of expertise of evaluators, multiple approaches to website evaluation were analysed including automated tools, expert evaluations and user testing [17]. It also sought to compare automated tools that were available at the time and identified the key issues of evaluating against WCAG 1.0. Findings indicate that testing with screen readers is the most thorough, whilst automated evaluation is the least. A perceived weakness in this study is that user testing was solely based on the baseline experiences of legally blind users with screen readers, therefore excluding a larger population of users with varying types of disability.

The role expertise plays in conducting effective, reliable and valid manual website evaluations has also been investigated [9]. The study, at its core, requires knowing what the real accessibility problems are in a given site: whilst it is assumed to a certain degree that more experts should mean more validity, more experts can also mean more disagreement. One of the major outcomes of the study is the establishment of the amount of testers that can produce reliable and valid results: 3 in the case of experts and 14 in the case of non-experts.

In this work we focus on the performance of automated guideline review inspection tools. The most relevant related work upon which this paper is based on analysed the effectiveness of automated tools [5], where effectiveness encompasses completeness, correctness and specificity qualities. Tool completeness is achieved by reducing the number of false negatives; in other words, by minimising the number of missed accessibility violations. Tool correctness is obtained by reducing the mistakenly reported issues, that is, false positives. Specificity is a quality that describes the number of tests implemented by each success criteria or checkpoint. Specificity measures the granularity level in which tests are implemented: the more level of detail, the more specificity. These metrics were computed on the automated test results of LIFT and Bobby, and then compared with the results for automated evaluation combined with manual evaluation.

The W3C Evaluation and Report Language (EARL) nomenclature for defining success criteria automation levels specifies three main types of tests that automated tools implement [1]: fully automated tests (`earl:automatic`), automated tests that require human intervention (`earl:semiAuto`) and those that solely rely on human evaluators (`earl:manual`). Previous works established that 44% of the rules to build ergonomic interfaces could be checked in an automated way [11]. When it comes to web accessibility evaluation, it was reported that a given tool could fully automate 15% of WCAG 1.0 checkpoints; 35% of checkpoints required some sort of human intervention and the remaining 50% required manual checks [10]. Therefore the range of semi-automation (automated + semi-automated tests) is around 44-50%, while full automation yields lower values.

3. TOOLS BENCHMARKING

In this benchmarking study we focus on the following specific metrics: *coverage*, *completeness* and *correctness*. Coverage measures the number of different success criteria that report at least one failure (true positive); completeness alludes to the ratio of reported violations over the actual number of violations; and finally correctness refers to how well tools minimise the number of mistakenly reported accessibility violations. In order to compute these metrics the next variables have to be taken into consideration:

- True positives (*tp*) are actual problems found by the tool.
- False positives (*fp*) are mistakenly flagged accessibility issues. When including them into reports, *fp* generate noise as these are taken as actual errors if no human judgement catches them.
- False negatives (*fn*) are those issues that the tool did not catch and are therefore missed.

Assessment of the capability and appropriateness of accessibility evaluation tools can be carried out in at least two ways: using test suites or selecting a representative sample of websites. The former consist often of web pages that may contain accessibility issues violating specific SC (see examples of such test suites elsewhere [2, 23]). Typically, these suites have several tests to assess tool behaviour with respect to a given SC: when accessibility issues have deliberately been injected tools should be able to catch these issues (producing *tp*); if not, that would be a *fn*. In those tests conforming to SC, tools should not trigger any notification (yielding a true negative, *tn*); if the tool catches any issue that would be considered a *fp*. Test suites are useful resources to test tool validity and reliability; their shortcomings are mainly caused because tests are often isolated pieces of code without any surrounding context. While this feature is often a strength (e.g., appropriate to test coverage), it is a weak point at the same time:

- Tests are defined in a high specificity level; however, combinations and permutations of these tests are normally found in the Web. That is why test suites are not suitable to test tool performance in ecological settings.
- For the same reason, tool performance cannot be characterised with respect to the type of website analysed in terms of (amongst many others): accessibility (high, low), content (dynamic, static), topic (news, social networks, e-commerce), etc.

Real sites provide a richer view of tool evaluation as SC can be violated in many unexpected and combinatorial ways in which test suites cannot foresee. Ideally, the selection should have a broad coverage of SC and selected pages should be representative enough.

3.1 Materials, Sampling and Apparatus

We selected three websites in the context of a broader project that analyses the impact of the current accessibility strategy of the Australian Government. The Web Accessibility National Transition Strategy [4] provides all Australian federal agencies with a plan for the adoption and implementation of WCAG 2.0.

The site of the Prime Minister of Australia (`http://www.pm.gov.au/`) was picked because it is a Federal Government website; therefore it is part of the mandated strategy. Consequently it should represent optimal accessibility levels due its high profile nature. Vision Australia site (`http://www.visionaustralia.org.au/`) was chosen as it is a non-government non-profit organisation which is regarded internationally as a repository for accessibility information. It is used by many people with disabilities and also developers for information for people with visual disabilities. Therefore, we expect the site to be accessible although it does not have to abide by the strategy. Transperth (`http://www.transperth.wa.gov.au/`), the site that gathers all the information about public transport services in the Perth area was chosen as it is state-government-affiliated and used constantly by people with disabilities because many of them are unable to drive a car and as such rely on public transport for mobility. In terms of accessibility, it is one of the worse sites of a pool of sites evaluated in previous works.

3 web pages were selected within each website following the *ad-hoc* sampling approach, as recommended elsewhere [26]. The criteria for the sampling was to incorporate the broader coverage of WCAG 2.0 SC. Consequently, we selected those pages containing the broader and more heterogeneous amount of functionalities. In this way we should be able to catch as much SC violations as possible. As a result, we were able to select not only information-centric pages, but also pages with high levels of interactivity and pages containing rich multimedia content such as video, audio streaming and live updates via AJAX.

The site of the Prime Minister of Australia has a clean appearance and sensible information architecture. The pages we selected contained videos describing the activity of the Prime Minister, the procedures to get in touch with her via social networks and forms to contact her. The homepage was an amalgamation of multimedia, social networks, widgets and static content. Vision Australia site has a neat and clear design and among the pages we selected, not only static content was tested but also a streaming radio service, forms where individuals could make a donation to the organisation and sitemaps with search functionalities. Transperth is very cluttered, has an information overload problem and is not trivial to operate. We selected those pages that users would more frequently use: the journey planner on the home page consisted of a form to set the departure and destination, time and means of transport to plan a journey; the home page also allowed to filter Twitter and RSS feeds, it had a login functionality to get a personalised update service and there was also a widget to monitor the current status of train lines. Remaining pages contained timetables and maps and also the procedure to get in touch with Transperth staff. We can say that the site was highly interactive and exhibited lots of functionalities which were not easy to interact with.

The aforementioned pages were retrieved using the HT-Track Website Copier; subsequently these pages were stored on a web server located at Edith Cowan University in Perth (Western Australia) in order to guarantee consistency across evaluations.

3.2 Expert Evaluations

Each of the authors of this paper evaluated independently the conformance to WCAG 2.0 of each page. In order to reach an agreement on the accessibility conformance of each of the 9 pages, judges collaborated, found issues and discussed them individually. Being an odd number of judges, decisions on particular issues were taken by majority. If no agreement was reached among the three judges a legally blind expert user was consulted. The expert was consulted not only about those SC that affect blind users, but also about the remaining ones as he is a web accessibility expert. 3 experts guarantee that almost all accessibility violations can be found with a high reliability [9]. This protocol goes further and establishes a debate between judges and last resort consultation with end-users. As a result, higher level of validity and reliability would be expected.

The techniques used to assess the accessibility of each web page are diverse across judges: evaluation tools that diverge from the ones benchmarked (WAVE[2]), mark-up validators, browser extensions for developers (Firebug, Web Developer Toolbar, Web Accessibility Toolbar), screen readers (VoiceOver, NVDA) and evaluation tools based on simulation such as aDesigner [24]. Dynamic content was tested conducting usability walkthroughs on the problematic functionalities: for instance, checking the status of a determined train line, submitting information on forms, registering to a website, etc. Accessibility conformance evaluation was carried out across platforms (Windows and MacOS) and browsers (Firefox, Safari and Internet Explorer). Evaluating each web page took an approximate time of 30 minutes on pages with lower number of violations and 60 minutes on pages with a higher number of them.

Table 1: Analysed accessibility evaluation tools.

Tool	License	Deployment	Reporting
AChecker	free	online	Web
SortSite	commercial	desktop and online	Various (email, XML, MS-Word, etc.)
TotalValidator	free and commercial	online	Web
TAW	free	online	Web
Deque	free and commercial	online	Web
AMP	commercial	desktop	Various (MS-Excel, etc.)

3.3 Automated Evaluation Tools

6 evaluation tools were employed to evaluate the aforementioned 9 web pages: AChecker, SortSite, Total Validator, TAW, Deque, AMP. Selected tools have in common their capability to test web pages against the WCAG 2.0 guideline set. Typically, those tools that require a commercial license (SortSite, AMP) provide a desktop evaluation environment, while free tools (remaining ones) are mainly online services. Table 1 provides some insight on the features of each tool. Some tools are exclusively focused on web accessibility evaluation, whereas others provide additional tests for usability evaluation and quality assurance. Tools tend to report in diverse and heterogeneous ways including web reports, XML files and email. Therefore, for consistency reasons, data was extracted and put into a common uniform report.

3.4 Protocol

W3C-WAI establishes three conformance levels for their guidelines: A, AA and AAA. While all levels are equally important, A and AA level success criteria are fundamental to provide access. In our study A and AA level success criteria were included and most of the AAA level success criteria were set aside due to: barrier severity reasons (we wanted to focus on the most crucial ones); because most tools do not cover them (except for the SC we considered); and because the W3C-WAI explicitly warns about enacting policies for AAA conformance, as it might be impossible to meet in some cases[3]). Only 2 AAA level success criteria were included in our study: "2.4.9 Link Purpose" and "2.4.10 Section Headings".

Expert evaluations reported the previously mentioned SC (A, AA and the two belonging to AAA) and included them by writing down the tag, attribute or piece of code that had caused each accessibility violation; the line where the problem had occurred was also collected. These reports were matched against those reports produced (and subsequently transformed by us) by automated evaluation tools. In the case of tools, only those issues considered fully automated by tools were taken into account, while semi-automatic tests were set aside. In order to count the number of *tp*, the HTML tag, attribute and line where the violation occurred (± 5 lines) had to match; otherwise a *fn* was counted. We were lenient with violation line numbers because tools tend to be inconsistent in this regard, especially when dealing with large HTML documents. Also, all errors produced by tools

[2] WAVE was used due to its visual reporting capabilities. We ruled out WAVE for benchmarking purposes because it does not produce a machine readable report and does not provide explicit support for WCAG 2.0 conformance. The no inclusion of WAVE into the tested set removed the potential bias it could have been introduced in establishing the 'ground truth'.

[3] `http://www.w3.org/TR/UNDERSTANDING-WCAG20/conformance.html#uc-conformance-requirements-head`

Table 2: Number of violations per site (PM: Prime Minister; VA: Vision Australia; TP: Transperth) across conformance levels (A, AA, AAA). The number of SC with at least one violation and the median of the violation frequency.

	PM			VA			TP		
	A	AA	AAA	A	AA	AAA	A	AA	AAA
Violations	82	13	5	71	35	11	262	147	24
Total		100			117			433	
Unique SC		17			11			20	
Median		3			0			3	

were reviewed to catch *fp*.

4. RESULTS

In establishing the 'ground truth' the authors of the paper carried out conformance evaluations following the method described in section 3.2. As a result, a total 650 accessibility violations were found. Even if experts perform better than tools, it should be noted that experts may also produce mistakes so the actual number of violations should be considered an estimation; henceforth, we will refer to them as the *actual* number of violations. The most frequently violated SC were "1.3.1 Info and Relationships", "1.4.3 Contrast", "1.1.1 Non-text Content", "1.4.4 Resize Text" and "2.4.4 Link Purpose" with 135, 95, 95, 83 and 40 issues respectively. When grouping SC violations according to the accessibility principle they belong to, 465 issues were found for Perceivable (72%), 139 for Operable (21%), 35 for Understandable (5%) and 22 for Robust (2%). There were also a number of SC for which violations were not found, for instance: "1.3.3 Sensory Characteristics", "2.4.5 Multiple Ways" and "3.2.1 On Focus". Out of 40 SC that were considered initially, 26 of them (65%) had at least one violation. This means that the scope of our study is limited to those SC that were actually found.

Table 2 shows the number of violations per site and across conformance levels. We can find that, Transperth website is by far the site with more violations (433) and therefore the one with lower accessibility. The Prime Minister site yields the lowest number of violations (100), closely followed by Vision Australia (117). In order to ascertain which of the two sites is more inaccessible, counting the number of violations is risky when the numbers are that close. When it comes to satisfying the most fundamental SC (all the SC belonging to conformance level A) the Prime Minister site and Vision Australia violate 82 and 71 SC respectively. Computing the average number of violations per SC would not be appropriate as the frequency of the number of violations in each success criteria show positevely skewed distributions (long tails on the right). That is, most SC are violated a few times, while a few SC are violated a high number of times. Therefore we calculate the median instead of the arithmetic mean in order to provide a better approach for central tendency [27]. Otherwise, high values would bias average number of violations per SC. As a result, the medians for Vision Australia and Prime Minister sites are 0 and 3 respectively. In light of these data we can say that the Vision Australia website, even if it is similar to the Prime Minister site at first sight, is the most accessible of the three sites. Transperth outnumbers remaining sites so it would stand out for its non-accessibility. Therefore, based on their accessibility level, the sites are categorised as Vision Australia ≫ Prime Minister ⋙ Transperth.

Table 3: Number of SC that were caught at least once (coverage scores) across principles and per tool. Percentage terms denote the relative amount of different success criteria caught with respect to all success criteria in a principle. P stands for Perceivable, O for Operable, U for Understandable and R for Robust.

Tool	P	O	U	R	overall
AChecker	3 (38%)	3 (25%)	1 (25%)	1 (50%)	8 (31%)
SortSite	3 (38%)	5 (42%)	1 (25%)	1 (50%)	10 (38%)
TV	3 (38%)	3 (25%)	1 (25%)	1 (0.5)	9 (35%)
TAW	3 (38%)	5 (42%)	2 (50%)	2 (100%)	13 (50%)
Deque	3 (38%)	3 (25%)	4 (100%)	1 (50%)	11 (42%)
AMP	2 (25%)	3 (25%)	1 (25%)	0 (0%)	6 (23%)

A total number of 117 violations across 11 SC were found in the most accessible site mainly due to inconsistent (SC 1.3.1) and non-conformant nesting of headers (SC 2.4.10). The specification of font sizes in absolute terms (SC 1.4.4) and missed identification of required fields in forms (SC 3.3.1) are some other accessibility problems found. In the medium accessibility site, we get fewer violations than in the most accessible site (100) although they are more distributed across 17 SC: accessibility problems are mainly caused by `fieldset` tags without `legend` attribute and inconsistent nesting and lack of headings (SC 1.3.1); we found a number of headings that did not use the appropriate tag although the intention was explicitly put as a class name `<p class="h1">`. There are also significant amount of problems generated by the usage of `&nbsp` for layout purposes (SC 1.3.2), lack of alternative text for video screenshots (SC 1.1.1) and colour contrast problems (SC 1.4.3). Finally, not all functionalities (e.g., videos and forms) were reachable through keyboard (SC 2.1.1) and the purpose of many links was unclear as many of them contained the same text while pointing to different web resources (SC 2.4.4). The least accessible site produced 433 violations across 20 SC amongst many others, there was a lack of alternative text for images and buttons and lack of empty `alt` attribute for decorative pictures (SC 1.1.1); inconsistent headings, unlabelled form controls, tables without `summary`, `th` or `scope` and forms without `fieldset` (SC 1.3.1); colour contrast issues (SC 1.4.3), ambiguous link purpose with *click here* and question mark as a text (SC 2.4.4) in addition to small and fixed font sizes (SC 1.4.4).

4.1 Coverage

Out of 26 SC violated, we found that only 23-50% of SC are covered by automated tests in tools. A success criterion is considered to be covered if at least one true violation is reported by a given tool. TAW reports at least one true violation in 50% of the criteria, while the coverage of the remaining tools is smaller: 42% for Deque, 38% for SortSite, 35% for Total Validator, 31% for AChecker and 23% for AMP.

Coverage of SC across accessibility principles is quite variable according to table 3. Considering that at least one *tp* is found for 8, 12, 4 and 8 SC belonging to Perceivable, Operable, Understandable and Robust principles respectively, those principles with a smaller amount of covered SC – Understandable and Robust – reach full coverage with some tools. However, the coverage across tools varies from 25 to 100% for Understandable and from 0 to 100% for Robust. More modest – and less variable – coverage values can be found for those principles with a broader coverage of violations. Coverage for Perceivable SC ranges between 25-50%, whereas for Operable it does between 25-42%.

Table 4: Number of *tp* (2nd column), percentage of *tp* over the actual number of violations (completeness scores) across principles (col 3-6) and *fn* (7th column) per tool.

Tool	tp	P	O	U	R	fn
AChecker	91 (14%)	14%	12%	11%	73%	559 (86%)
SortSite	192 (30%)	26%	47%	3%	73%	458 (70%)
TV	206 (32%)	35%	23%	3%	73%	444 (68%)
TAW	249 (38%)	38%	45%	6%	90%	401 (62%)
Deque	181 (28%)	30%	23%	20%	27%	469 (72%)
AMP	142 (22%)	23%	24%	3%	0%	508 (78%)

4.2 Completeness

Completeness measures the number of true violations found by tools with respect to actual number of violations reported by experts. In other words, completeness conveys how well a tool minimises *fn* while maximising *tp*. The second column in table 4 shows the absolute number of *tp* and its relative score in percentage terms between parentheses – calculated as the ratio between the overall number of *tp* found by a given tool and the actual number of violations, 650. Completeness values range from a minimum of 14% obtained by AChecker to a maximum of 38% in the case of TAW. False negatives in the last column of table 4 measure the number of missed true violations by tools, which is the number of violations remaining up to 650.

The number of *tp* across accessibility principles and tools is depicted in Figure 1. It can be observed that for most tools Robust is the principle that scores higher when it comes to completeness except for AMP, which shows no completeness at all. Additionally TAW is the tool that better performs across all principles but for Understandable where Deque with a modest 20% of completeness outperforms remaining tools.

Then, we explored the effect of SC conformance levels (A and AA) on tool completeness. A Wilcoxon Signed-rank test confirms that there is a significant effect of conformance levels on completeness, $W = 21, Z = 2.20, p < 0.05, r = 0.63$, which entails that the ratio of *tp* over the actual number of violations is significantly higher for those SC belonging to conformance level A.

Websites with different accessibility levels may pose different challenges to evaluation tools due to the number and nature of violations they contain. To shed some light on this possible variability we analyse the behaviour of tools (in terms of completeness) across the three websites with their corresponding accessibility levels. Figures 2(a), 2(b) and 2(c) illustrate tool completeness across tool, principle and site. At first sight, high completeness values can be observed across the three websites for Robust and much lower scores for the remaining principles. There is a pattern showing higher completeness scores in Operable and lower completeness in Understandable for Vision Australia and Prime Minister sites. However, there are some tools that manage to stand out in Operable and Understandable: Deque for Operable SC in the most accessible site and AChecker for Understandable SC in the medium accessibility site. By mere observation of histograms it can be said that the medium accessibility site follows the pattern of the most accessible site although in a more attenuated way, confirming their closeness in terms of violations per SC as shown in table 2. On the other hand – if we discard Robust principle – Figure 2(c) shows high completeness scores in the low accessibility site for Perceivable except in the case of SortSite and TAW, which exhibit high completeness in Operable. In this case, only Deque is able to stand out among the rest of the tools for Understandable. Overall completeness across sites change according to the values shown in table 5.

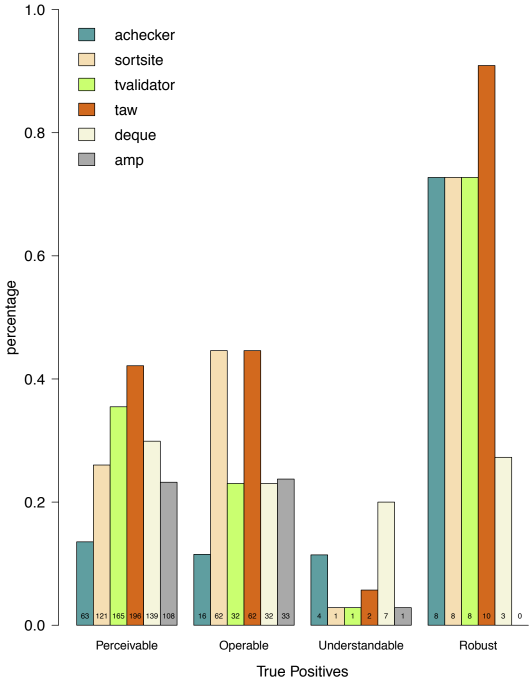

Figure 1: Completeness per tool and principle.

Table 5: Percentage of *tp* over the actual number of violations (completeness scores) across type of sites (high (VA), medium (PM) and low accessibility (TP) and tool.

Tool	high	medium	low
AChecker	8%	17%	15%
SortSite	6%	11%	39%
TotalValidator	9%	10%	42%
TAW	19%	13%	48%
Deque	9%	0%	39%
AMP	10%	8%	28%

In order to ascertain the effect of the accessibility level on tool completeness we performed a one-way repeated-measure ANOVA: Mauchly's test yields $W(2) = 0.31, p = 0.09$ meaning that sphericity against website type is not violated; a significant effect of website's accessibility level on overall tool completeness $F(2, 10) = 19.82, p < 0.001, \eta^2 = 0.72$ with 95% CI [0.3, 0.83] was found. If only highly accessible and low accessibility sites are compared we also find a significant effect $F(1, 5) = 33.67, p < 0.01, \eta^2 = 0.69$ with 95% CI [0.21, 0.89]. As effect size is independent of the sample, these large values suggest that the effect exists and it is considerable. This entails that the accessibility level of the site being tested affects tool completeness in that the more inaccessible a site is, the higher rate of true violations that may be found by tools.

In order to compute tool similarity with respect to the number of *tp* caught, Cronbach's $\alpha = 0.96$, 95% CI [0.68, 0.98] yields a very high value, entailing that tools are quite similar in this regard. A two-dimensional space, allows not only to depict similarity among tools, but also between tools and an hypothetical *optimal* tool, which is based on our 'ground truth'. In order to visualise such relationship, we employed classical multidimensional scaling and computed

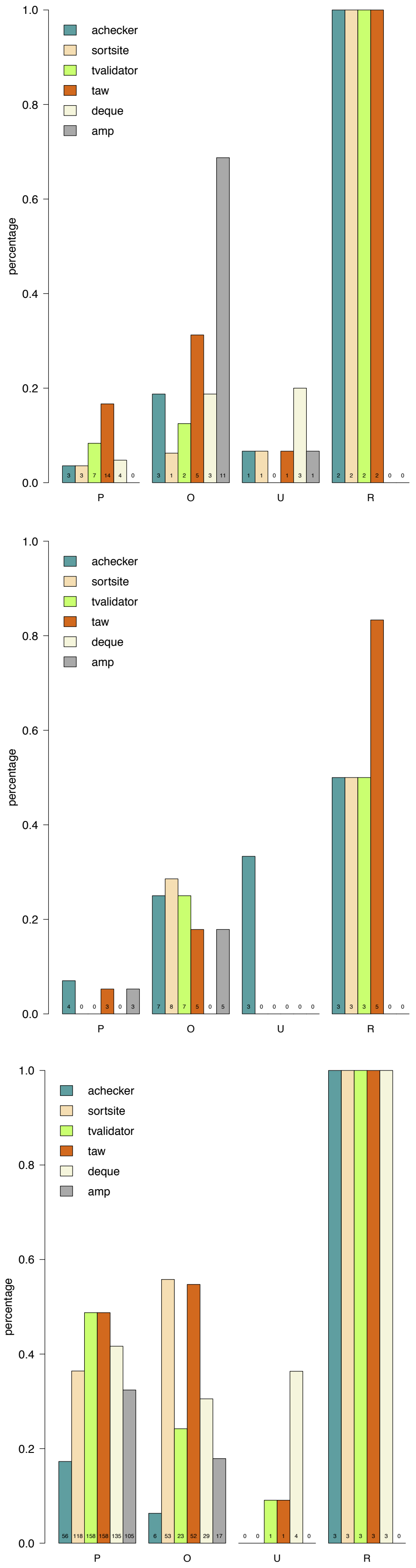

Figure 2: Completeness scores across sites, see figure (a) at the top for Vision Australia, (b) in the middle for Prime Minister and (c) at the bottom for Transperth.

the Euclidean distance between the *tp* of the SC across tools (including those violations found by experts). Figure 3 shows how tools locate in the two-dimensional space; the closer tools are the more similar they are – entailing to some extent tool interchangeability. Results show that indeed, tools are very similar as they are aligned along the same y-axis area. However, they are far from performing as the optimal tool: if we compute distances, AChecker and AMP (187 and 177 distance units) are the tools which are most dissimilar to the optimal tool, whereas TAW (138 distance units) is the one which is more similar – while still different.

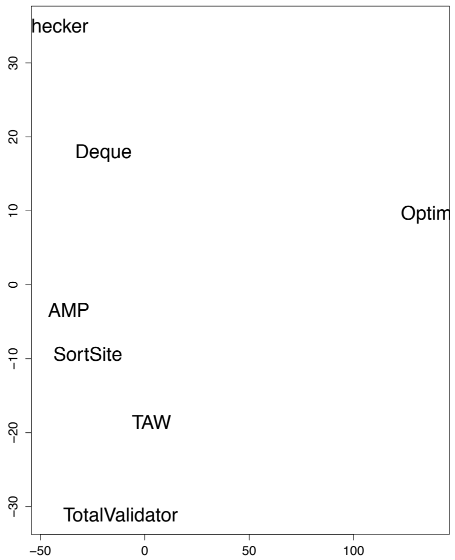

Figure 3: Distance between tools based on the *tp* caught.

If we observe tool similarity on the most accessible, medium accessibility and least accessibility website we obtain $\alpha = 0.67$, 95% CI [0.32, 0.87], $\alpha = 0.83$, 95% CI [0.28, 0.89], $\alpha = 0.96$, 95% CI [0.62, 0.98] respectively; this shows that, as far as completeness is concerned, the less accessible a site is the more more similarly tools would behave.

4.3 Correctness

Correctness measures how well a tool is able to minimise the number of *fp*; in other words, how tools are able to reduce the number of mistakenly reported errors. False positives generate noise on reports and can turn out to be quite harmful if human testers rely just on tools, especially on their automated tests. Evaluation tool users assume that reported accessibility violations are correct and they are seldom manually verified.

Table 6: Number of *fp* (2nd column) and ratio of *fp* over the actual number of violations (3rd column) across tools.

Tool	fp	incorrectness
AChecker	7	7%
SortSite	11	5%
TotalValidator	104	34%
TAW	78	29%
Deque	8	4%
AMP	1	7%

Table 6 shows the number of *fp* (2nd column) and incorrectness scores (3rd column) calculated as the ratio of *fp* over the total of issues reported (*tp*+*fp*). In order to compute correctness, incorrectness scores should be subtracted from 100. Most tools exhibit high levels of correctness that range between 93-96% except for TAW and TotalValidator, which yield 71% and 66% respectively. This means that 1 out of 3 automated errors reported by the mentioned tools are flawed.

5. DISCUSSION

The vast majority of accessibility violations in the websites we analysed belong to Perceivable (with a 72%) and Operable principles, accounting for a 93%. Compared to this figure, the number of Understandable and Robust violations is somewhat marginal, which accounts for 7% of them. Considering we analysed 14 SC for Perceivable and Operable, 10 for Understandable and 2 for Robust we can say that the number of criteria does not correspond to this unbalance.

Regarding the generalisability of the results, we are aware that the sample of websites is hardly representative of the entire Web. Yet, this study is still representative of the general accessibility problems a user might come across in the Web as we found at least one accessibility violation in all WCAG 2.0 principles and 75% of guidelines. However, we also acknowledge there are limitations in this study (see section 5.1).

On tool coverage.

Tool coverage, measured in terms of the number of SC that have been reported at least once, is quite variable: coverage is very narrow (all tools cover less than 50% of SC). This figure is consistent with previous research on the automation limits of usability evaluation [3, 10, 11], which establish that around 44-55% of tests can be automated.

Among the tools we analysed, only TAW can manage to report on half of them. In the worst-case scenario (employing just Deque) only 1 out of 4 SC are covered. This means that a vast amount of the SC are missed by automated tests. When analyzing coverage per principle we notice that Perceivable and Operable SC are the ones having more room to improve. Obviously there are limitations on automating certain SC; for instance, ascertaining whether there are keyboard traps (“2.1.2 No Keyboard Trap”) requires real interaction or simulation. However, there are some SC that just require checking the presence of certain HTML tags and content, which is not even covered by some tools – for example, checking whether the web page has a title (“2.4.2 Page Titled”).

On tool completeness.

Completeness, measured as the ratio between violations reported by tools over the actual number of violations, gets very low values. The tool that in overall shows the best performance, TAW, can only catch around 38% of violations. This score is drastically reduced by remaining tools until a 14%, which is the lower value exhibited for completeness. This means that in the best case scenario automated tests are able to adequately catch 4 out 10 accessibility violations.

Across principles, completeness scores are quite acceptable for Robust (> 73%) even if there are some exceptions (Deque and AMP). This can happen because of the small number of SC involved. Operable and Understandable SC show a similar behaviour across tools (on the range of 14-47% of completeness) although SortSite and TAW perform better for Operable reaching around 45-47%. However, scores are still very low and are even lower for the Understandable principle where Deque exhibits the best performance with a 20%. The lack of automation in Understandable is evident as some SC are quite challenging; however tools do not implement techniques that are commonly applied in other domains (e.g. Natural Language Processing) such as detecting changes on the language [18] for SC such as “3.1.2 Language of Parts”; similarly, the automatic detection of the genre of texts (legal, fiction, science and technology, etc.) is feasible [13], which is useful to raise warnings in those sites that require legal commitments (see SC “3.3.4 Error Prevention”). This lack of automation leads us to conclude that evaluation tool developers are disengaged with common computing practice.

On the effect of the accessibility level on completeness and tool similarity.

We have found that the accessibility level of the website being tested may have an effect on the completeness of tools. The more inaccessible the website is, the higher completeness values are obtained. This might happen because accessibility violations on highly accessible sites are more subtle and thus more difficult and challenging to catch. On the other hand, low accessibility websites contain a number of usual and expected accessibility problems. That is, tools are better designed to catch stereotypical and more frequent accessibility issues, especially if they belong to Perceivable principle (see Figure 2(c)), while the less frequent ones are hardly targeted.

Analogously, tools behave more similarly when the tested site is non-accessible. Again, this supports that tools are designed to catch archetypical and expected accessibility issues, while little effort is devoted to implement less frequent although still present SC violations. We have found that even if tools exhibit different strengths and weaknesses their performance regarding completeness is equally low.

On the effect of conformance levels in completeness.

We found that tools show significantly higher completeness in those SC that are considered more crucial (A) to ensure accessibility than in those of a relatively lower importance (AA). This may entail that tool developers prioritise A level SC and thus they focus more on implementing them. Alternatively we could say that tests to catch AA level SC are more challenging to implement.

On tool correctness.

In general tools show a high level of correctness, which is higher than 93%. The low number of *fp* contrasts with those found in similar studies [5] where not only automated tests but also warnings were considered. Thus, we can say that tool developers do not take risks when it comes to automated tests and these are only reported under high levels of certainty. However, there are two exceptions, TotalValidator and TAW, which report 34% and 29% of incorrectness respectively. These tools are also the ones with higher completeness scores 32% and 38% each. Therefore, the tools with higher completeness are also the most incorrect ones. This suggests that, as opposed to remaining tools, these tools aim at catching as many accessibility violations as possible at the cost of making mistakes and increasing their incorrectness. Finally, we can say that SortSite is the tool with a more balanced approach – measured by maximising qualities –, 30% of completeness and 95% of correctness.

Which tool should I use?.

Even if, as demonstrated, tool coverage and completeness is minimal and can often lead to incorrect results, as far as automated tests are concerned, tools help to assure this 'minimal' accessibility level, which is better than nothing. However, as acknowledged by the community, this should not overshadow accessibility evaluation by experts. One possible solution to improve the low effectiveness would be to use multiple tools. This statement has some truth to it as long as we are able to identify weaknesses and strengths of tools. As a result tools could be employed on those SC they show higher effectiveness levels in order to maximise coverage, completeness and correctness. We have shown that tool behaviour can change between high and low accessibility sites; we also learned that some tools perform better when certain principles are tested. Therefore, the outcomes of this research can inform decision-making on tool selection. Being aware that there are more tools than the ones benchmarked in this study, we still could maximise completeness by employing at each SC the tool that reports the higher number of *tp*. Note that by following such a procedure completeness reaches 55% obtaining an increase of 17 percentage points (83 SC violations more caught) from the tool that shows the best completeness performance. A similar example but restricted to those tools that not require a commercial license yields 52% of completeness (68 SC violations more). Note that these increases are not generalisable but are limited to the sites we analysed. In any case they are indicative of how valuable it is to know the strengths of tools.

Tools exhibit different strengths and weaknesses. Some of them perform well across various types of website and accessibility principles, no matter the effectiveness quality we analyse. Others are less consistent but perform well in some particular situations where remaining tools are weak. SortSite, TotalValidator and TAW belong to the former group; they show the highest coverage and completeness scores across all principles and sites. However, except for SortSite, the number of incorrectly identified issues that TAW and TotalValidator report are also the highest. AChecker, Deque and AMP belong to the group where tools can exhibit their strength in specialized situations – additionally they get high correctness scores. When it comes to completeness, Deque outperforms remaining tools when evaluation SC under Understandable principle; AChecker shows good completeness scores for SC "3.1.1 Language of Page" and AMP does a good job catching violations of "2.4.10 Section Headings".

5.1 Limitations of the Study

The study has several limitations in that only 65% of SC were violated in the analysed websites, while guidelines coverage is of 75%. If we explore further the 14 SC that were not analysed, 5 belong to multimedia SC which are typically related to real-time captioning of video and real-time audio descriptions. Our study has partially covered these issues as a number of violations for video and audio content have been analysed under SC 1.1.1 and 1.2.2. Hence, even if this study is not exhaustive when it comes to real-time multimedia and accessibility issues, we can say that multimedia accessibility has not been set aside. There are another 3 SC addressing website consistency (SC 2.4.5, 3.2.3, 3.2.4) that did not produce failures due to the employed sampling method. Lastly, according to "3.3.4 Error Prevention", those transactions that require legal commitments such as online licences, contracts and similar should provide the means so that users do not make mistakes. The main cause why the aforementioned 9 SC were not violated is because the pages we analysed did not contain potential violation points. That is, we were unable to address the failure of those SC that could only be tested through the evaluation of sites in a more ecological fashion that would include real time streaming of multimedia content, website level evaluation and specific genre pages.

Regarding the remaining 5 SC that did not produce any failure (accounting for a 12% of the SC), these were "1.3.3 Sensory Characteristics", "2.4.6 Headings and Labels", "3.1.2 Language of Parts", "3.2.1 On Focus", "3.2.4 Consistent Identification", "3.3.3 Error Suggestion" we can only say that tools produced true negatives. If we had broaden our sample we would be able to increase the number of failures of the SC described in the previous paragraph. However, those SC related to higher levels of ecological validity would still be challenging to test.

6. CONCLUSION

The analysis of effectiveness of 6 state-of-the-art accessibility evaluation tools in terms of coverage, completeness and correctness corroborates that employing tools alone and leaving out human judgment is indeed not recommended. In this paper we quantify how harmful it can be to rely on automated tests alone: in the pages we sampled, if the right choice of tool was made half of the SC would be missed and 6 violations out of 10 would not be caught. Results would be even worse if the best tool for each effectiveness quality was not employed.

We have found that the accessibility level of the website may affect tool completeness. Tools show a more similar behaviour and higher completeness the more inaccessible a website is, whereas on more accessible pages that still contain accessibility flaws tools tend to diverge and produce lower completeness scores. This might happen because there are stereotypical (and perhaps more frequent) accessibility issues that tools aim at catching. Those issues that are subtler or less frequent are not well covered by tools and indeed those SC with A level exhibit higher levels of completeness. Another finding is that tools exhibiting an overall higher value for completeness show the lower values for correctness. This suggests that automating as many SC as possible takes its toll in terms of an increased number of *fp*. However, as the behaviour of SortSite demonstrates this phenomenon does not necessarily have to occur.

We finally demonstrate how effectiveness in terms of coverage and completeness can be boosted if the right combination of tools are employed for each SC. Using our sample of tools in our sampled sites, completeness was increased in 17 percentage points. Further work can potentially explore how to select the tools that maximise completeness without harming correctness. Also, we will investigate the number of tools needed to guarantee acceptable levels of effectiveness.

7. ACKNOWLEDGEMENTS

The datasets used in this paper as well as the websites tested are available at `http://www.markelvigo.info/ds/bench12/index.html`.

8. REFERENCES

[1] S. Abou-Zahra. Evaluation and report language (earl) 1.0 schema – w3c working draft, 2011.

[2] S. Abou-Zahra and M. Cooper. Wcag 2.0 test samples repository. In *Universal Access in Human-Computer Interaction. Applications and Services*, volume 5616 of *LNCS*, pages 619–627. 2009.

[3] A. Aizpurua, M. Arrue, M. Vigo, and J. Abascal. Transition of accessibility evaluation tools to new standards. In *Proceedings of the 2009 International Cross-Disciplinary Conference on Web Accessibililty (W4A)*, W4A '09, pages 36–44, 2009.

[4] Australian Government. Web accessibility national transition strategy available at `http://www.finance.gov.au/publications/wcag-2-implementation/docs/wcag-transition-strategy.pdf`. 2010.

[5] G. Brajnik. Comparing accessibility evaluation tools: a method for tool effectiveness. *Universal Access in the Information Society*, 3:252–263, 2004.

[6] G. Brajnik. Beyond conformance: The role of accessibility evaluation methods. In *Proceedings of the 2008 international workshops on Web Information Systems Engineering*, WISE '08, pages 63–80, 2008.

[7] G. Brajnik. A comparative test of web accessibility evaluation methods. In *Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility*, Assets '08, pages 113–120, 2008.

[8] G. Brajnik, A. Mulas, and C. Pitton. Effects of sampling methods on web accessibility evaluations. In *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, Assets '07, pages 59–66, 2007.

[9] G. Brajnik, Y. Yesilada, and S. Harper. The expertise effect on web accessibility evaluation methods. *Human-Computer Interaction*, 26(3):246–283, 2011.

[10] M. Cooper, Q. Limbourg, C. Mariage, and J. Vanderdonckt. Integrating universal design into a global approach for managing very large web sites. In *Proceedings of the 5th ERCIM Workshop on User Interfaces for All*, 1999.

[11] C. Farenc, V. Liberati, and M.-F. Barthet. Automatic ergonomic evaluation: What are the limits? In *Proceedings of the Second International Workshop on Computer-Aided Design of User Interfaces*, CADUI '96, pages 159–170, 1996.

[12] T. D. Gilbertson and C. H. C. Machin. Guidelines, icons and marketable skills: an accessibility evaluation of 100 web development company homepages. In *Proceedings of the International Cross-Disciplinary Conference on Web Accessibility*, W4A '12, pages 17:1–17:4, 2012.

[13] B. Kessler, G. Numberg, and H. Schütze. Automatic detection of text genre. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, ACL '98, pages 32–38. Association for Computational Linguistics, 1997.

[14] M. King, J. W. Thatcher, P. M. Bronstad, and R. Easton. Managing usability for people with disabilities in a large web presence. *IBM Systems Journal*, 44(3):519–535, 2005.

[15] J. Lazar, A. Dudley-Sponaugle, and K.-D. Greenidge. Improving web accessibility: a study of webmaster perceptions. *Computers in Human Behavior*, 20(2):269–288, 2004.

[16] R. Lopes, D. Gomes, and L. Carriço. Web not for all: a large scale study of web accessibility. In *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A)*, W4A '10, pages 10:1–10:4, 2010.

[17] J. Mankoff, H. Fait, and T. Tran. Is your web page accessible?: a comparative study of methods for assessing web page accessibility for the blind. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '05, pages 41–50, 2005.

[18] B. Martins and M. J. Silva. Language identification in web pages. In *Proceedings of the 2005 ACM symposium on Applied computing*, SAC '05, pages 764–768, 2005.

[19] C. Power, A. Freire, H. Petrie, and D. Swallow. Guidelines are only half of the story: accessibility problems encountered by blind users on the web. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, CHI '12, pages 433–442, 2012.

[20] D. Sato, H. Takagi, M. Kobayashi, S. Kawanaka, and C. Asakawa. Exploratory analysis of collaborative web accessibility improvement. *ACM Transactions on Accessible Computing*, 3(2):5:1–5:30, 2010.

[21] C. C. Shelly and M. Barta. Application of traditional software testing methodologies to web accessibility. In *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A)*, W4A '10, pages 11:1–11:4, 2010.

[22] D. Sloan, A. Heath, F. Hamilton, B. Kelly, H. Petrie, and L. Phipps. Contextual web accessibility - maximizing the benefit of accessibility guidelines. In *Proceedings of the 2006 international cross-disciplinary workshop on Web accessibility (W4A)*, W4A '06, pages 121–131, 2006.

[23] C. Strobbe, J. Koch, E. Vlachogiannis, R. Ruemer, C. Velasco, and J. Engelen. The bentoweb test case suites for the web content accessibility guidelines (wcag) 2.0. In *Computers Helping People with Special Needs*, volume 5105 of *LNCS*, pages 402–409. 2008.

[24] H. Takagi, C. Asakawa, K. Fukuda, and J. Maeda. Accessibility designer: visualizing usability for the blind. In *Proceedings of the 6th international ACM SIGACCESS conference on Computers and accessibility*, Assets '04, pages 177–184, 2004.

[25] M. Vigo and G. Brajnik. Automatic web accessibility metrics: Where we are and where we can go. *Interacting with Computers*, 23(2):137–155, 2011.

[26] W3C-WAI. Conformance evaluation of web sites for accessibility. 2005.

[27] R. Walpole. *Elementary statistical concepts*. Macmillan, 1976.