**Can a Web Accessibility Checker be enhanced by the use of AI?**

E.A. Draffan, Chaohai Ding, Mike Wald, Harry Everett, Jason Barrett, Abhirami Sasikant, Calin Geangu and Russell Newman

WAIS, ECS, University of Southampton, Southampton, UK

ead@ecs.soton.ac.uk

cd8e10@ecs.soton.ac.uk

mw@ecs.soton.ac.uk

**Abstract.**

There has been a proliferation of automatic web accessibility checkers over the years designed to make it easier to assess the barriers faced by those with disabilities when using online interfaces and content. The checkers are often based on tests that can be made on the underlying website code to see whether it complies with the W3C Web Content Accessibility Guidelines (WCAG). However, as the type of code needed for the development of sophisticated interactive web services and online applications becomes more complex, so the guidelines have had to be updated with the adoption of new success criteria or additional revisions to older criteria. In some instances, this has led to questions being raised about the reliability of the automatic accessibility checks and whether the use of Artificial Intelligence (AI) could be helpful. This paper explores the need to find new ways of addressing the requirements embodied in the WCAG success criteria, so that those reviewing websites can feel reassured that their advice (regarding some of the ways to reduce barriers to access) is helpful and overcomes issues around false positive or negatives. The methods used include image recognition and natural language processing working alongside a visual appraisal system, built into a web accessibility checker and reviewing process that takes a functional approach.

**Keywords:** digital accessibility, disability, automatic checkers, artificial intelligence

## **Introduction**

Over the past twelve years, a Web2Access[[1]](#footnote-1) system of 15 accessibility checks has been used as a functional review system for the accessibility of websites used for elearning by students, teachers and other academics. Anyone could access the reviews or add an evaluation that went some way to ensuring the online services listed, highlighted possible barriers for people with disabilities. The checks were originally based on the W3C Web Content Accessibility Guidelines (WCAG) version 2.0 [1] and took the user on a journey through a website. They started at the login stage catching the issues that might arise with reCAPTCHAs and unlabelled forms. The reviewer then went on to test for a lack of alternative text for images, style sheets that changed the navigation and then to the type of page that might require checks involving the use of keyboard only access, magnification and colour contrast levels. Other important interactive elements included in the review list were videos and audio accessibility, appropriate feedback from forms and access to tables, page integrity and text styles etc. to encompass the concept of readability.

Updates to WCAG 2.1 with the addition of seventeen new success criteria [2] meant that the Web2Access review system had become outdated and it was time to follow others looking into the potential of AI as a method to support checks [3]. The original Web2Access tests required additional elements and the method for the reviews with a mix of automatic and manual checks needed to be overhauled. It was necessary to abide by the UK Government’s guidance stating that web accessibility compliance must include the specific requirements mentioned in the WCAG 2.1 Success Criteria at levels A and AA. This meant that the update to Web2Access needed to include five additional success criteria at Level A and seven at Level AA. Furthermore, in order to allow the reviewers the chance to evaluate more than one web page at a time the Web Accessibility Conformance Evaluation Methodology (WCAG-EM)[[2]](#footnote-2) was included in the update. This enabled the team to produce an automated accessibility statement that could be added to a website as stipulated in the recent Public Sector Bodies (Websites and Mobile Applications) (No.2) Accessibility Regulations 2018 [4]

## **Method**

Much of the original database design of Web2Access remained in place with individual tests having additional text added to incorporate the extra information required for the updated success criteria. Some of the tests were merged to allow for the new ones from WCAG 2.1. However, the web accessibility checker that was developed to assist with any reviews required a new build using an accessible React a JavaScript interface[[3]](#footnote-3). Behind the interface that presents the results of multiple tests, the team implemented the use of the open source Pa11y[[4]](#footnote-4) accessibility checker with the additional checks. An innovative series of visual appraisal pages were integrated within the drop down results to allow the reviewer to see where issues might arise when the automated checking could have produced false positives or negatives. Algorithms were developed to offer the reviewer the chance to check the results, so where the success criteria required for instance no overlaps when using text spacing, a visual representation would have outlined areas where there were suspected issues. The reviewer could then accept or reject the result of the automated test.

AI models were used when it came to evaluating the accuracy of the alternative text offered for images. Pre-trained neural networks model based on the MobileNet23[[5]](#footnote-5) image classifier and the COCO-SSD24[[6]](#footnote-6) model for object detection and classification provided a comparison between the actual image used and the alternative text provided. Once again the output could be seen on the appropriate visual appraisal page, so that the reviewer had the opportunity to make a final decision should there be any doubt resulting from the automatic check. As an addition to the completed work there is the intention to use the work of Sen (2019) [5] and to explore further the issue related to hypertext links that fail to comply with the WCAG 2.1 Success Criterion 2.4.4 Link Purpose (In Context)[[7]](#footnote-7). This is possible by capturing groups of words, such as three words before and after the hyperlink from the source and then comparing this with a similar amount of words at start of the target page using word-embedding techniques. “A two-layer neural net that processes text by “vectorizing” words”[[8]](#footnote-8) called word2vec[6] was chosen as this works with word association and Word Movers Distance (WMD) [7], Sen hypothesised that if the WMD scores were low this should show that the target text and the link text would be helpful to users.

## **Results**

The team never intended to evaluate the success or otherwise of the Pa11y automatic checker but to see whether it was feasible to use AI for some of the new WCAG 2.1 success criteria and would a visual appraisal system reduce the questions that might arise from possible false positives or negatives with some checks. A series of website reviews were performed on a sample taken from the top one million sites, according to alexa.com[[9]](#footnote-9). No false positives or negatives were discovered. Where results were negative the code producing the fault appeared in the collapsible content under the various sections for each success criteria on the results page once the checker had gone through all the pages. It was also possible to see additional results for some of the criteria using the visual appraisal system where issues presented could be checked for accuracy. With regards to the contextual hyperlink detection task, the performance of the word2vec model on the corpus was compared against that of pre-trained embeddings. The results for the model created on the generated dataset looked promising but further work is needed.

Five digital accessibility experts tested the application at various times during the design and implementation phases. Due to time constraints the number of people able to evaluate the accessibility checker for usability purposes was low, but as Nielsen stated in 2000; “The best results come from testing [with] no more than 5 users and running as many small tests as you can afford”[8]. Constant feedback via weekly meetings and the use of Slack[[10]](#footnote-10) for messaging, as part of an iterative process, resulted in several changes over the course of the project. All the experts were able to use the system and commented on the helpfulness of the visual appraisal pages as a way of confirming results from the automatic checker. It was noted how useful it was to have automatically captured images from the web pages showing where issues were arising. These can be helpful for the web developer as well as being able to feed issues into the automatically produced accessibility statement.

## **Conclusion**

There were some limitations to this research in terms of time constraints, failure to access password protected sites in a secure manner and issues around model windows and contrast levels. Nevertheless, there are WCAG 2.1 Success Criteria that can respond to the use of AI despite the fact that there has been little use of machine learning techniques as a way of supporting web accessibility checks in the last few years. Abou-Zahra et al commented that the “significant drawback of artificial intelligence for web accessibility at this time is a lack of accuracy and reliability” [2]. Ultimately, it is hoped that further use of machine learning, neural networks and natural language processing can be implemented with increased accuracy and reliability, especially when there are larger data sets available from accessibility checks. However, these need to be open and available in order to build useful corpora. At present models are dependent on external data sets that have the potential to skew results.

Nevertheless, by using a system that not only offers an increased number of automated checks, but also provides a way of visually appraising issues, such as orientation, overlaps, text spacing and image alt tags, a reviewer can be relatively reassured about an accurate result, especially when the tests are carried out on multiple browsers on both mobile and desktop devices. It is felt that this process has provided a means of speeding up the multiple web page checking process for any large organisation. It also enables digital accessibility experts in a team who may not be coders, to highlight issues that are arising with an increased evidence base.

## **References**

|  |  |
| --- | --- |
|  | W3C Web Content Accessibility Guidelines (WCAG) Web page <https://www.w3.org/WAI/standards-guidelines/wcag/> Accessed 15th April 2020 |
|  | W3C WCAG 2.1 What’s new in WCAG 2.1 Web page <https://www.w3.org/WAI/standards-guidelines/wcag/new-in-21/> Accessed 15th April 2020 |
|  | Abou-Zahra, S., Brewer, J. and Cooper, M., 2018. Artificial Intelligence (AI) for Web Accessibility: Is Conformance Evaluation a Way Forward?. In Proceedings of the Internet of Accessible Things (pp. 1-4). |
|  | UK Government Legislation Public Sector Bodies (Websites and Mobile Applications) (No.2) Accessibility Regulations 2018. Web page <http://www.legislation.gov.uk/uksi/2018/852/contents/made> Accessed 15th April 2020 |
|  | Sen, S., 2019. Artificial Intelligence and Web Accessibility (Unpublished Master’s thesis) Web page <https://shaunaksen.github.io/AI-for-Web-Accessibility/> Accessed 15th April 2020 |
|  | Mikolov, T., Chen, K., Corrado, G. and Dean, J., 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781. |
|  | Kusner, M., Sun, Y., Kolkin, N. and Weinberger, K., 2015. June. From word embeddings to document distances. In International conference on machine learning (pp. 957-966). |
|  | Nielsen, J., 2000. Why you only need to test with 5 users. Web page <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/> Accessed 15th April 2020 |

1. <https://web2access.org.uk/> [↑](#footnote-ref-1)
2. <https://www.w3.org/WAI/test-evaluate/conformance/wcag-em/> [↑](#footnote-ref-2)
3. <https://reactjs.org/> [↑](#footnote-ref-3)
4. <https://pa11y.org/> [↑](#footnote-ref-4)
5. <https://github.com/tensorflow/tfjs-models/tree/master/mobilenet> [↑](#footnote-ref-5)
6. <https://github.com/tensorflow/tfjs-models/tree/master/coco-ssd> [↑](#footnote-ref-6)
7. <https://www.w3.org/TR/WCAG21/#link-purpose-in-context> [↑](#footnote-ref-7)
8. <https://pathmind.com/wiki/word2vec> [↑](#footnote-ref-8)
9. <https://www.alexa.com/topsites> [↑](#footnote-ref-9)
10. <https://slack.com/intl/en-gb/> [↑](#footnote-ref-10)