UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

AALE – AVALIAÇÃO DE ACESSIBILIDADE DE LARGA ESCALA

Beatriz Parrinha Martins

Mestrado em Informática

Dissertação orientada por:
Prof. Doutor Carlos Alberto Pacheco dos Anjos Duarte
e co-orientada pelo Prof. Doutor Luís Manuel Pinto da Rocha Afonso Carriço

2021

Agradecimentos

Agradeço a todos os que me acompanharam nesta fase da minha formação académica, com destaque para os meus orientadores. Quero agradecer em especial ao professor Carlos Duarte pela disponibilidade, empenho e paciência que foram fundamentais para a prossecução dos objetivos deste trabalho, realizado ao longo do último ano.

Resumo

A acessibilidade da web refere-se à inclusão de práticas no desenvolvimento de conteúdos web, por forma a que todos os recursos possam ser utilizados por todos os grupos de utilizadores, incluindo utilizadores com deficiências. Para verificar o estado da acessibilidade de cada recurso web, as avaliações da acessibilidade devem ser consideradas. A avaliação da acessibilidade dos recursos da Web é normalmente desempenhada através da verificação da conformidade de cada recurso com um conjunto de diretrizes definidas por um determinado padrão (por exemplo, as WCAG, Web Content Accessibility Guidelines). O resultado da avaliação indica quais as diretrizes que o recurso Web respeita, e pode incluir outros detalhes importantes referentes ao número de elementos. Tipicamente, o conteúdo a ser avaliado é selecionado de acordo com os requisitos do estudo e as respetivas restrições. No que diz respeito à avaliação da acessibilidade, esta pode ser efetuada de forma manual, automática ou híbrida. A escolha do método mais adequado a utilizar tem como base o contexto do estudo, bem como os recursos disponíveis.

As avaliações da acessibilidade em larga escala, i.e. avaliação da acessibilidade de um grande conjunto de páginas web, pretendem observar a evolução e/ou o estado da acessibilidade da Web. Deste modo, é possível conhecer quais os aspetos mais relevantes a serem melhorados e instituir boas práticas no desenvolvimento de conteúdo web. Por esta razão, foi realizado um estudo da acessibilidade em larga escala, com o objetivo de avaliar alguns aspetos relativos à acessibilidade da Web. Primeiramente, foi realizada uma revisão da literatura referente a avaliações da acessibilidade automáticas e avaliações em larga escala. De seguida, através da ferramenta automática QualWeb desenvolvida pelo Departamento de Informática na Faculdade de Ciências da Universidade de Lisboa, foram executadas avaliações da acessibilidade de 2.8 milhões de páginas web. Com base nos resultados das avaliações, foi possível obter conclusões e estudar o estado da acessibilidade, por forma a compreender quais os problemas mais frequentemente reportados. Como exemplo, a maioria das páginas web avaliadas apresentam problemas relacionados com o contraste do texto. Os restantes principais problemas, que foram identificados em mais de 20% das páginas, focam-se na ausência de um nome acessível em certos elementos HTML. A solução para este tipo de situações é relativamente simples e podia ser facilmente evitada, se os desenvolvedores de conteúdo web apresentassem maior sensibilidade e consciência sobre o assunto.

Os resultados provenientes de avaliações da acessibilidade podem ainda ser analisados com o intuito de entender se existem certos aspetos que podem influenciar esses resultados. No desenvolvimento de um site web são utilizadas diversas tecnologias com determinados propósitos. No entanto, o desenvolvimento e manutenção de um site web é uma tarefa desafiante, podendo por vezes influenciar outros âmbitos como a acessibilidade dos conteúdos a serem desenvolvidos. Isto significa que as tecnologias utilizadas nestes processos podem ter influência na acessibilidade do conteúdo web. Neste sentido, os

resultados da identificação de tecnologias utilizadas no desenvolvimento de 166 mil sites web, pelas ferramentas automáticas Wappalyzer e SimilarTech, alinhados com os resultados do QualWeb foram explorados. Esta investigação consistiu na realização dos testes estatísticos Mann-Whitney e Kruskal-Wallis e ainda testes post-hoc, quando aplicável. O primeiro teste teve o objetivo de perceber se as categorias de tecnologias influenciavam a acessibilidade das páginas e se essa influência era positiva ou negativa. O segundo teste pretendeu estudar as diferenças entre as várias tecnologias de cada categoria, no que diz respeito à influência que têm na respetiva categoria. Os testes post-hoc visaram explorar se as tecnologias das categorias consideradas são estatisticamente diferentes. As conclusões deste estudo demonstraram haver certas tecnologias e respetivas categorias que foram aplicadas no desenvolvimento do conteúdo web de páginas mais acessíveis ou menos acessíveis. Deste modo, os resultados das análises permitem perceber o impacto das tecnologias na acessibilidade web e quais delas podem implicar pontuações de acessibilidade mais positivas ou mais negativas.

Para além deste aspeto relacionado com as tecnologias web, e uma vez que a comparação dos níveis de acessibilidade de diferentes recursos ou de diferentes versões do mesmo recurso, na maioria das vezes é complexa, a utilização de métricas de acessibilidade para sintetizar o nível de acessibilidade de um recurso web em um (ou mais de um) valor quantificável, pode ser considerada. As métricas de acessibilidade são importantes quando há a necessidade de comparar o nível de acessibilidade de sites web. Esta comparação está a tornar-se um caso de uso relevante para agências de monitorização da acessibilidade ou para as organizações que avaliam o impacto das alterações ao nível da acessibilidade dos seus recursos como parte dos seus processos de garantia de qualidade. No que diz respeito às métricas de acessibilidade, é interessante perceber se elas se relacionam e que agrupamentos conseguem formar, de modo a facilitar a escolha de uma ou mais métricas num determinado contexto de estudo ou investigação. Consequentemente, foi possível calcular 11 métricas de acessibilidade web referentes a páginas e sites web identificadas na literatura, a partir dos resultados obtidos pelo QualWeb relativos aos testes verificados nas páginas web. As relações entre as várias métricas de acessibilidade foram exploradas, bem como as semelhanças e diferenças entre os diversos grupos formados. As métricas foram agrupadas através do método de Clustering Hierárquico, que associa as métricas consoante as suas semelhanças. Posteriormente, foi realizada uma outra análise que estuda a validade de cada uma das métricas de páginas e de sites web. Um conjunto de páginas acessíveis e inacessíveis foi selecionado de acordo com fontes que afirmam a natureza da acessibilidade das páginas. Foram aplicadas todas as métricas de acessibilidade com o objetivo de perceber quais reportavam os resultados mais coerentes e viáveis relativamente às páginas que afirmavam ser mais acessíveis e menos acessíveis.

Ao longo de toda a pesquisa realizada neste trabalho, é percetível uma quantidade considerável de estudos e análises realizados somente com recurso a *home pages* de vários sites web. Esta preferência coloca a dúvida relativa à representatividade dos níveis de acessibilidade destas páginas face aos níveis de acessibilidade das restantes páginas dos sites web. Por essa razão, as diferenças entre a acessibilidade das *home pages* e das restantes páginas de cada site web foi medida. Esta medição foi realizada através do estudo da correlação entre as pontuações da métrica A3 das *home pages* com as pontuações da mesma métrica das páginas interiores dos vários sites web. Para além deste estudo, foram também analisados os problemas que as páginas interiores e as *home pages* reportam, com o intuito em perceber se diferentes tipos de erros de acessibilidade são identificados nas duas abordagens distintas. Com esta análise foi

possível concluir que existe uma correlação forte entre os níveis de acessibilidade das *home pages* e das páginas interiores e que foram detetados os mesmos problemas de acessibilidade em ambos os tipos de páginas. Como consequência, é possível afirmar que a acessibilidade das *home pages* pode ser utilizada como referência para a acessibilidade de todo o site web.

Contudo, e devido ao facto de este trabalho ser unicamente focado em avaliações e ferramentas automáticas, existem problemas de acessibilidade que as ferramentas automáticas não conseguem detetar, limitando, assim, as conclusões obtidas. Não obstante, considerar avaliações manuais realizadas por peritos, mesmo que a grupos específicos de páginas, seria um processo inviável e bastante custoso, considerando o número elevado de páginas avaliadas.

Palavras-chave: Avaliação da Acessibilidade Web em Larga-escala, Tecnologias Web, Métricas de Acessibilidade Web, Acessibilidade de *Home Pages*, QualWeb

Abstract

Evaluating the accessibility of web resources is usually performed by checking the conformance of the resource against a standard or set of guidelines (e.g. the WCAG 2.1). The result of the evaluation will indicate what guidelines are respected (or not) by the resource, as well as other important details. Typically, the content to be evaluated is selected according to the study requirements and constraints.

Accessibility evaluation results can be analyzed in order to understand if there are certain aspects that can influence them. Since developing and maintaining a website is always a challenging task, the technologies used in this process may have influence in the accessibility of the web content. In this regard, several analyzes were performed that demonstrated that certain web technologies may entail more positive or negative accessibility scores.

Aside from this, and since the comparison of the accessibility levels of different resources or of different versions of the same resource most of the time is complex, the use of web accessibility metrics to synthesize the accessibility level of a web resource into one (or more than one) quantifiable value can be reached. Therefore, they are important when there is a need to compare websites. In this regard, it is interesting to understand if they relate to each other and the groups they can form. Consequently, the relationships between several accessibility metrics were explored, as well as the similarities and differences between the groups.

Over the conducted research in this thesis, it is noticeable a considerable quantity of studies that rely only on the home pages of a set of websites. This preference raises the doubt related to the representativeness of these home pages' accessibility levels compared to the remaining website pages' accessibility levels. For this reason, their accessibility levels and reported issues were compared. Based on this analysis, it was possible to conclude that the accessibility of the home pages can be used as a reference for the overall website accessibility.

To achieve the above purposes, we used the QualWeb automated engine to perform the large-scale web accessibility evaluation over 2.8 million web pages and Wappalyzer and SimilarTech automated tools to identify the web technologies of the pages' domains. From the evaluations' results obtained using QualWeb, we computed 11 web accessibility metrics and we measured the difference between the accessibility of the home pages and the remaining website pages.

Keywords: Large-scale Web Accessibility Evaluation, Web Technologies, Web Accessibility Metrics, Home Pages Accessibility, QualWeb

Contents

List of Figures

List of Tables

Chapter 1

Introduction

As the Web becomes more available, it is more challenging to guarantee all groups of users can access it and use its resources, regardless of their disabilities. Since different users have distinct interactions, it is crucial to develop web content always aiming at minimizing the barriers the users could find that would hinder the interaction. The barriers users find during the interaction end up excluding them from using and accessing the Web. To avoid excluding people from utilizing web services, accessibility is key to create high-quality web content.

1.1 Motivation

Web accessibility is responsible for making the Web (i.e., World Wide Web) open to all groups of users, including users with disabilities. It aims to achieve high quality in the websites and web tools creation and development, making products more accessible to any user and avoiding the occurrence of barriers during the interaction. It is defined as the availability and usability of the web resources by every single individual, no matter his/her disabilities. "Web accessibility means that websites, tools, and technologies are designed and developed so that people with disabilities can use them" [W3C, 2005]. The fact that a website is accessible to users with different disabilities, ensures that the web becomes a place without social exclusion, where all users have the same opportunities and access. It also improves the user experience for everyone.

One of the biggest challenges in website development and maintenance is the inclusion of practices that improve the accessibility to all types of users, by using specific tools or technologies. Web technologies are tools and techniques capable of establishing a communication between different types of devices [Reeves, 2019], allowing the creation of platforms and applications for the Web. These web technologies are going to structure the web page to make sure the best user experience is given. Since there are various factors associated with web accessibility that can impact its level, either positively or negatively, web technologies used during the development of a website may be one of those factors with a considerable impact on the accessibility level, as they are the root of every web application.

In order to study the web technologies impact in web accessibility, the accessibility of the web must be assessed through an evaluation. The accessibility of the Web can be evaluated by verifying the conformance with standards or guidelines, the most common ones being the Web Content Accessibility Guide-

lines (WCAG)[1]. Testing accessibility conformance can be performed automatically, semi-automatically or manually. These three different types of evaluation require the adoption of specific tools and procedures. According to the context that is being applied, there is the need to choose what type of evaluation is going to be performed. In several cases, it is possible to combine the different types all together to have the best of all worlds. Nevertheless, this is not always possible. For instance, on a large-scale web accessibility evaluation, manual procedures are not cost-effective, since they need human experts to perform verification of hundreds, thousands, or millions of web pages. Therefore, automatic tools are the best option for these instances, even though their limited coverage of the guidelines may compromise the results.

Before performing a web accessibility evaluation, there is the need to select a sample of web pages that are going to be assessed. Each web page reports its own accessibility results. Since web pages from a given website may report similar results, a lot of studies performed accessibility evaluations over the home page of the website, assuming that the results of that page are valid enough to be representative of the overall website accessibility. However, this assumption may not hold because of the different types of interaction that are provided in different pages.

Given that the evaluation results typically show the accessibility of a page or site in terms of conformance to the set of guidelines being checked, it is not always easy to gauge the accessibility level of the evaluated resource. Approaches such as considering a resource accessible only if it conforms to all the guidelines checked are easy to understand, but do not support understanding how far from being accessible the resource is. Approaches that capture the nuances in the levels of accessibility of web pages or sites could be more useful. This is what web accessibility metrics try to achieve. Web accessibility metrics measure the accessibility level of websites or web pages, by measuring web properties. The number of links or the size of an HTML file are two examples of metrics that can be computed for a site or web page [Vigo et al., 2012]. Web accessibility metrics are formulas that are applied using data provided by accessibility evaluations. This data can be gathered manually, semi-automatically or automatically. For instance, there are metrics that use data collected through experts' procedures which, when conducting large-scale evaluations (e.g., comparing multiple sites or monitoring a site with hundreds of pages), make them an expensive and impractical choice.

Considering that there is a large number of web accessibility metrics available for researchers, auditors or practitioners to choose from, the uncertainty about which one(s) should be used emerges. To help provide an answer to this question, it is important to understand how these web accessibility metrics relate to each other and if it is possible to group them according to their similarities and understand the differences between each group.

1.2 Objectives

The main objective of this work is to analyze several aspects regarding web accessibility. In particular, this work aims to:

1. Study web accessibility in a large scale, by evaluating the accessibility of millions of web pages;

[1] https://www.w3.org/WAI/standards-guidelines/wcag/

2. Infer the way web technologies impact web accessibility;
3. Compare existing web accessibility metrics concerning their relationships;
4. Examine whether the accessibility of a home page is representative of the overall website accessibility.

To address all the above-mentioned aspects, a large-scale accessibility evaluation was performed over 2.8 million web pages, as well as the identification of their domains' web technologies. The obtained data results allowed conducting analyzes to describe the impact the web technologies have in web accessibility, to understand the differences of the web pages' accessibility and to identify existing relationships between 11 computed web accessibility metrics.

1.3 Contributions

The work presented in this document led to the following contributions:

- Analysis of the accessibility of millions of web pages and web technologies that were identified in those pages' domains.
- Conclude which web technologies influence the web accessibility of the evaluated web pages.
- Analysis and comparison of existing web accessibility metrics.
- Investigate whether the accessibility of the home pages can be representative of all website accessibility.

1.4 Structure of the document

This document is organized as follows:

- **Chapter 2 – Related work**: this chapter provides a background about web accessibility evaluation, large-scale web accessibility evaluation and web accessibility metrics. It explores how web accessibility is evaluated in different contexts, including large-scale approaches. Nineteen web accessibility metrics that were proposed in the literature are reviewed.
- **Chapter 3 – Large-scale accessibility evaluation**: this chapter presents the development phase concerning the large-scale architecture that was used to perform the accessibility evaluation as well as the web technologies identification. It explains how the implementation was performed and describes the execution of a pre-test with 13 thousand web pages.
- **Chapter 4 – Analysis of the large-scale evaluation results**: this chapter presents and discusses the results of the large-scale evaluation.
- **Chapter 5 – Comparing accessibility metrics**: here it is possible to find the results of the 11 web accessibility metrics analysis that were computed from the accessibility evaluations' results. The results are discussed afterwards.

- **Chapter 6 – Comparing the accessibility of the home pages with the remaining website's accessibility**: this section discusses whether the accessibility of the home pages can be representative of all website pages.
- **Chapter 7 – Conclusions**: in this chapter, we conclude the thesis with a summary of the most important considerations.

Chapter 2

Related work

Web accessibility aspires to achieve an open Web to disabled users [Harper and Yesilada, 2008]. According to Brajnik [2004], a website is accessible when it can be perceived, operated and understood by all groups of users.

The accessibility of the web content can be assessed using different mechanisms including automatic, semi-automatic or manual evaluations, to verify the conformance with standard guidelines. After evaluating the accessibility, it is possible to determine its level through web accessibility metrics.

This section covers accessibility evaluation, in particular using automated engines. It considers large-scale approaches, including some that were performed using only home pages. It finalizes with a description of 19 web accessibility metrics that have been proposed in the literature.

2.1 Web Accessibility Evaluation

To assure a website or web application is accessible, it is crucial to evaluate its accessibility to identify accessibility barriers and try to address them as early as possible. In most cases, accessibility is evaluated through a validation of the web content conformance with standards recommendations. The Web Content Accessibility Guidelines (WCAG) [Henry, 2021] are the most used standards that cover a wide range of the accessibility references. WCAG has 12 to 13 guidelines (depending on the version) that are organized into 4 principles: Perceivable, Operable, Understandable and Robust. Each guideline has testable success criteria that are grouped into three conformance levels: A, AA and AAA.

There are three types of web accessibility evaluation: automated testing, manual inspection, and semi-automated testing. The automated testing is based on automatic tools that analyze the web page code and verify compliance with the accessibility guidelines [Abascal et al., 2019].

Manual inspections are conducted by human experts that generally verify the conformance with a list of accessibility criteria [Abascal et al., 2019]. An example of this type of evaluation is the Barrier Walkthrough [Brajnik, 2009] that aims to evaluate the web accessibility from the identification of the frequency and severity of website accessibility barriers. Although it is more reliable and less likely to identify false positives (flagged accessibility problems that do not exist) or false negatives (accessibility problems that are not detected), it becomes more limited, as the web pages sample increases. Manual testing is desirable when accessibility evaluators want to evaluate the accessibility of a few web pages. As the number of web pages or websites to be evaluated increases, the need to use automatic procedures

also increases.

Semi-automated testing performs automated evaluations that are guided by human expert evaluators. The use of a semi-automated approach can assure a better accessibility evaluation, covering more accessibility aspects. Nevertheless, these approaches cover the limitations of the manual and the automated testing, previously discussed. Depending on the study context, the choice of the best approach to use should take into consideration all the drawbacks.

With the scope of this work being large scale accessibility evaluation, automated evaluation is the only viable option given the available resources. Therefore, this section will focus on automated web accessibility evaluation.

2.1.1 Automated tools for web accessibility evaluation

Several tools and methods are used to evaluate the accessibility of a website or a web page. These methods are organized in three categories: automated testing, manual inspection, and semi-automated testing, as previously discussed. This section focuses on tools to assist automated evaluation.

Many authors studied and compared a set of accessibility evaluation automated tools. For instance, in 2004, Brajnik published a work that discusses and illustrates a method that is able to compare a pair of accessibility evaluation tools [Brajnik, 2004]. This method takes into account the correctness, completeness and specificity of each tool. These three measurements support the evaluation of a website conformance with guidelines. The completeness wants to minimize the number of accessibility violations that were not identified, i.e., false negatives. The correctness aims to minimize the number of violations that are incorrectly identified, i.e., false positives. The tool's specificity defines the number of different accessibility problems that a tool can identify and describe. These three measurements are, later, considered by Vigo, Brown, and Conway [2013] who measured the harm of sole reliance on automated tests by studying six automated tools (AChecker, SortSite, Total Validator, TAW, Deque, AMP). The evaluated six automated tools [Vigo et al., 2013] have in common their capability to automatically verify the conformance with the WCAG 2.0 guidelines. All errors produced by the automated tools were reviewed in order to find false positives, i.e., problems that are incorrectly identified by the automatic tool. The authors stated that the more inaccessible a website is, the higher the tool completeness, since the errors will be easily found. When the website is accessible, the automated tool struggles to find its accessibility issues. Also, the results indicate that even the tool that is considered to have the best performance (TAW) compared to the remaining tools, is far from being an optimal tool.

Besides the Vigo, Brown, and Conway [2013] study, Padure and Pribeanu [2019] also explored the differences between five accessibility evaluation tools. This study analyzes five accessibility testing tools (AChecker, Cynthia Says, TAW, Wave, and Total Validator) and illustrates them with three case studies of Romanian websites. Results show substantial differences in terms of the number of errors that each tool can find, suggesting that we should use more than one tool when performing the accessibility evaluation.

Frazão and Duarte compared the quality of eight free accessibility evaluation plugins extensions for the Chrome browser (aXe Chrome Plugin, Tenon Check, Wave Chrome Extension, TotalValidator, ACCESS Assistant Community, Microsoft Accessibility Insights and ARC Toolkit). The goal was to analyze (1) the differences between all these eight tools, (2) how they are implemented, (3) what success criteria they use, (4) how many tests for each success criteria, (5) the overall differences between the

reports obtained by each tool and their results, and (6) whether they perform only automated tests or manual testing. The quantity of errors found per success criteria by the different tools was analyzed. Results show that the tools provide limited coverage of the success criteria and the 4.1.1 (Parsing) success criteria [W3C, 2016] was the most often violated. Similar to Padure and Pribeanu, this article's authors recommend using more than one automated tool and adding manual inspections to complement the automated evaluation.

2.1.2 Discussion on automated tools

From the literature reviewed in section 2.1.1, it is possible to retain some important aspects regarding automated tools. Although automated accessibility evaluation tools reveal some benefits as they (1) are a fast and efficient way to check the accessibility [Padure and Pribeanu, 2019], (2) have the possibility to provide recommendations that help correcting the problems [Frazão and Duarte, 2020, Padure and Pribeanu, 2019], (3) and are also cost-effective and affordable when a large number of websites are evaluated, they represent a disadvantage when we do not complement this evaluation method.

Even if there are enough resources that allow the combination of different accessibility tools to improve the coverage, as reported in [Frazão and Duarte, 2020], this increase will always be limited, due to the fact that automated tools do not cover all checkpoints [Lim et al., 2020, Kimmons, 2017, Lopes et al., 2010], specially those that need human interpretation. For instance, Vigo et al. [2013] stated that the success criteria coverage of the analyzed automated tools is, at most, 50%. This means that 1 out of 2 success criteria is not analyzed. Frazão and Duarte also concluded that the automated tools cannot cover all success criteria: if all the eight tools are merged, only 62 WCAG success criteria out of 78 are tested.

Since automated tools cannot automatically detect all conformance failures, evaluators should not solely depend on automated tools when evaluating web accessibility [Frazão and Duarte, 2020, Vigo et al., 2013, Duarte et al., 2016]. Thus, human judgement is necessary to complement this approach, so it is possible to interpret conflictive situations and verify if the tool correctly identified the accessibility problems. Nevertheless, there are limitations about manual testing. For instance, when it is necessary to perform the evaluation of many websites, the expert evaluation becomes more limited in terms of scalability. This task gets exhaustive when each evaluator has to evaluate a large set of web pages. If each evaluator, however, evaluates only one page, it reduces the exhaustion. Yet, it gets more expensive as a higher number of evaluations is needed.

2.2 Large scale web accessibility evaluation

A large-scale accessibility evaluation performs evaluations of hundreds, thousands or millions web pages or websites. It is useful to study general accessibility aspects that should be representative having a large data sample. To perform these large-scale evaluations, automated tools are the best option to consider. They are a more cost-effective process to evaluate the accessibility of a larger number of web pages, avoiding conducting manual evaluations.

2.2.1 Large-scale accessibility evaluation approaches

Several authors performed large-scale accessibility evaluations, which means they evaluated the accessibility of a considerable number of web pages. By means of a large-scale evaluation, it is possible to conclude general aspects about a certain context. For instance, if there is the need to understand the level of accessibility between two time periods, a large-scale evaluation is executed in order to have more representative data results. Rau et al. did such research in 2014, where it is possible to observe the changes and examine the evolution regarding web accessibility in China from 2009 and 2013. For this purpose, the authors studied the accessibility evaluation of websites from 2009 and the evaluation of websites from 2013. First, they choose the 100 most popular websites from 2009. Then, they classified each website into categories according to their content and removed the categories with no interest for this study. The authors only evaluated the home pages of the most popular websites, due to constraints related to time and resources. None of the websites met the basic accessibility requirements. However, the results show that, in 2013, people were more conscious about web accessibility.

Besides analyzing the evolution of the accessibility or even the accessibility status of various countries [Snaprud and Sawicka, 2007], large-scale evaluations enable the investigation of factors that can somehow impact the web accessibility. As Duarte et al. did in 2016, 1669 web pages were evaluated and the web technologies used in the development were identified in order to understand if web technologies could influence the web accessibility. After the accessibility evaluation using QualWeb and the technologies identification using Wappalyzer, the authors computed three web accessibility metrics (conservative, optimistic and strict) based on QualWeb reports, and concluded that web technologies have a significant impact in web accessibility. Another study [WebAIM, 2021] performed a similar analysis regarding web accessibility and web technologies, where results could identify a set of technologies that may lead to more accessibility errors. The authors of the [WebAIM, 2021] study, conducted accessibility assessments over three years, starting in 2019. They concluded that the number of accessibility errors and WCAG conformance failures decreased in 2021.

Other aspects like the correlation between the number of HTML nodes (i.e., web site complexity) and accessibility levels [Lopes et al., 2010] can also be achieved through a large-scale evaluation. Lopes et al. verified that as the number of HTML elements increases, the accessibility quality rate decreases, which leads to the hypothesis of a more complex website tends to have a lower accessibility quality.

Furthermore, large-scale evaluations can co-operate with organizations and institutions on behalf of an accessibility overview. For instance, Kimmons evaluated the accessibility of the home pages of Institutions of Higher Education, so it would be possible to obtain an accessibility overview of this matter. Similar to [Kimmons, 2017] study, WebAIM also evaluated the accessibility from 2019 and 2021 and focused the majority of its evaluations on home pages.

Besides all studies that aim to perceive and relate web accessibility in different fields, in 2016, an approach capable of reducing the number of pages that are necessary to perform a large-scale evaluation, using machine learning, was developed [Mucha et al., 2016]. By applying clustering methods, it is possible to group similar pages according to their accessibility barriers. The groups of pages are based on the large-scale accessibility evaluation results. After having the clusters, the authors randomly analyzed their members. Using this approach, it is possible to reduce the number of pages needed to calculate the accessibility level and determine the accessibility problems, since each cluster is composed of a set of

web pages that have similar accessibility problems. Thus, it is easier to solve the accessibility problems by pointing to the problematic cluster.

Generally, large-scale accessibility studies want to perceive and conclude several points regarding a specific context. These large-scale assessments can help locating and evaluating potential barriers, as well as encouraging developers to improve the accessibility of the Internet [Snaprud et al., 2006]. Nevertheless, the higher the web pages sample of the accessibility evaluations, the more resources are required to conduct manual evaluations, becoming a more expensive procedure.

2.2.2 Discussion on large-scale accessibility assessments

To do large-scale accessibility analysis, automated evaluation might be the most appropriate alternative, since it provides scalability and objectivity. Since large-scale accessibility assessments are performed over large sets of web pages, the manual or semi-manual approaches are not viable nor efficient. Nevertheless, when choosing among several automated procedures, it is important to retain the disadvantages previously discussed in section 2.1.2, that lay over the fact that expert evaluations offer more reliable and deep results in comparison with these automated evaluations [Lopes et al., 2010].

In sum, large-scale accessibility evaluations allow the observation and comparison of possible changes in the accessibility of the Web. Since it is a representative method to obtain a generalization about the accessibility in several contexts, it can be used to perceive the current level and progress of the web accessibility. In the presented studies, the large-scale evaluations allowed the comparison of the web accessibility between periods of times and between countries or educational institutions. Besides evaluating the accessibility levels, it is also possible to uncover reasons that explain them. For instance, the number of HTML elements and web technologies used in the development of the website are two matters that have shown an impact in the accessibility levels. Although developers are more aware and conscious about web accessibility nowadays, there is still a long way to go to make the Web accessible to everyone.

Apart from the above, an interesting aspect regarding the large-scale analyzed approaches in section 2.2.1, is the fact that some studies do perform and investigate the web accessibility of the Web using only home pages [Rau et al., 2014, WebAIM, 2021, Kimmons, 2017]. Nevertheless, the accessibility of the home pages might not be a reference of overall website accessibility since the website accessibility might not solely depend on the home page accessibility. This limitation runs the risk of exhibiting accessibility results that are not coherent with the remaining web pages' results of a given website.

2.3 Web accessibility metrics

According to Brajnik, Vigo, and Connor [2014], web metrics measure websites' or web pages' properties. These metrics can summarize results obtained from a guideline review based evaluation [Freire et al., 2008a]. Additionally, the authors of [Song et al., 2018] state that web accessibility metrics have the ability to measure the accessibility levels of websites.

Metrics should meet five different aspects [Freire et al., 2008b]. They should (1) be simple to understand; (2) be precisely defined; (3) be objective; (4) be cost-effective; and (5) give such information so it is possible to have meaningful interpretations. The authors also mentioned that web accessibility metrics are important to understand, control and improve products and processes in companies. Nevertheless,

they state that it is not possible to define which metric is more effective, since it depends on the project in question and its needs.

In [Parmanto and Zeng, 2005], the authors believe that an accessibility metric should be summarized into a quantitative score that provides a continuous range of values so it is possible to understand how accessible and inaccessible the web content is. It is also important to guarantee that the range of the metric's values can have other discriminations besides accessible and inaccessible. A high-quality metric should also consider the complexity of the websites. It would be convenient if the accessibility metric could be scalable to conduct a large-scale accessibility evaluation.

In conclusion, metrics are useful to process and understand the results obtained from an accessibility evaluation. This approach can also help ranking web pages or even explore the accessibility level of web pages or websites. The computation of accessibility metrics can produce, as a result, ordinal or quantitative values. Depending on how the data is collected, accessibility metrics can be automated if the data is based on automatic accessibility evaluation tools, manual if the data derived from human judgments or semi-automated if it is based on data produced by tools and interpreted by experts.

2.3.1 Theoretical Background

Before diving into the web accessibility metrics' details, it is important to clarify some concepts that help understanding how each metric behaves.

Some metrics use the barrier concept. A barrier is a condition caused by the website or web page that prevents the user's access to the web content [Abuaddous et al., 2017], i.e. a problem found in a certain website or web page that creates an inaccessible interaction between the user and the web content. Each barrier can have different levels of severity.

Whenever an automatic accessibility evaluation is performed, its outcome varies according to the compliance with standard recommendations. Different outcomes are considered by different metrics, but they can be summarized into: (1) pass, which means that the web content fulfills a certain recommendation; (2) fail, which indicates that the web content does not meet the recommendation; (3) warning, an outcome produced by automated evaluation tools to represent those instances where the tool could not determine the conformance, or lack or conformance, with the recommendation, and is therefore required the intervention of an human expert.

Besides the above aspects, it is important to note that some of the web accessibility metrics that have been reviewed verify the conformance with checkpoints and these checkpoints are grouped into priority levels: priority 1, priority 2 or priority 3. The priority levels in some metrics have associated weights that vary from 0 to 1. This applies to metrics proposed before the introduction of WCAG 2.0. Metrics proposed after WCAG 2.0 typically verify conformance with success criteria grouped at conformance levels A, AA and AAA.

In the following, we present the metrics we found by searching the existing literature on web accessibility. For each metric, we describe the data it is based on, its output range, and any other considerations regarding its application (e.g., if it is applicable to web pages or web sites).

Failure-Rate (FR)

The Failure Rate (FR) was developed by Sullivan and Matson in 2000 [Sullivan and Matson, 2000]. According to Vigo et al. [2009], this metric relates the actual points of failure with the potential points of failure. For instance, if a web page has 10 images, all these images are potential barriers if they are not properly defined. And if 5 out of these 10 images do not have a proper alternative text, according to the accessibility evaluator, they are actual barriers.

A point of failure can be interpreted in two ways: as an accessibility problem or barrier that occurs on web page's elements preventing the interaction of a user with the web content; or as the elements that cause accessibility problems. In the first interpretation, each element can have multiple points of failure, which allows us to count more accessibility problems and so, better estimate the accessibility level. Therefore, we decided to consider a point of failure as an accessibility problem that occurs on a web page. Consequently, the failure rate can be the ratio between the actual problems that were encountered in a web page and the potential barriers, i.e., all potential problems of a web page that can lead to accessibility issues if they are not properly designed.

Vigo and Brajnik [2011] say the failure rate quantitatively measures the accessibility conformance, having a score from 0 to 1. A web page with a failure rate of 0, is totally accessible whereas a totally inaccessible web page has a failure rate score of 1.

The simplicity of this metric can be explained with the fact that it does not consider the error nature, i.e., "whether checkpoints are automatic errors, warnings or generic problems" [Vigo et al., 2007], or the fact that it does not take into consideration the checkpoints' weights.

$$I_p = \frac{B_p}{P_p} \tag{2.1}$$

Equation 2.1 presents the formula for computing the Failure Rate metric, where I_p is the Failure Rate final score, B_p identifies the actual points of failure, and P_p identifies the potential points of failure.

Unified Web Evaluation Methodology (UWEM)

According to Sirithumgul et al., UWEM 1.0 is an improved UWEM version that was developed in 2006. It is based on user feedback rather than WCAG priority levels [Song et al., 2018]. The final value of this metric represents a probability of finding a barrier in a website or web page that could prevent users from completing a certain task [Freire et al., 2008b,a, 2009]. This metric also considers the potential problems and barriers' weights. The UWEM formula is based on the product of the checkpoints' failure rates [Freire et al., 2009]. Its results are precise and accurate, however, it only takes into consideration 2 priority levels of the WCAG guidelines [Martínez et al., 2009].

The formula can be interpreted as a web page score or a website score. If the website score is wanted, then the UWEM formula will be the sum of the UWEM score of each webpage divided by the total number of pages of that website, i.e. the arithmetic mean.

This formula's final score varies between 0 and 1, where 0 means the web page is accessible and 1 means the web page is inaccessible.

$$UWEM = 1 - \prod 1 - \frac{B_i}{P_i} W_i \tag{2.2}$$

Equation 2.2 presents the formula for computing the UWEM metric, where B_i is the total of actual points of failure of a checkpoint i, P_i is the total of potential points of failure of a checkpoint i, and W_i identifies the severity of a certain barrier i (this weight is calculated by simple heuristics, by combining the results of an automatic evaluation and manual testing or by disabled users feedback [Bühler et al., 2006]).

A3

In 2006, Bühler et al. proposed some changes to the UWEM 0.5 metric, in particular, some probability properties were used as well as some issues related to the complexity of the web page were aggregated. A3 is an improved aggregation formula based on UWEM [Freire et al., 2008b,a, 2009]. Similar to UWEM, A3 also considers the failure rate, i.e., the ratio between the number of barriers (violation of a given checkpoint) and the total number of potential barriers. UWEM and A3 consider the barriers weights coefficients based on the impact on the user of each given barrier [Freire et al., 2008b].

According to Vigo and Brajnik [2011], the range of a metric "is important because it tells whether the metric uses all the possible output values rather than squeezing all the results onto a smaller range". This metric produces a small range of values, that are all between 0 and 1, where 0 means the web page is accessible whereas 1 means the web page is inaccessible.

$$A3 = 1 - \prod_{b}(1 - F_b)^{\frac{B_{pb}}{N_{pb}} + \frac{B_{pb}}{B_p}} \tag{2.3}$$

Equation 2.3 presents the formula for computing the A3 metric, where B_{pb} is the total of actual points of failure of a checkpoint b in page p, b is the barrier (checkpoint violation), N_{pb} is the total of potential points of failure of a checkpoint b in page p, and F_b identifies the severity of a certain barrier b (this weight is calculated by simple heuristics, by combining the results of an automatic evaluation and manual testing or by disabled users feedback [Bühler et al., 2006]).

The authors of this metric performed an experimental study to compare the results between A3 and UWEM and understand the differences between them. A checkpoint weight of 0.05 was used for all checkpoints, considering that all of them would have the same importance. This experiment was conducted with a group of six disabled users that evaluated six web pages. After applying both metrics, the authors concluded that A3 outperformed UWEM in the experiment [Freire et al., 2008a].

Web Accessibility Barriers (WAB)

The WAB metric was proposed by Hackett et al. in 2004 [Vigo and Brajnik, 2011]. Parmanto and Zeng proposed a new version of the WAB metric in 2005. It quantitatively measures the accessibility of a web site considering the 25 WCAG 1.0 checkpoints (5 checkpoints in Priority 1, 13 checkpoints in Priority 2, and 7 checkpoints in Priority 3). It applies the concepts of potential problems and weights of the barriers. Barriers' weights are related to the relative importance of a given checkpoint. It takes into consideration the total number of pages of a certain website. The WAB formula is defined as the ratio between the sum of the failure rate of each checkpoint and the priority of that checkpoint [Vigo and Brajnik, 2011]. The arithmetic mean of all pages of a website represents the metric score for that website. The Hackett

and the Parmanto & Zeng [Vigo and Brajnik, 2011] formulas are represented in equations 2.4 and 2.5, respectively.

The range of this metric's values is not normalized [Vigo et al., 2007], as there is no limit for this metric's score. The only reference this metric has is the higher its score, the worse the accessibility level of the website.

Besides, checkpoint weighting does not have solid empirical foundations. Since it takes into consideration 25 WCAG checkpoints out of 65, this metric offers a guideline support of 38%. Nevertheless, according to Vigo and Brajnik [2011], WAB is the best individual metric compared to A3, Page Measure (PM) and Web Accessibility Quantitative Metric (WAQM) since it yields an accuracy rate of 96%.

$$WAB = \frac{1}{N_p} \sum_{p} \sum_{c} \frac{fr(p,c)}{priority_c} \tag{2.4}$$

Equation 2.4 presents the formula for computing the WAB by Hackett metric, where $fr(p,c)$ is the failure rate of a certain checkpoint c in web page p, $priority_c$ identifies the priority level of the checkpoint c (1, 2 or 3), and N_p is the total number of web pages of a given website.

$$WAB = \frac{\sum_{j=1}^{T} \sum_{i=1}^{n} (\frac{b_{ij}}{B_{ij}})(W_i)}{T} \tag{2.5}$$

Equation 2.5 presents the formula for computing the WAB by Parmanto and Zeng metric, where b_{ij} is the number of actual violations of checkpoint i in page j, B_{ij} is the number of potential violations of checkpoint i in page j, n is the total number of checkpoints, W_i identifies the weight of the checkpoint c, according to its priority level (this weight is calculated from experiments with users with different disabilities [Freire et al., 2008a]), and T is the total number of web pages of a given website.

According to [Parmanto and Zeng, 2005], Parmanto and Zeng weighted the priority levels in the calculation of the WAB score. Priority 1 violations represent a higher weight score since web pages with this level of violations are more difficult to access by people with disabilities.

Martínez, Juan, Álvarez, and del Carmen Suárez [2009] went further and created a quantitative metric based on the WAB metric: *WAB*$*$. The *WAB*$*$ metric is based on WAB and has some UWEM-like extensions. It gets the WAB's precision of the accessibility score and uses more detailed checkpoints, as UWEM does. With all these tools, the authors could build a new metric *WAB*$*$. According to Martínez et al. [2009], the authors point out the main problems and the main advantages of WAB and UWEM metrics. For instance, WAB performs tests to evaluate checkpoints, yet it is not precise in the way it determines the number of potential violations of each checkpoint. However, it specifies all three priorities' checkpoints. Concerning UWEM, this metric produces more precise results, although it only focuses on priority 1 and 2 checkpoints. Thus, these two metrics are merged into *WAB*$*$. Consequently, *WAB*$*$ has more precision in terms of the obtained results. In conclusion, this new metric considers 3 priority levels and has 36 checkpoints (25 WAB checkpoints + 11 UWEM checkpoints). This metric was tested by evaluating 30,600 web pages from banking sector websites. The results show that *WAB*$*$ outperforms WAB and UWEM.

Overall Accessibility Metric (OAM)

In 2005, Bailey and Burd proposed OAM. The calculated value considers the number of violations of a checkpoint and the weight of that checkpoint as the confidence level. This confidence level depends on how certain the checkpoint is. There are four confidence levels: certain checkpoints weigh 10, high certainty checkpoints weigh 8, low certainty checkpoints weigh 4 and the most uncertain checkpoints weigh 1. The higher the weight, the more the barrier is penalized.

This metric does not have a limited range of values. The higher this metric's score, the more inaccessible the web page is.

$$OAM = \sum_{c} \frac{B_c W_c}{N_{attributes} + N_{elements}} \tag{2.6}$$

Equation 2.6 presents the formula for computing the OAM metric, where B_c is the number of violations of checkpoint c, W_c is the weight of the checkpoint c, $N_{attributes}$ is the number of HTML attributes on a given web page, and $N_{elements}$ is the number of elements on a given web page.

Page Measure (PM)

Later, in 2007, Bailey and Burd proposed Page Measure (PM). This metric "analyzes the correlations between the accessibility of web sites and the policies adopted by software companies regarding usage of CMS or maintenance strategies" [Vigo and Brajnik, 2011]. It is similar to OAM (Overall Accessibility Metric), however, instead of using checkpoint weights, the checkpoint priority levels are considered. This metric does not have a limited range of values.

The higher this metric's score, the more inaccessible the web page is.

$$PM = \frac{\sum_c \frac{B_c}{priority_c}}{N_{attributes} + N_{elements}} \tag{2.7}$$

Equation 2.7 presents the formula for computing the PM metric, where B_c is the number of violations of checkpoint c, $priority_c$ identifies the priority level of the checkpoint c (1, 2 or 3), $N_{attributes}$ is the number of HTML attributes on a given web page, and $N_{elements}$ is the number of elements on a given web page.

SAMBA

Brajnik and Lomuscio proposed SAMBA, a semi-automatic method for measuring barriers of accessibility, that joins automatic evaluations with human judgement, and, for this reason, is a semi-automated methodology.

SAMBA was proposed by Brajnik and Lomuscio in 2007 and it is based on WCAG 1.0. This method applies human judgment in a context of the Barrier Walkthrough analysis [Brajnik and Lomuscio, 2007] to estimate aspects related to the automated tool errors and the severity of the barriers. The Barrier Walkthrough method is used for evaluating the web accessibility [Brajnik, 2009] and it is performed by experts. This manual approach contextualizes the accessibility barriers identified by experts within usage scenarios and these barriers receive a severity score. The severity score of a barrier assumes a value from $\{0, 1, 2, 3\}$ that corresponds to false positive (FP), minor, major or critical barriers.

This semi-automated approach [Brajnik and Lomuscio, 2007] applies a set of sequential steps. Initially, automatic accessibility tools are used to identify the potential accessibility barriers and the provided results are submitted to human judgement. Then, it is possible to statistically estimate the false positives and the severity of barriers for each website. Finally, barriers are grouped according to disability types and it is possible to derive scores that represent non-accessibility.

This metric computes two accessibility indexes: Raw Accessibility Index (AI_r) and Weighted Accessibility Index (AI_w). Since AI_w is based on confidence intervals manually computed by human experts, its result is represented by an interval $[\underline{AIw}, \overline{AIw}]$. The confidence intervals express the minimum and the maximum percentages of a type of barriers (FP, *minor*, *major* or *critical*) for a specific disability (blind users, deaf users, among others) on a given website. For example, having the interval $[6, 12]$ in column 'critical' and row 'blind' means that, in a given website, there are between 6% and 12% of critical barriers for blind users. The AI_w index considers weights that are associated with severity levels *minor* and *major*. If both *minor* and *major* weights are equal to 1, AI_w becomes unweighted (AI_u).

SAMBA has a limitation: it cannot cope with false negatives, i.e. problems that were not identified [Vigo and Brajnik, 2011]. This means that, although human judgments are used to evaluate and validate the results obtained by the automated tools, they do not deal with the problem of the false negatives, since the experts only verify the identified problems. For this reason, the actual issues that were not identified, are not going to be analyzed by the experts, i.e. the problems that are not identified by the evaluation tools are not considered.

$$AI_r = \prod_d (1 - F \cdot \overrightarrow{D}_d)^2 \tag{2.8}$$

$$\underline{AI_w} = \prod_d (1 - F \cdot min\{1, \overline{H_d}\})^2 \tag{2.9}$$

$$\overline{AI_w} = \prod_d (1 - F \cdot \underline{H_d})^2 \tag{2.10}$$

$$F = \frac{\text{number of potential barriers}}{\text{number of HTML lines}}, \tag{2.11}$$

$$\underline{H_d} = \frac{\underline{f}_{d,mnr}}{w_{mnr}} + \frac{\underline{f}_{d,maj}}{w_{maj}} + \underline{f}_{d,cri}, \tag{2.12}$$

$$\overline{H_d} = \frac{\overline{f}_{d,mnr}}{w_{mnr}} + \frac{\overline{f}_{d,maj}}{w_{maj}} + \overline{f}_{d,cri} \tag{2.13}$$

In equation 2.8, F is the barrier density of a website, d is a disability type, and D is the disability vector of a website. In equations 2.9 and 2.10, H_d is the severity of the barriers of a disability type d. Equations 2.12 and 2.13 identify f as the relative frequency, mnr as a minor barrier, maj as a major barrier, and cri as a critical barrier.

Web Accessibility Evaluation Metric (WAEM)

The Web Accessibility Evaluation Metric Based on Partial User Experience Order study [Song et al., 2017] was proposed by Song et al. and intends to analyze data from the user experience of people with disabilities. To do so, the authors defined a formula that calculates the weighted accessibility score (equation 2.15), by using the pass rate (equation 2.14), of a certain checkpoint on a website. Besides these formulas, this metric also considers users' experience evaluations through PUEXO pairs. PUEXO (Partial User EXperience Order) defines pairs of websites that establish a comparison in terms of user experience. For instance, the (a, b) pair indicates that a certain user had better browse experience in website a compared to website b. The PUEXO pairs are then compared to the weighted accessibility scores of the websites in question, by equation 2.16.

Subsequently, the results of equation 2.16 and the users' evaluations are both used to calculate the optimal checkpoint weights (equation 2.17). Equation 2.17 is not, however, adequate once the user experience is a subjective aspect. For this reason, the authors developed equation 2.18, where they make use of machine learning.

As seen in [Song et al., 2017], "results demonstrate that WAEM really can better match the accessibility evaluation results with the user experience of people with disabilities on Web accessibility". Nevertheless, the user experience is a subjective problem and varies according to the user. This means that it is complicated to confirm a relationship between user experience and web accessibility, since different users can have different user experiences [Song et al., 2017].

The higher the weighted accessibility score, the more accessible the website is.

$$p = \frac{s}{h} \tag{2.14}$$

Equation 2.14 presents the formula for computing the Pass Rate, where p is the pass rate of a checkpoint, s is the number of pages of a website a checkpoint passed, and h is the total number of web pages of a website.

$$q_i = P_i w = \sum_{j=1}^{m} P_{i,j} w_j \tag{2.15}$$

Equation 2.15 presents the formula for computing the Weighted Accessibility Score, where q_i is the weighted accessibility score of a website i, $P_{i,j}$ is the pass rate of a checkpoint j on a website i, m is the number of checkpoints, and w_j is the weight of a checkpoint j, according to its priority level.

$$f((a,b),w,P) = \begin{cases} 1 : P_a w > P_b w \\ 0 : otherwise \end{cases} \tag{2.16}$$

Equation 2.16 presents the formula for computing the function f, where (a, b) is a PUEXO pair that represents an order identified by disabled users, w is the set of checkpoints' weights, and P is the matrix of the pass rates of all websites.

$$
\begin{aligned}
argmax_w &= \sum_{i=1}^{k} f(L_i, w, P) \\
s.t. \sum_{j=1}^{m} w_j &= 1; \forall i, w_i > 0
\end{aligned}
\tag{2.17}
$$

Equation 2.17 presents the formula for computing the optimal checkpoint weight vector w, where w is the set of checkpoints' weights, L is the matrix that contains all pairs of websites, i is the website, j is the checkpoint, m is the number of checkpoints, and P is the matrix of pass rates.

$$
\begin{aligned}
argmin_w &= \sum_{i=1}^{k} e_i \\
s.t. \sum_{j=1}^{m} w_j &= 1; \forall, e_i \geq 0, w_i > 0, P_{L_{i,1}} w + e_i > P_{L_{i,2}} w
\end{aligned}
\tag{2.18}
$$

Equation 2.18 presents the formula for computing the optimal checkpoint weight vector w, where i is the website, e is the error tolerance vector, P is the matrix of pass rates, m is the number of checkpoints, and L is the matrix that contains all pairs of websites.

Reliability Aware Web Accessibility Experience Metric (RA-WAEM)

RA-WAEM is a metric that assesses the severity of accessibility barriers by considering the user experience of disabled people [Song et al., 2018]. The authors of this metric wanted to overcome the limitation of only using checkpoint weights, by reflecting the user experience of people with disabilities.

This metric's approach is similar to WAEM's approach. RA-WAEM is also aligned with PUEXO, which represents a pair of ordered websites, according to user experience. This metric also aims to calculate the optimal checkpoint weights as shown in equation 2.22. However, this last equation is not continuous. For this reason, equation 2.23 emerged. Yet, the fact that user experience is subjective and influenced by users' expertise level and objectivity, led to a reliability aware model (equation 2.24) where they introduce the reliability level. This new formula is the main difference between RA-WAEM and WAEM.

The results shown in the Song et al. [2018] article assert that RA-WAEM outperforms WAEM, since it is more stable and reliable concerning the user experience of disabled people.

One limitation of both RA-WAEM and WAEM metrics is that the users that are picked to evaluate the accessibility of a set of web pages, may not have a certain expertise level, ending up compromising the final metric results. For instance, users with low expertise would probably have more difficulty, considering a website as inaccessible [Song et al., 2018]. Whenever the experience of more volunteers is considered, the performance of both metrics decreases. Nevertheless, results indicate that RA-WAEM is significantly less affected than WAEM [Song et al., 2018].

The higher the weighted accessibility score, the more accessible the website is.

$$
p = \frac{s}{h} \tag{2.19}
$$

Equation 2.19 presents the formula for computing the Pass Rate, where p is the pass rate of a checkpoint, s is the number of pages of a website a checkpoint passed, and h is the total number of web pages of a website.

$$q_i = P_i w = \sum_{j=1}^{m} P_{i,j} w_j \tag{2.20}$$

Equation 2.20 presents the formula for computing the Weighted Accessibility Score, where q_i is the weighted accessibility score of a website i, $P_{i,j}$ is the pass rate of a checkpoint j on a website i, m is the number of checkpoints, and w_j is the weight of a checkpoint j, according to its priority level.

$$f((a,b),w,P) = \begin{cases} 1 : P_a w > P_b w \\ 0 : P_a w \leq P_b w \end{cases} \tag{2.21}$$

Equation 2.21 presents the formula for computing the function f, where (a,b) is a PUEXO pair that represents an order identified by disabled users, w is the set of checkpoints' weights, and P is the matrix of the pass rates of all websites.

$$\begin{aligned} argmax_w = \sum (a,b,u) \in L f(a,b,w,P) \\ s.t. \sum_{j=1}^{m} w_j = 1; \forall 1 \leq j \leq m, w_j > 0 \end{aligned} \tag{2.22}$$

Equation 2.22 presents the formula for computing the optimal checkpoint weight vector w, where w is the set of checkpoints' weights, L is the matrix that contains all pairs of websites a and b ordered by disabled user u, i is the website, j is the checkpoint, m is the number of checkpoints, and P is the matrix of pass rates.

$$\begin{aligned} argmin_w = \sum_{i=1}^{k} e_i \\ s.t. \sum_{j=1}^{m} w_j = 1; \forall 1 \leq j \leq m, w_j > 0; \\ \forall (a,b,u) \in L, e_{a,b} \geq 0, P_a w + e_{a,b} > P_b w \end{aligned} \tag{2.23}$$

Equation 2.23 presents the formula that corrects 2.22, since it is not continuous, where (a,b,u) is a tuple containing the PUEXO pair of websites a and b that were evaluated by the disabled user u, e is the error tolerance, P is the matrix of pass rates, m is the number of checkpoints, j is the checkpoint, w is the checkpoints' weights, and L is the matrix that contains all pairs of websites a and b ordered by disabled user u.

$$\begin{aligned} argmin_w = \sum (a,b,u) \in L e_{a,b} r_u \\ s.t. \sum_{j=1}^{m} w_j = 1; \forall 1 \leq j \leq m, w_j > 0; \\ \forall (a,b,u) \in L, e_{a,b} \geq 0, P_a w + e_{a,b} > P_b w \end{aligned} \tag{2.24}$$

Equation 2.24 presents the formula for computing the reliability aware model, where (a, b, u) is a tuple containing the PUEXO pair of websites a and b that were evaluated by the disabled user u, e is the error tolerance, r is the reliability level vector, P is the matrix of pass rates, m is the number of checkpoints, j is the checkpoint, w is the checkpoints' weights, and L is the matrix that contains all pairs of websites a and b ordered by disabled user u.

Barrier Impact Factor (BIF)

BIF is the barrier impact factor. According to Battistelli et al. [2011], this metric analyzes each accessibility error with respect to the way it affects disabled users' browsing by means of assistive technologies. It evaluates the accessibility, against the WCAG guidelines, using a list of assistive technologies or disabilities affected by each error. Each error represents a success criterion failure that was detected by the accessibility evaluation tool. It is necessary to define a barrier-error association table in advance, that represents a list of assistive technologies affected by each error.

The main goal is to understand the impact factor of each barrier on a specific assistive technology or disability (for example, a screen reader). The result score refers to the amount of detected errors that were identified for each assistive technology and it also considers the weight of that assistive technology. This weight's value varies according to the success criterion conformance level: level A errors weigh 3, level AA weigh 2 and level AAA weigh 1.

This metric's range of values is not defined. Nevertheless, the minimum score it can have is 0, which represents the absence of barriers. The higher this metric's score, the higher the impact of a certain barrier on a specific type of assistive technology/disability.

$$BIF(i) = \sum_{error} error(i) \times weight(i) \tag{2.25}$$

Equation 2.25 presents the formula for computing the BIF metric, where $BIF(i)$ is barrier impact factor of an assistive technology i, $error(i)$ is the number of detected errors that affect the assistive technology i, and $weight(i)$ is the weight of assistive technology i (1, 2 or 3).

Web Accessibility Quantitative Metric (WAQM)

WAQM was proposed by Vigo et al., and it overcomes some previous measures limitations (i.e. lack of score normalization and consideration of manual tests). It considers the WCAG guidelines classified according to the 4 principles: Perceivable, Operable, Understandable and Robust [Freire et al., 2008b]. This metric measures the conformance using percentages [Vigo et al., 2011], and it produces one score for each WCAG guideline in addition to an overall score. It considers the severity of checkpoint violations according to WCAG priorities and it provides normalized results.

Unlike other metrics, WAQM also takes into account the problems that are identified as warnings by the accessibility evaluation tools [Freire et al., 2008b]. It not only considers automatic tests but also manual tests.

According to Vigo, Arrue, Brajnik, Lomuscio, and Abascal [2007], this metric was proposed to overcome the drawbacks of the WAB and FR metrics as they do not focus on specific user groups, cover less guidelines and do not consider expert manual evaluation results.

This metric is based on the sum of failure rates for groups of checkpoints and these checkpoints are grouped according to their priority levels and their WCAG 2.0 principles (Perceivable, Operable, Understandable, Robust) [Freire et al., 2009]. The authors defined weights for each priority level: $W_1 = 0.8$, $W_2 = 0.16$ and $W_3 = 0.04$ for checkpoints with priorities 1, 2 and 3, respectively.

Since WAQM was considered to be tool dependent, there was the need to see if it was possible to prove the opposite [Vigo et al., 2007]. Therefore, the authors of the [Vigo et al., 2007] study wanted to have similar outcomes, regardless of the evaluation tool being used. For this matter, the authors proposed a method to reach independence of the tools for every possible scenario. A total of 1363 web pages from 15 websites were evaluated against the WCAG guidelines, using the automated evaluation tools EvalAccess and LIFT. They used 2 different tools to understand the behavior of the WAQM metric when the accessibility is measured by different tools. So, they tuned two WAQM parameters (a and b) to obtain independence. However, WAQM proved to be tool independent when conducting large scale accessibility evaluations with more than 1400 web pages [Vigo and Brajnik, 2011].

WAQM's normalized values range from 0 to 100, where the last one corresponds to the maximum accessibility level.

$$WAQM = \frac{1}{N} \sum_{x \in \{p,o,u,r\}} N_x \sum_{y \in \{e,w\}} \frac{N_{x,y} \sum_{z \in \{1,2,3\}} W_z A(x,y,z)}{N_x} \tag{2.26}$$

$$A(x,y,z) = \begin{cases} \frac{-100}{b} \frac{B_{x,y,z}}{P_{x,y,z}} + 100, if \frac{B_{x,y,z}}{P_{x,y,z}} < \frac{a-100}{a-100/b} \\ -a(\frac{B_{x,y,z}}{P_{x,y,z}}) + a, otherwise \end{cases} \tag{2.27}$$

Equations 2.26 and 2.27 present the formulas for computing the WAQM metric, where N is total number of checkpoints, N_x is the number of checkpoints from a specific principle x (x $\in$ {Perceivable, Operable, Understandable, Robust}), $N_{x,y}$ is the number of checkpoints from a principle x and type of test y (y $\in$ {automatic, manual)}, W_z is the weight of the checkpoint, according to its priority level z, $B_{x,y,z}$ is the number of accessibility errors of a checkpoint of priority level z, type of test y and principle x, $P_{x,y,z}$ is the number of test cases of a checkpoint of priority level z, type of test y and principle x, a is a variable that varies between 0 and 100, and b is a variable that varies between 0 and 1.

Navigability and Listenability

Fukuda et al. [2005] proposed two different web metrics. These metrics are responsible for evaluating the usability for blind users.

Navigability is responsible for evaluating the structure of the web page elements. It evaluates headings, intra-page links, labels, among other HTML elements of a certain web page. Listenability takes into consideration the alternative texts and denotes how properly built they are.

Each of these two metrics executes a set of calculations using the aDesigner (Accessibility Designer) engine. This approach is responsible for the visualization of the Web's usability for blind users through colors and graduations [Fukuda et al., 2005].

Web Interaction Environments (WIE)

Lopes and Carriço proposed, in 2008, a metric that quantifies Web accessibility [Lopes and Carriço,

2008]. It calculates the proportion of checkpoints that are violated on a web page [Vigo and Brajnik, 2011]. To do so, it considers a set of checkpoints and, for each of them, it verifies if a checkpoint c is successfully evaluated or if it fails [Lopes and Carriço, 2008]. If it is successfully evaluated, then $v_c = 1$, otherwise $v_c = 0$.

This metric's values have a limited range from 0 to 1, where 1 means the web page in question is totally accessible and all checkpoints that were evaluated in that web page have passed.

$$WIE(p) = \frac{\sum v_c}{n} \tag{2.28}$$

Equation 2.28 presents the formula for computing the WIE metric, where $WIE(p)$ is this metric's final score for a page p, v_c is a variable that assumes 1 if a checkpoint c passes, otherwise is 0, and n is the number of checkpoints.

Conservative, Strict and Optimistic

Conservative, Strict and Optimistic are the three web accessibility metrics defined in Lopes, Gomes, and Carriço [2010]. These metrics are based on the results of a checkpoint evaluation of an HTML element: PASS, FAIL or WARN. For each checkpoint, a PASS result indicates that an HTML document compliance is verified; a FAIL result specifies an HTML document compliance that is not verified; and a WARN result specifies it is impossible to verify the HTML document compliance. The main difference between these three metrics resides in the way they consider WARN results. They all contemplate the number of PASS results and the number of applicable elements to evaluate the accessibility results of an automatic accessibility evaluation tool. The conservative metric considers WARN results as failures, the optimistic metric considers them as passes, and the strict metric does not consider them at all.

These three metrics' scores range from 0 to 1, where 1 means the web page in question is totally accessible.

$$rate_{conservative} = \frac{passed + warned}{applicable} \tag{2.29}$$

$$rate_{optimistic} = \frac{passed}{applicable} \tag{2.30}$$

$$rate_{strict} = \frac{passed}{applicable - warned} \tag{2.31}$$

Equations 2.29, 2.30 and 2.31 present the formulas for computing the Conservative, Optimistic and Strict metrics, respectively, where $passed$ is the number of passes, $applicable$ is the number of applicable elements, $warned$ is the number of warnings.

eXaminator

According to Benavidez [2012], this metric classifies specific situations that can be positive or negative. eXaminator presents a quantitative index that measures the accessibility of a web page. It uses the WCAG 2.0 as a reference. It has two different modes to calculate the qualifications:

- Standard: eXaminator applies all tests. Some of the tests verify the errors, while others are responsible to qualify the correct situations.

- Strict: eXaminator applies only the set of most secure tests, i.e. the tests that have less possibilities of creating false positives or false negatives.

The author considers that not all tests have the same importance, i.e. they need different weights. This means that it is necessary to first weight the tests to make sure their relative weight reflects their differences from each other. The weight calculation is reflected in equation 2.32, and it is the multiplication between the Confidence of the test and the Value. Both Value and Confidence vary between 0 and 1, meaning that the weight will always be a value between 0 and 1. The Value variable depends on the WCAG conformance levels: level A: $V = 0.9$; level AA: $V = 0.5$; level AAA: $V = 0.1$. The Confidence variable verifies, for each test, what procedures are applicable when running it and, for each procedure that cannot be verified, the confidence decreases by 0.1.

eXaminator uses a matrix with information about each test, in particular the Element (E), Situation (S), Note (N), Tolerance (T) and Fraction (F). The Element identifies one or a set of HTML elements and the test is only applied if the element is present in the web page or if the element is *all*. The Situation identifies one or a set of HTML elements that fulfills a certain condition. The Note is the initial qualification of the test that was applied to the first detected situation. It is an absolute value that varies between 1 and 10, where 10 means the test classification result is excellent. The Tolerance is the error tolerance threshold, i.e., indicates the maximum number of errors that are allowed to happen in a specific situation. If the number of errors exceeds the Tolerance, the final test classification decreases by 1 point. Finally, the Fraction variable represents the quantity of errors that decrease the initial note by 1.

The final score of a web page is the ratio between the sum of all tests by the sum of their respective weights. This metric's result uses a scale from 1 to 10, where 1 represents a very bad accessibility level and 10 means otherwise.

$$P = C * V \tag{2.32}$$

Equation 2.32 presents the formula for computing the Test Weight, where P is the final weight score, C is the confidence of the test, and V is the value of the test.

Afterwards, there are three different tests that can be applied: True/False tests, Test of proportional type and Test of decreasing type [Benavidez, 2012].

$$R = N * P \tag{2.33}$$

Equation 2.33 presents the formula for computing the True/False tests, where R is the result of the test, N is the Note, and P is the weight of the test.

$$R = N * (1 - S/E) * P \tag{2.34}$$

Equation 2.34 presents the formula for computing the Proportional Type tests, where R is the result of the test, N is the Note, S is the Situation, E is the Element, and P is the test weight.

$$R = (N - (S - T)/F) * P \tag{2.35}$$

Equation 2.35 presents the formula for computing the Decreasing Type tests, where R is the result of the test, N is the Note, S is the Situation, T is the Tolerance, P is the test weight, and F is the Fraction.

Web Accessibility Barrier Severity (WABS)

Instead of being concerned about conformance to priority levels, this metric focuses on the barriers that limit the accessibility based on their severity. It ranks each web accessibility barrier found in all web pages of a dataset of websites [Abuaddous et al., 2017]. Each barrier has an associated numerical weight according to its severity and impact on the accessibility level. The authors define the barriers' weights as suggested in Vigo et al. [2007].

The final result of this metric represents a value for each barrier based on the priority class it violates and based on the web page that is being assessed, i.e. the final score will relate to a specific barrier in a certain web page of a website.

It was possible to emphasize two aspects when using this metric. First, each barrier is unique and has different properties, even if it belongs to the same priority level as other barriers, and, second, barriers that belong to priority level 1 are not necessarily more severe compared to the others.

This metric's formula covers three different measurements [Abuaddous et al., 2017]: (1) the importance of a barrier to the remaining barriers that violate the same priority level; (2) the importance that barrier has to the web page; (3) the importance of the barrier to all the remaining barriers in all websites.

The formula considers the frequency of a given barrier in a certain web page, the total number of barriers that violate the same priority level in a specific web page, the total number of web pages where a given barrier appears in, the total number of web pages and the total number of barriers that appear in a given web page.

The score result is normalized from 0 to 1. The closer the result is to 0, the less severe the barrier is.

$$WABS = \frac{\sqrt{\sum_{d=1}^{k} freq(bi)^2}}{\sqrt{\sum_{d=1}^{k} b(pc)^2}} * \frac{n(bi)}{N} * \frac{Pc}{\sqrt{\sum_{d=1}^{k} (b)^2}} \tag{2.36}$$

Equation 2.36 presents the formula for computing the WABS metric, where $freq(bi)$ is the frequency of a barrier bi, d is the web page that is being checked, k is the last web page to be tested, $b(pc)$ is the total number of barriers that violate the same priority level pc, $n(bi)$ is the number of web pages the barrier bi appears in, N is the total number of web pages, Pc is the weight of the priority level of a checkpoint, and b is the total number of barriers of that web page d.

Chapter 3

Large-scale accessibility evaluation

In order to address the objectives listed in section 1.2, in particular, to (1) understand the way technologies and categories impact the web accessibility, (2) compare the 11 accessibility metrics and (3) study the accessibility of the home pages in comparison with the whole website accessibility, we developed a large-scale accessibility evaluation structure. Therefore, it was possible to perform specific accessibility analyzes that would answer our main goals.

This chapter presents the development of a system to perform large-scale accessibility evaluation and technology identification. It will discuss the tools that are going to be used, in particular, to evaluate the web accessibility of a sample of web pages and to identify the technologies of these web pages' domains. Afterwards, the architecture used to run the accessibility evaluations and technology identifications is going to be presented. Finally, we explain some details about the implementation that started in March 2021, regarding how we collected the sample of web pages, as well as the pre-test we executed in order to identify and correct possible problems.

3.1 Accessibility evaluation

To run a study based on a large number of accessibility evaluations of web pages, the only viable option is to conduct automated accessibility assessments, due to the limited resources available. To run those evaluations we used QualWeb. QualWeb[1] is an automated web accessibility evaluation engine, developed by the Informatics Department of FCUL[2] (Faculdade de Ciências da Universidade de Lisboa). QualWeb performs a set of tests on a web page that check conformance with ACT-Rules[3] and WCAG 2.1 Techniques[4].

For each web page evaluated, we extracted from the QualWeb report the number of elements that passed, the number of elements that failed and the number of warnings for each test. We also collected information about the test being applicable or not to the web page. This information is useful since we want to consider only applicable tests when computing accessibility metrics. When an applicable test passes, it means that it has no failures nor warnings. If an applicable test returns no errors, but has at least one warning, the test outcome is "warning". If the test has at least one element that fails, the test

[1] http://qualweb.di.fc.ul.pt/evaluator/
[2] https://ciencias.ulisboa.pt/
[3] https://act-rules.github.io/rules/
[4] https://www.w3.org/WAI/WCAG21/Techniques/

fails.

As previously mentioned, QualWeb has two types of tests: ACT-rules tests, which test a web page against a set community approved checks; and WCAG techniques, which test a web page against the tool developer's interpretation of specific WCAG techniques. To ensure that only checks that correspond to consensual interpretation of the WCAG and increase the validity of the results, we only used the outcomes of the ACT-Rules tests in this study. We applied the 0.6.1 version of QualWeb, which tested a total of 72 ACT-Rules.

3.2 Analysis of web technologies identification tools

To identify the web technologies used to develop each website, we used the Wappalyzer[5] tool, since QualWeb already integrates and implements this tool. Nevertheless, there is also the need to apply another web technologies identification tool to guarantee more reliability in our results. Thus, it is possible to increase the degree of certainty about the web technologies used in the development of the domains. Consequently, we did a research to identify which tool could be chosen in addition to Wappalyzer.

We used SimilarWeb[6] to verify how many visits the web page from each tested tool had. Based on the overall amount of visits each tool's web page has, six tools, BuiltWith[7], W3Techs[8], Netcraft[9], WhatCMS[10] and SimilarTech[11] were considered since they are the most widely used tools, having a number of visits higher than 100,000, as presented in table 3.1. We discarded the Larger.io[12], the Shielder.it[13]

Table 3.1: Number of visits and duration of each visit of 10 tools

Tool	**Total number of visits**	**Duration of each visit**
BuiltWith	131.07K	3 minutes
W3Techs	126.13K	1 minute
Netcraft	175.51K	1 minute
SimilarTech	152.45K	2 minutes
WhatCMS	252.79K	2 minutes
Larger.io	N/A	N/A
WhatRuns	70.38K	58 seconds
Shielder.it	N/A	N/A
Isitwp	412.79K	1 minute

and the IsItWP[14] tools due to the following limitations: (1) Larger.io did not provide any information whenever a domain was inserted; (2) Shielder.it performs evaluation of products security and does not

[5] https://www.wappalyzer.com/
[6] https://www.similarweb.com/pt/
[7] https://builtwith.com/
[8] https://w3techs.com/sites
[9] https://sitereport.netcraft.com/
[10] https://whatcms.org/
[11] https://www.similartech.com/
[12] https://www.larger.io/
[13] https://www.shielder.it/
[14] https://www.isitwp.com/

identify web technologies; (3) IsItWP only identifies WordPress hosting, theme, plugins and other aspects concerning all websites that use WordPress.

The remaining tools were tested over a sample of 25 web pages to verify their coverage, regarding the technologies they could find. Table 3.2 presents the web pages sample as well as the technologies that were used to build the web pages. To identify which web pages we would consider for our sample and their respective technologies, we used the Hunter.io[15] platform that identifies a list of web pages that apply a given technology. Browser extensions, such as Wappalyzer extension, WhatRuns[16] and Library Detector, and code inspections were also considered in order to confirm that Hunter could correctly provide reliable results. The web technologies that were chosen were based on a study [WebAIM, 2021]

Table 3.2: List of 25 domains used to test the web technologies identification tools and list of actual technologies of each website

Websites	**Technologies**
https://wordpress.org/	Wordpress, jQuery
https://fightersmarket.com/	Shopify, Vue
http://squarespace.com/	Squarespace
https://pt.wix.com/	Wix, Zepto, React, LoDash
https://www.weebly.com/	Weebly, AdRoll, LoDash, jQueryUI, jQuery, Bootstrap, Mustache
https://www.joomla.org/	Joomla, jQuery
https://matomo.org/	Elementor, WordPress, jQueryUI, jQuery
https://www.drupal.org/	Drupal, AppNexus, jQuery
https://developers.googleblog.com/2018/03/transitioning-google-url-shortener.html	Blogger, jQuery
https://www.packtpub.com/	Magento, Underscore, jQueryUI, jQuery
https://mootools.net/	MooTools
https://www.linkedin.com/	React
https://dandomain.dk/	Angular
https://www.w3.org/	Backbone.js
https://www.flickr.com/	NodeJS, YUI, Moment.js
https://www.sav.com/	Codeigniter, React, jQuery, Bootstrap
https://dan.com/	Ruby on Rails, LoDash, React, jQuery, Bootstrap
https://www.booking.com/	Criteo, Underscore, jQuery
https://www.domainshop.com/main.php	Google AdSense, Bootstrap, jQuery
https://emberjs.com/	Ember
https://greensock.com/showcase/	jQuery, Underscore, Mustache, Hammer.js
https://beans.agency/	Modernizr, WordPress, jQuery
https://www.vinylc.com/en/main	RequireJS, jQuery
https://dojotoolkit.org/	Dojo
https://stackoverflow.com/	ASP.net, jQuery

that analyzes technologies. All technologies are listed in table 3.3.

Table 3.3: Technologies we wanted to search on the websites of our sample

CMS	**Javascript Frameworks**	**Javascript Libraries**	**Web Frameworks**	**Advertisement Networks**	**Programming Languages**
Squarespace	MooTools	YUI	Bootstrap	AdRoll	PHP
Wix	RequireJS	Zepto	CodeIgniter	AppNexus	Javascript
Weebly	React	LoDash	Ruby on Rails	Criteo	Python
Joomla	Backbone.js	Modernizr	Angular JS	Google AdSense	Ruby
Elementor	Mustache	Hammer.js	ASP.net	Doubleclick	Java
Drupal	Vue.js	jQuery	Drupal		
WordPress	Angular	Dojo			
Blogger	Ember	jQueryUI			
Shopify	Node.js	Moment.js			
Magento		Underscore.js			

[15] https://hunter.io/techlookup
[16] https://www.whatruns.com/

Having a sample of web pages and their actual technologies, we tested the six tools. During the tests, we could state that Zepto, Hammer.js and AdRoll are three technologies that none of the six tools could find. As such, we used the browser extensions mentioned before (Wappalyzer, WhatRuns and Library Detector) to guarantee that Hunter had incorrectly identified these technologies. Since the browser extensions and the tools could not find the three technologies, we discarded them.

Results indicate that BuiltWith has the best coverage percentage: 69,05%. Table 3.4 shows the coverage percentage of each of the six tools. However, BuiltWith could not find Magento, Backbone.js and NodeJS. Then, we decided to check if BuiltWith could actually find those three technologies using other domains. Table 3.5, shows the different domains used to cover the three technologies BuiltWith could not find from the starting sample. Interestingly, BuiltWith seemed to have difficulties finding NodeJS. As such, and since the next best coverage percentage is from SimilarTech, we verified if it could identify these three technologies. Fortunately, SimilarTech was able to correctly identify Magento, Backbone.js and NodeJS.

Table 3.4: Coverage of each tested tool

Tools	**Coverage (%)**
BuiltWith	69.05%
W3Techs	51.87%
Netcraft	36.15%
WhatCMS	20.23%
ReScan	32.86%
SimilarTech	60.88%

Table 3.5: Extra domains used to test if BuilWith could find the Backbone.js, Magento and NodeJS

Domain	**Technologies**
https://backbonejs.org/	Backbone.js
https://www.joelandsonfabrics.com/	Magento
https://medium.com/	NodeJS
https://www.trustpilot.com/	NodeJS
https://www.eklablog.com/	NodeJS
https://www.att.com/	NodeJS
https://www.vice.com/en	NodeJS

Even if BuiltWith had problems identifying NodeJS, we tried to integrate the BuilWith tool, due to its coverage. BuiltWith provides a free plan, however, this plan only counts the number of technologies by category, instead of identifying them. Through Puppeteer[17], we tried to access the BuiltWith web page with all technologies' results obtained after running the tool with an url. After all the data process, we ended up having a list of technologies of a given domain. However, BuiltWith only allowed up to three requests within a minute. The fourth request would return different results, as shown in table 3.6.

Since SimilarTech provides an API which only allows 10 evaluations per month, we tried to apply the same methodology with SimilarTech instead. Luckily, SimilarTech could answer all the requests with the same valid responses. Consequently, SimilarTech became the best option to be used along with Wappalyzer.

[17] `https://github.com/puppeteer/puppeteer`

Table 3.6: Results of 4 tests performed with BuiltWith

Execution Order	Domain	Result
1st	https://dojotoolkit.org/	Google Analytics, Foundation, Apple Mobile Web Clips Icon, Viewport Meta, IPhone / Mobile Compatible, Cloudflare, Dojo Toolkit, Cloudflare DNS, eNom DNS, Comodo SSL, LetsEncrypt, SSL by Default, Cloudflare SSL, Google Apps for Business, Cloudflare Hosting, Apache, Debian, IPv6, Content Delivery Network
2nd	https://dojotoolkit.org/	Google Analytics, Foundation, Apple Mobile Web Clips Icon, Viewport Meta, IPhone / Mobile Compatible, Cloudflare, Dojo Toolkit, Cloudflare DNS, eNom DNS, Comodo SSL, LetsEncrypt, SSL by Default, Cloudflare SSL, Google Apps for Business, Cloudflare Hosting, Apache, Debian, IPv6, Content Delivery Network
3rd	https://dojotoolkit.org/	Google Analytics, Foundation, Apple Mobile Web Clips Icon, Viewport Meta, IPhone / Mobile Compatible, Cloudflare, Dojo Toolkit, Cloudflare DNS, eNom DNS, Comodo SSL, LetsEncrypt, SSL by Default, Cloudflare SSL, Google Apps for Business, Cloudflare Hosting, Apache, Debian, IPv6, Content Delivery Network
4th	https://dojotoolkit.org/	Plausible Analytics, IXXO Cart, Sydney Theme, Renard, Strathcom Media

Before employing both tools, it was important to understand the analysis performed. If the technologies' results for a given website are the same in all its web pages, only one execution is necessary. For instance, we tested both tools using a home page (https://dojotoolkit.org/) and one of its interior pages (https://dojotoolkit.org/download/). We concluded that the results are identical, so we only need to perform the identifications once for each web domain, reducing the number of HTTP requests needed.

3.3 Data Model

The next step is the implementation of the data model. Before this phase, it was necessary to investigate all the existing models and its respective benefits and disadvantages to apply the most adequate approach.

According to Karee [2020], there are two types of data models: relational and NoSQL. The first one is a structure where data is organized into relations that represent tables. Tables have rows and columns. This type of model should be used (1) whenever the ACID properties[18] need to be met; (2) when there is the necessity to join multiple tables; (3) when the schema is already delineated; (4) when scalability is not wanted, i.e. this model is only vertically scalable, however, it is scalable to a certain point, and it becomes really expensive.

The second one, however, does not need a schema previously defined. Since it has a dynamic structure, the schema can change when needed. According to Schaefer [2021], the NoSQL model type is not optimized for data duplication and does not require to have ACID properties. However, this model is appropriate whenever it is necessary to perform simple queries and quick data analysis, due to its performance speed. Also, it should be considered when developers want to horizontally scale large volumes of data [MongoDB, 2021], i.e., data is spread into different servers, offering availability.

We need consistency in our analysis, plus we do not need horizontally scalability. Hence, the re-

[18] https://www.geeksforgeeks.org/acid-properties-in-dbms/

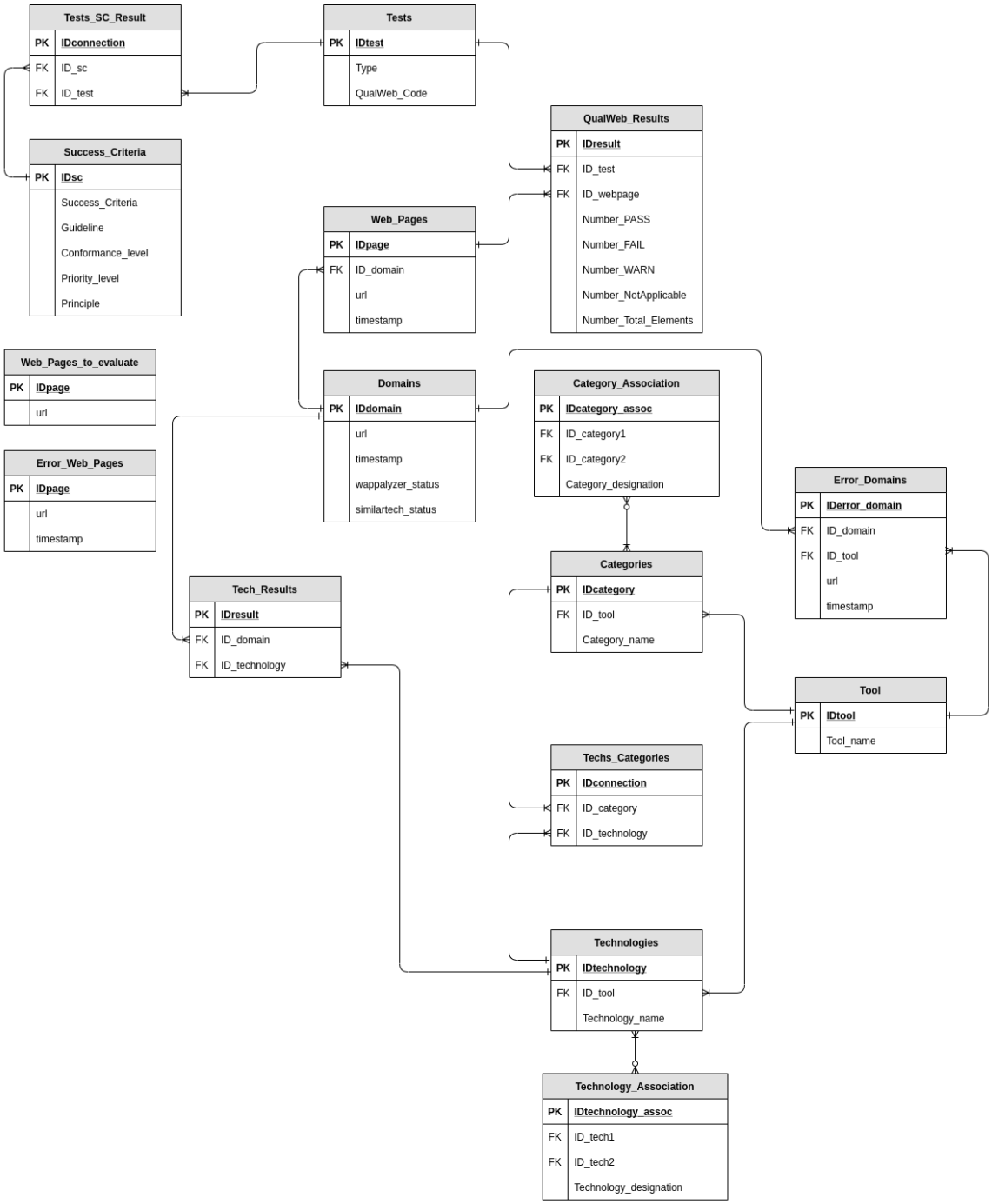

Figure 3.1: Data model

lational database is more appropriate. Also, the schema can and should be previously defined, before developing the architecture. The data model, with all discriminated tables and columns, is represented in figure 3.1.

3.3.1 Information flow

The flow of information of the data model represented in figure 3.1 begins with the insertion of urls into the Web_Pages_to_evaluate table. If this table is not empty, the evaluations proceed. When an url is chosen to be evaluated, its domain is extracted and it is inserted into Domains table, if it is not already there. Then, the evaluation with the QualWeb tool begins by evaluating the web page and, if it succeeds, the web page url is inserted into Web_Pages table.

Regarding the evaluation of an url by the QualWeb tool, the obtained results are organized into different parsed parameters that are prepared to be sent to the database. The report of each url has information that represents all tests performed on that web page. Qualweb_Results table will insert the results of each test as a single row. Each row has a test identifier that represents the test that was performed on the evaluation of a certain web page. Also, each test can have none, one or more related success criteria and this relation is represented in Tests_SC_Result table. The tests evaluate a set of HTML elements, so there is the need to count the elements per test. The number of elements that failed, that passed, and that are warnings of each test are three important pieces of data that are stored on each row. Also, there is a column that stores binary data: Number_NotApplicable. This column says whether the test was applicable, if the cell value is 0, or inapplicable, otherwise.

In the Wappalyzer and SimilarTech evaluations, the dynamics are equivalent. When running both these tools with one url domain, the technologies and the categories are obtained. The results of the technologies' identifications are stored into Tech_results table that only contains the identifiers of the domains and the technologies. Each technology is stored into the Technologies table with a unique identifier. This last table is going to contain all the technologies identified by both tools. The Domains table refers to the status of each domain evaluation. It provides a result status for each domain for Wappalyzer and SimilarTech tools. This status represents whether the domain url was ready to be evaluated, is being evaluated, its technologies were successfully identified or the tool could not identify its technologies, as to_evaluate, in_evaluation, success or fail, respectively.

Since we are using two different tools, they might identify the same technology and category, yet using different designations. For instance, Wappalyzer may identify a technology as jQuery while SimilarTech may identify the same technology as jquery. To point both different designations into a single one, it is necessary to manually identify and insert all the technologies and categories that relate with each other into the Technology_Association and Category_Association tables. After this manual procedure, both tables will have all the final categories and technologies' designations.

Whenever QualWeb, Wappalyzer and SimilarTech cannot evaluate a certain web page or domain, the respective url is inserted into Error_Web_Pages and Error_Domains tables, respectively.

The information flow of the system is represented in figure 3.2. This flowchart helped understanding the information flow as well as the way each functionality would be implemented.

3.4 System Architecture

To implement the information flow, we consider a reliable and quick solution: containers, more specifically Docker containers. A container is a "standardized unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another"

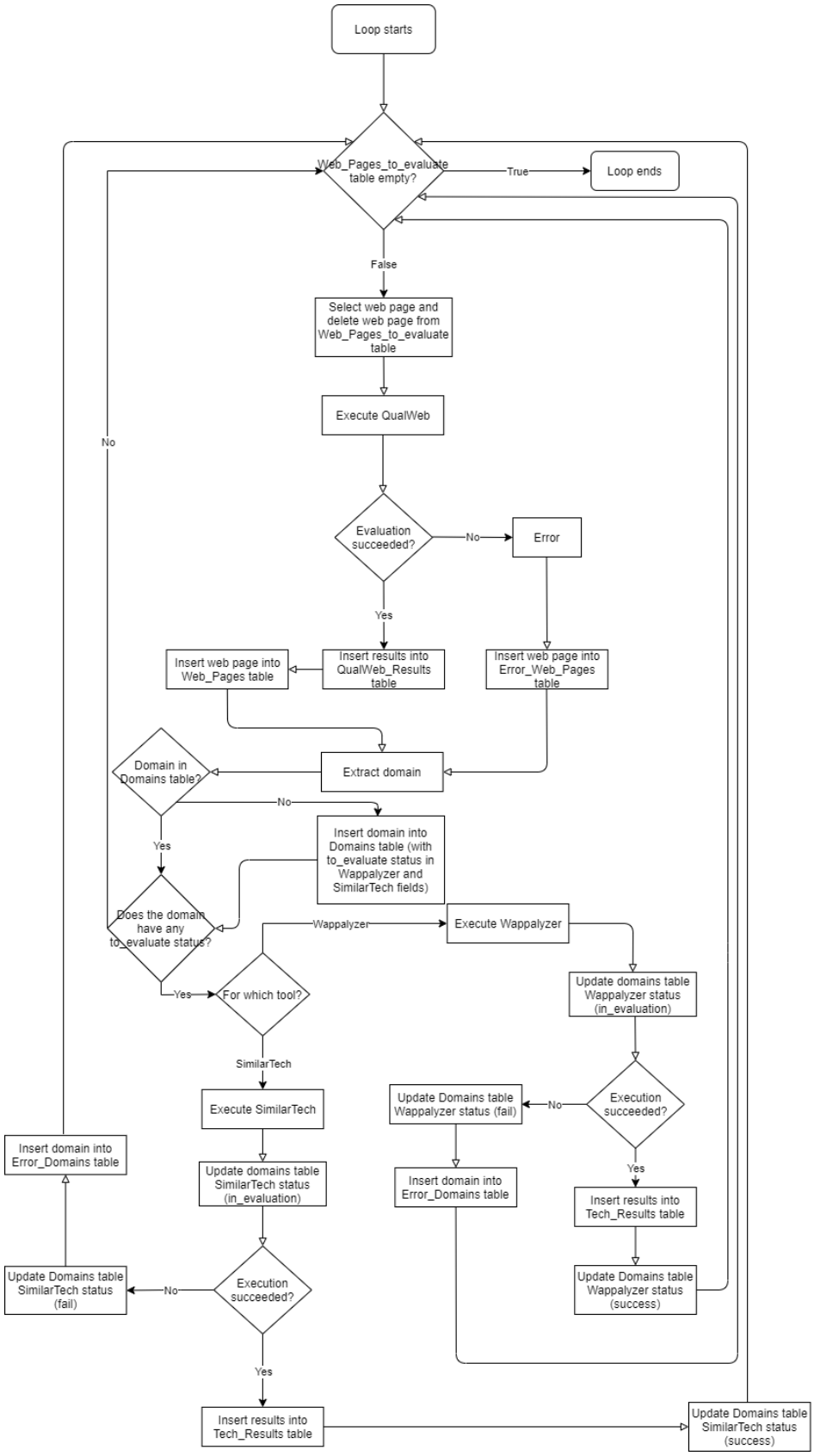

Figure 3.2: Flow of the information in our architecture

Table 3.7: Popularity percentages of container tools

Container tool	**Popularity Percentage**
Docker	57%
Kubernetes	48%
AWS ECS/EKS	44%
Azure Container Service	28%
Docker Enterprise	27%
RedHat openShift	24&
Docker Swarm	21%
Google Container Engine (GKE)	15%
Pivotal Cloud Foundry	13%
Mesosphere	10%

[Docker, 2021]. Docker was chosen due to its popularity and available documentation and also because it is the most worldwide used technology to containerize applications among businesses, according to Bayern [2019]. Table 3.7 shows the usage percentages of each container tool.

The architecture of the large-scale system is represented in figure 3.3. On the client side, there are two Docker containers running the SimilarTech and Wappalyzer tools and three QualWeb containers that will run the accessibility evaluations at the same time. The QualWeb containers were chosen according to the machine's hardware specifications. Since we did identify the technologies of the domains of all web pages, less requests were needed. Thus, and in order to balance the resources' usage with the QualWeb parallel evaluations, we decided to run only one container per web technology identification tool. QualWeb containers make a request to the server to obtain an array of urls, while the Wappalyzer and SimilarTech containers request only one url from the PostgreSQL database and, after each execution, they will send the parsed information to the database. PostgreSQL was the technology used to build our database since we were more familiar with it.

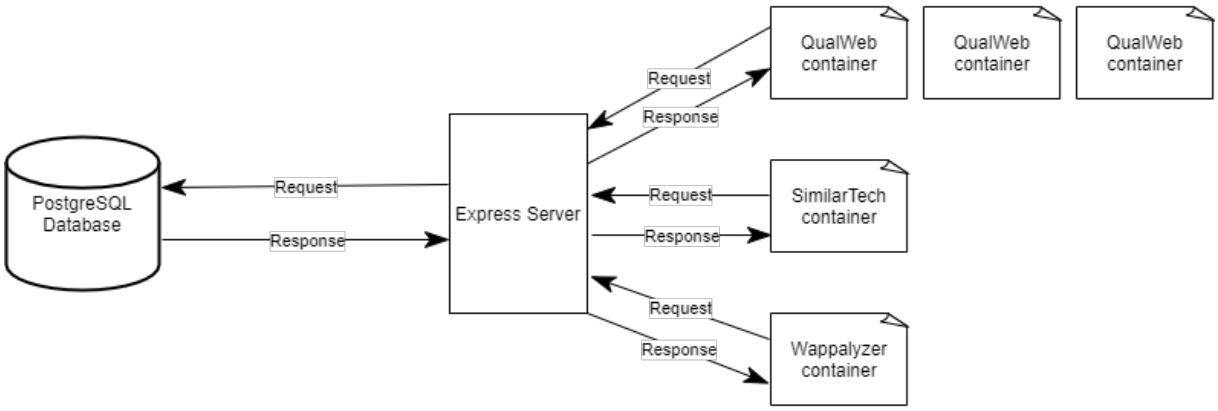

Figure 3.3: System architecture that runs the accessibility evaluations and the technologies identifications

3.5 Implementation and pre-test

Before the implementation and the pre-test, we had to choose from where we would obtain our data sample of web pages. Since CommonCrawl[19] provides crawled data that can be accessed and analyzed by everyone, we decided to use this platform to extract the urls for our dataset. We considered the crawled data from November 2020 to December 2020. The available data includes the crawl dates and the urls. The first step is to extract the urls and their respective crawl date. Then, all the data is processed by removing duplicates as well as removing unwanted urls (for example, robots.txt). The crawled dates are important since the Docker containers require a web page by its crawl date, so they evaluate the most recent crawled web pages.

The pre-test helped us understand certain aspects like how many QualWeb containers can run in the remote machine, given its hardware specifications. We first tried 8 containers and started reducing this number in order to avoid out of memory errors. The pre-test was carried out in the period from February 2021 to March 2021. We started to run 13 thousand web pages to verify if the accessibility evaluations and the technologies' identifications were correctly implemented. In addition to that, we could improve the error handling. For instance, the QualWeb application was modified so that it can have a 2-minute timeout. This timeout will allow the tool to continue the evaluations if it cannot evaluate a certain url, avoiding being suspended waiting for a response. This means that, if a web page has a pop-up or for some other reason it prevents QualWeb from performing the accessibility evaluation, this web page will be inserted into the errors table after 2 minutes.

After the pre-test, we began the large-scale evaluation in March 2021 and stopped the containers in September 2021. During the evaluation, we could select groups of web pages from the CommonCrawl dataset, taking into account their Top-Level Domain (TLD), to allow further analysis about the accessibility of web pages in each country. The most important TLDs we have considered are represented in table 3.8.

[19] `https://commoncrawl.org/`

Table 3.8: Examples of TLD that have a number of web pages that allows us to perform more representative analyzes

Top-level domain	**Description**
.org	Organization
.com	Commercial
.net	Network
.info	Information
.gov	Government
.eu	European Union
.news	News
.edu	Education
.es	Spain
.pt	Portugal
.fr	France
.us	United States
.uk	United Kingdom
.ca	Canada
.it	Italy
.br	Brazil

Chapter 4

Analysis of the large-scale evaluation results

After obtaining the large-scale evaluation results, it is necessary to process and interpret them so we can gather conclusions about accessibility aspects that will help us analyzing the impact of the web technologies in the web accessibility.

The results of the accessibility assessment analysis are going to be presented as well as the results of the web technologies impact in web accessibility. We will discuss the findings afterwards.

4.1 Results

In this section we present the results of the large-scale accessibility evaluation analysis of 2,884,498 web pages from 166,311 websites, averaging 21 pages per website. We first introduce general findings from the tools that were used to perform the large-scale evaluation. Then, we analyze the accessibility and web technologies results, based on the evaluated web pages and websites.

4.1.1 Dimension of the sample

The percentage of web pages and domains that were evaluated was determined in order to understand each tool's progress.

QualWeb could evaluate 2,884,498 pages which corresponds to a percentage of 81% of all web pages that were to be evaluated. This means that 19% of them could not be evaluated for some reason, and so they were inserted into the errors' table.

The Wappalyzer tool could evaluate 132,733 domains with success, out of all the 166,311 domains. This tool also presents an evaluated domains percentage of around 80%, equivalent to QualWeb.

SimilarTech, however, states a lower percentage of the domains that were evaluated (67%), that is equivalent to 110,670 domains. Nevertheless, this number indicates that this tool could evaluate the majority of the domains.

4.1.2 Web accessibility

The accessibility evaluation found a total of 86,644,426 errors, averaging 30 errors per page and 521 errors per website. The highest number of errors on a single web page was 15,645 and the lowest was

0. The highest number of errors on a website was 878,776 and the lowest 0. The total number of encountered errors (x-axis) is demonstrated in graphs 4.1 and 4.2, by the percentage of web pages and the percentage of domains (y-axis), respectively. According to figure 4.1, approximately 95% of the web pages, which corresponds to the majority, have 0 to 100 accessibility errors. The same happens to the domains, represented in figure 4.2, where approximately 93% of the domains present 0 to 2,000 accessibility errors.

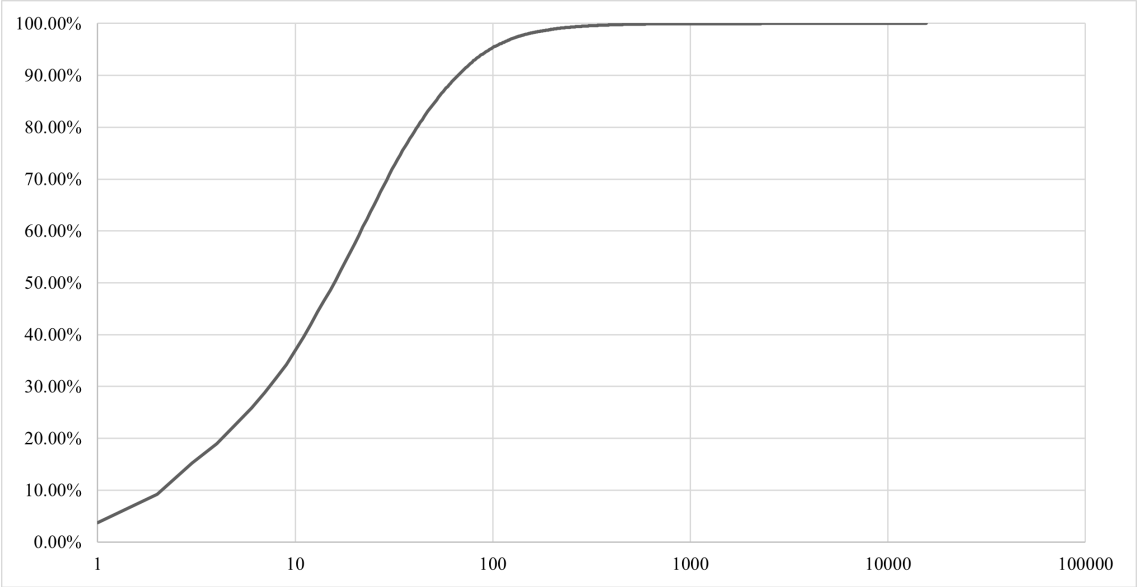

Figure 4.1: Percentage of web pages and respective maximum of errors (logarithmic scale)

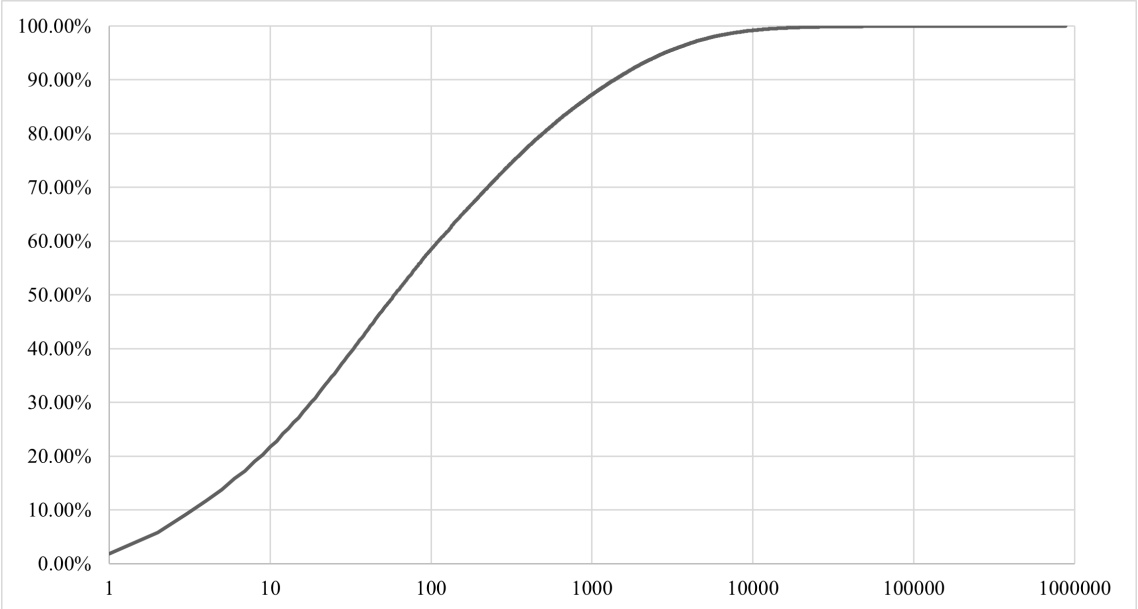

Figure 4.2: Percentage of web domains and respective maximum of errors (logarithmic scale)

Table 4.1: Number of errors by page of some top-level domains

Top-level Domain	Number of errors by page	Number of pages
gov	16.278	130,689
edu	17.614	93,956
us	24.211	106,432
org	25.197	154,154
uk	26.935	122,543
it	27.651	113,536
au	28.895	174,371
eu	29.052	102,995
info	31.837	125,233
com	31.864	166,388
de	32.174	109,135
net	33.915	102,671
pt	35.078	139,807
fr	35.336	113,513
es	35.859	120,085
asia	36.907	322,208
news	39.954	105,196
br	40.231	100,353

Top-level domains

Regarding top-level domains (TLD), we considered web pages from certain countries and other categories, as shown in table 3.8. Each TLD has, on average, 115,373 web pages and 324,510.96 errors. We managed to guarantee that certain TLDs had more than 90 thousand web pages, so our analysis could be more representative, as shown in table 4.1. Table 4.1 shows the number of errors by web page of each TLD as well as the number of pages. By comparing the Top-Level Domains represented in table 4.1, it is possible to verify that web pages from Brazil have the highest number of errors by page, revealing an average of 40 errors on each page. In contrast, .gov domains exhibit approximately 16 errors by page.

ACT-Rules and success criteria

The ACT-Rules that present the highest number of errors are ACT-R76 (Text has enhanced contrast), ACT-R37 (Text has minimum contrast) and ACT-R12 (Link has accessible name), as presented in table 4.2. This table also presents the percentages of the web pages in which accessibility issues were identified in each test. The complete table is represented in the appendixes.

In this study there are 79% of the web pages that had accessibility issues regarding the contrast of the text between the foreground and the background colors. For instance, a gray text in a white background may result into an accessibility failure since the contrast is not evident, and so, users with visual impairments may have difficulties reading the text. The ACT-Rule R76 refers to the enhanced contrast, i.e. each character meets the enhanced contrast requirement (at least 7:1), while the R37 refers to the minimum contrast, i.e. each character meets the minimal contrast requirement (at least 4.5:1). The third rule that was more frequently violated (in 52% of the web pages) concerns links and their accessible name. In case there is a link in a web page, there is the need to specify some accessible

Table 4.2: 10 ACT-Rule with the highest number of accessibility errors

ACT-Rule	Description	Number of errors	Percentage of web pages
ACT-Rule R76	Text has enhanced contrast	33,109,298	79%
ACT-Rule R37	Text has minimum contrast	18,322,594	66%
ACT-Rule R12	Link has accessible name	9,026,352	52%
ACT-Rule R18	id attribute value is unique	6,569,778	31%
ACT-Rule R17	Image has accessible name	6,274,421	30%
ACT-Rule R16	Form control has accessible name	1,878,778	22%
ACT-Rule R19	iframe element has accessible name	1,207,887	19%
ACT-Rule R48	Element marked as decorative is not exposed	792,267	7%
ACT-Rule R14	meta viewport does not prevent zoom	658,061	22%

text through HTML attributes so users can know the main content of the url. The same happens with ACT-Rule R17, ACT-Rule R16 and ACT-Rule R19, but this time they regard the web images, form fields and iframe elements, respectively. All these HTML elements need to have associated attributes with accessible names or accessible text in their content. Blind users need to use screen readers to visualize the web content and, if these HTML elements have accessible names, users can understand their matter. Having 32% of the web pages where ACT-Rule R18 failed, we can conclude that more than the majority of the web pages had HTML elements with unique identifications. For example, having two HTML elements with the same id attribute ”myDiv”. Interestingly, the rule that refers to hidden HTML elements that are marked as decorative (in 7% of the web pages) produced more errors than the rule that refers to the possibility for the user to zoom (in 22% of the web pages), yet it was violated in a significant less number of web pages.

Table 4.3 shows the number of errors of each success criteria. The success criterion with the highest number of errors is 1.4.6, presenting a total of 33,109,298 errors, followed by 1.4.3 with 18,322,594 errors. These criteria refer to contrast issues. In contrast, with 262 errors, the success criterion 1.3.4 presents the lowest number of accessibility errors, not counting those success criteria with no accessibility errors. As can be verified in table 4.3, success criteria 2.4.4 and 2.4.9 present the same number of errors. Both success criteria are mapped to ACT-R12 and ACT-R44, which explains the same number of errors. Although the 2.4.9 success criterion is also mapped to ACT-R9, this test only throws warnings that were not considered in this analysis.

4.1.3 Web technologies

Wappalyzer and SimilarTech could identify 3482 different technologies from 166 categories. On average, 27 technologies were identified on each web domain and 31 technologies were identified by each category. Wappalyzer could identify 1197 different technologies, whereas SimilarTech could identify 2733 technologies. Although the number of technologies SimilarTech could identify was two times more compared to Wappalyzer, most of the identified technologies are not interesting for this study, as they do not directly communicate with the information that is presented to Web users. For instance, the Databases category includes web technologies like Oracle, MongoDB, among others, that aim at storing several types of data from the web content.

Table 4.3: 10 success criteria with the highest number of accessibility errors

Success Criteria	Description	Number of errors
1.4.6	Contrast (Enhanced)	33,109,298
1.4.3	Contrast (Minimum)	18,322,594
4.1.2	Name, Role, Value	13,585,522
2.4.4	Link Purpose (In Context)	9,026,352
2.4.9	Link Purpose (Link Only)	9,026,352
1.1.1	Non-text Content	6,934,650
4.1.1	Parsing	6,569,778
1.3.1	Info and Relationships	2,529,133
1.4.4	Resize text	658,061
1.4.10	Reflow	658,061

Table 4.4: Web technologies ordered by the number of times they were identified

Web technology	Number of identifications
jQuery	188,054
Google Analytics	114,015
Apache	111,302
PHP	108,462
WordPress	85,826
Bootstrap	63,001
Modernizr	46,521
jQuery UI	40,803
Asp .Net	9,731
React	8,434
Drupal	6,603

With respect to the number of times each web technology is identified, jQuery is the most frequently identified technology, followed by Google Analytics, Apache and PHP that appeared around 111 thousand times, on average, in the evaluated domains. Table 4.4 presents some technologies ordered by the number of times they were identified in all the 166,311 domains.

The proportion of accessibility errors by the number of pages of some interesting web technology categories is represented in figure 4.3. Through the representation of the proportion of errors by each category, it is possible to see that Advertising category that contains technologies such as DoubleClick, Google AdSense, Facebook Advertiser, among others, has the highest proportion of accessibility errors (36.1). It stands out from the remaining categories that vary between 31 and 22 errors by page. The Accessibility category demonstrates the lowest number of accessibility failures by the number of pages where this category was identified in (22.4 errors by page). Perhaps the reason that justifies this category's rank is the fact that its technologies' main focus resides in dealing with accessibility issues.

4.1.4 Web technologies impact in web accessibility

In order to analyze what web technologies and categories lead to worse or better accessibility levels, we used an accessibility metric for two reasons: (1) since all the web accessibility metrics intend to calculate

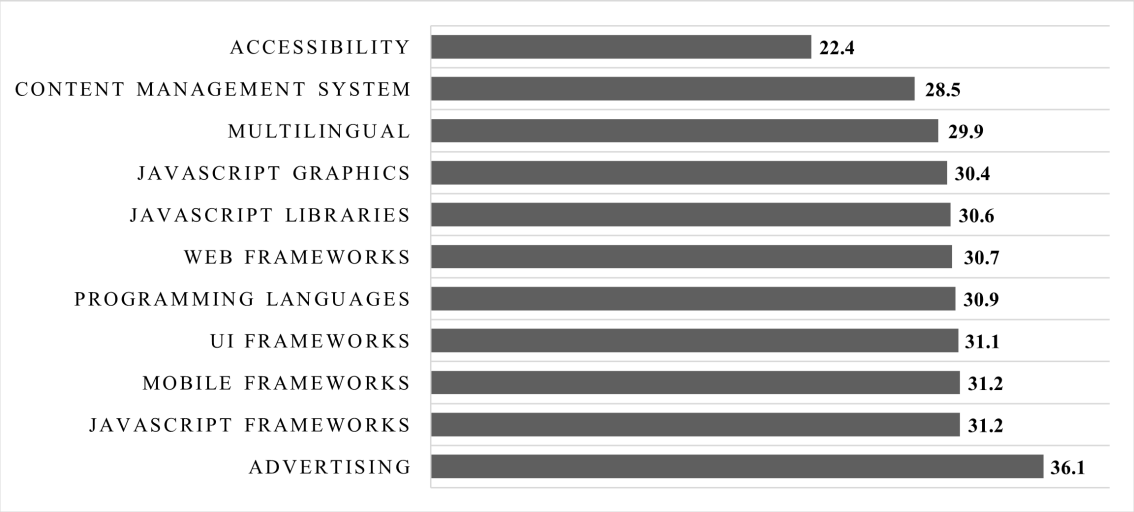

Figure 4.3: Proportion of the number of errors by the number of web pages of each category

a score that represents an accessibility level, it becomes a more accurate procedure than just calculating the number of errors; (2) and, as one of the main objectives of this study is to compare some existing metrics, we have detailed information about each metric's behavior, so the selection is justified. We considered the A3 metric due to the metric validity experiment results described in section 5.1.3, that shows that this metric is more discriminative.

Accessibility impact of different categories

We carried out two experiments. The first experiment intends to study the impact each web category has in web accessibility. For this purpose, this experiment considers the identification of 34 categories on our web pages sample. These 34 categories were chosen as they were considered to be the most interesting in this web accessibility impact study. Having two data samples containing (1) the accessibility scores of all pages that were not developed using technologies from a certain category and (2) the accessibility scores of web pages that use technologies from that category, and since our samples are not normally distributed, we applied the Mann-Whitney U rank test[1] to compare both samples. Thus, the impact that each category has in web accessibility is reached. The Mann-Whitney test provides a p value that indicates whether the category influences the accessibility. If the significance level (ρ-value) is higher than 0,0015 (0,05/34 categories), the category has no influence in the web accessibility, otherwise, the average of both samples should be analyzed to conclude whether it is a positive or a negative impact.

As we considered 34 different categories, we applied 34 tests. Each test varies in terms of the sample that contains the scores of the web pages that utilize one of the 34 categories in its development. The results of the Mann-Whitney test are presented in table 4.5. They indicate that all categories impact the accessibility of the evaluated web pages, except for Website Builder, Mobile Frameworks and Photo Galleries categories that have a ρ-value higher than 0,0015. Regarding the categories that impact the web accessibility, and according to the difference of the average of the web pages' A3 scores with and without each category, the Accessibility, Content Management Systems, LMS, Rich Text Editors, Static

[1] https://en.wikipedia.org/wiki/Mann%E2%80%93Whitney_U_test

Site Generator and Wikis categories present a notable positive impact in web accessibility as they all improve the accessibility level of the web pages. Conversely, Advertising, Comment System, Editors, LiveChat, MessageBoards, Social Logins and Forum Software categories showed an A3 score difference higher than 0.050 whenever web pages were built with technologies of these categories.

Nevertheless, and although the LMS category would be interesting as it shows a significant decrease in the accessibility level of the evaluated web pages, this category only has 2 web technologies: Moodle and uPortal. From all the 4,130 web pages that use LMS, only one page applies uPortal. The remaining 4,129 web pages use Moodle. As such, we could not compare both technologies through a Mann-Whitney test, since uPortal is only identified in one page. Hence, we decided to compare the A3 metric average of Moodle ($\mu = 0.366$) technology with the A3 metric score of uPortal (0.855). Results indicate that web pages with Moodle revealed a significant positive difference in the accessibility scores than those with uPortal.

Accessibility impact of technologies within the same category

The second experiment evaluates the impact of each web technology in its category. To perform this analysis, we selected some categories that we considered to be the most interesting to study the impact of their technologies, from those categories that manifested a statistically significant difference to web accessibility, i.e. web categories with ρ-value below 0.0015. Then, we selected all the technologies of the categories that were identified in no less than 2% of the web pages of their category. We applied this criterion, since some categories have a large amount of technologies. Thus, it is possible to filter the technologies that were most frequently identified. For the reason that the Advertising category had 43 technologies that were identified in more than 2% of the web pages, we decided to tune this percentage in order to analyze a smaller number of technologies in this category. When using a minimum threshold of 8%, we can obtain 6 advertising technologies. Consequently, we ended up having 13 categories and their respective web technologies that were identified in at least 2% (or 8%, in case of the Advertising category) of the web pages of their category. We considered the following categories: Maps, Comment Systems, Programming Languages, Video Players, Content Management Systems, Web Frameworks, JavaScript Libraries, JavaScript Frameworks, UI Frameworks, Accessibility, Wikis, JavaScript Graphics and Advertising.

This experiment relied on the Kruskall-Wallis test[2] to verify whether the technologies impact the category where they belong to. To perform this test, we had to guarantee the samples were independent from each other. In this regard, we did not consider those web pages which more than one technology of the same category was identified. Tables 4.6, 4.7 and 4.8 show the results of this experiment. As it is represented in the results tables, all web technologies that were tested in this experiment show to have an impact in web accessibility, as the ρ-value obtained from the Kruskal-Wallis test is lower than 0.0038 (0.05/13 tests). For each technology, the average of its A3 metric scores were calculated in order to understand which technologies would lead to higher or lower accessibility levels. Nevertheless, it is necessary to apply a post-hoc test, with the intention of verifying which technologies, of each category, have a statistically significant difference. The post-hoc test will perform pairwise comparisons of the web technologies in each category.

[2] https://en.wikipedia.org/wiki/Kruskal%E2%80%93Wallis_one-way_analysis_of_variance

Table 4.5: Mann-Whitney tests to analyze the impact of the web categories in the web accessibility

Category	**ρ value**	**Average of A3 without category**	**Average of A3 with category**	**Number of Pages with category**
Accessibility	<0.001	0.666	0.526	8,505
Advertising	<0.001	0.640	0.700	1,240,981
Analytics	<0.001	0.648	0.675	2,052,025
CMS	<0.001	0.711	0.627	1,776,338
Comment System	<0.001	0.664	0.731	120,676
Editors	<0.001	0.664	0.773	31,723
JavaScript Frameworks	<0.001	0.666	0.663	2,324,247
JavaScript Graphics	<0.001	0.666	0.645	80,260
JavaScript Libraries	<0.001	0.686	0.661	2,626,325
LiveChat	<0.001	0.664	0.720	146,116
LMS	<0.001	0.666	0.366	4,130
Maps	<0.001	0.664	0.701	163,485
Message Boards	<0.001	0.665	0.764	22,734
Mobile Frameworks	0.005	0.666	0.662	34,159
PaaS	<0.001	0.670	0.622	1,449,356
Page Builders	<0.001	0.668	0.627	162,592
Photo Galleries	0.052	0.666	0.668	26,176
Programming Languages	<0.001	0.688	0.655	1,928,355
Rich Text Editors	<0.001	0.668	0.473	38,861
Security	<0.001	0.662	0.681	540,850
Social Logins	<0.001	0.655	0.742	338,538
Static Site Generator	<0.001	0.666	0.559	3,076
UI Frameworks	<0.001	0.672	0.655	1,072,937
Video Players	<0.001	0.661	0.695	376,306
Wikis	<0.001	0.666	0.371	6,662
Web Frameworks	<0.001	0.666	0.660	261,150
Audio Video Media	<0.001	0.663	0.672	910,612
Captcha	<0.001	0.663	0.682	436,015
Forum Software	<0.001	0.665	0.873	2,323
Multilingual	<0.001	0.669	0.632	255,044
Online Forms	<0.001	0.666	0.662	122,336
Online Video Platform	<0.001	0.664	0.672	746,105
Web Hosting	<0.001	0.665	0.701	51,350
Website Builder	0.839	0.666	0.684	2,254

The Dunn's test[3] used in this study, is a post-hoc non parametric test that performs pairwise comparisons. Once it is applied, we obtain a significance level (ρ-value), which indicates whether the two web technologies that are being compared, present a statistically significant difference. If the ρ-value is less than 0.05, then the two technologies are statistically significantly different, and so, their A3 metric scores' average must be compared to understand which technology affects the category the most. Figures 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 4.10, 4.11, 4.12, 4.13, 4.14, 4.15 and 4.16 specify the boxplots of the A3 metric scores for all web technologies for the Accessibility, Advertising, Content Management Systems, Comment Systems, JavaScript Frameworks, JavaScript Graphics, JavaScript Libraries, Maps, Programming Languages, UI Frameworks, Video Players, Web Frameworks and Wikis categories, respectively.

[3] https://www.statisticshowto.com/dunns-test/

Table 4.6: Kruskal-Wallis test to analyze the impact of the web technologies in their categories

Categories	**Technologies (Number of pages with technology)**	**p-value**	**Average of the A3 scores for technology**
Accessibility	AccessiBe (909)	<0.001	0.610
	AudioEye (2,072)		0.315
	EqualWeb (295)		0.940
	UserWay (4,908)		0.570
Advertising	AppNexus (102,685)	<0.001	0.718
	DoubleClick (830,058)		0.652
	Facebook Advertiser (310,209)		0.755
	Google AdSense (443,042)		0.717
	Google AdWords Advertiser (173,995)		0.752
	Twitter Ads (154,270)		0.669
Comment Systems	Disqus (33,555)	<0.001	0.806
	Facebook Comments (57,819)		0.752
	LiveFyre (31,864)		0.241
Content Management Systems	Drupal (126,866)	<0.001	0.586
	Elementor (61,427)		0.789
	Jimdo (28,341)		0.855
	Joomla (80,764)		0.566
	TYPO3 (37,902)		0.633
	Wix (77,181)		0.610
	WordPress (1,302,963)		0.624

The Accessibility category results of the Dunn's test are represented in table D.1. All its web technologies present a ρ-value smaller than 0.001, which indicates that all pairs of technologies are statistically different. Hence, we can compare the four web technologies' averages, in order to understand which web technology impacts the most. EqualWeb's average (μ = 0.940) induces a decrease of the accessibility level, whereas AudioEye (μ = 0.315) reports the most accessible web content.

In the Advertising category, it is possible to verify pairs of web technologies that share a ρ-value lower than 0.05. For instance, AppNexus with DoubleClick, Google AdWords Advertiser and Google AdSense; DoubleClick with Google AdWords Advertiser, Google AdSense and Twitter Ads; Facebook Advertiser with Google AdWords Advertiser; Google AdWords Advertiser with Google AdSense and Twitter Ads; and Google AdSense with Twitter Ads. These ρ-value scores indicate that there are statistical significantly differences between the technologies of each pair. Based on the results, DoubleClick (μ = 0.652) presents the lowest average compared to AppNexus (μ = 0.718), Google AdWords Advertiser (μ = 0.752), Google AdSense (μ = 0.717) and Twitter Ads (μ = 0.669). In contrast, with an A3 metric scores' average of 0.755, Google AdWords Advertiser leads to better accessibility scores compared to Google AdSense and Twitter Ads. The results are shown in table D.2.

Regarding the Content Management Systems (CMS) results presented in table D.3, Joomla (μ = 0.566) leads to better accessibility scores than the remaining CMS, followed by Drupal (μ = 0.586) that has a better accessibility scores' average compared to Elementor, Jimdo, TYPO3 and WordPress. Jimdo (μ = 0.855), however, is identified in more inaccessible web pages when comparing with Drupal, Joomla, TYPO3, Wix and WordPress.

In table D.4, the results concerning the Comment Systems indicate better accessibility levels when

Table 4.7: Continuation of Kruskal-Wallis test to analyze the impact of the web technologies in their categories

Categories	**Technologies (Number of pages with technology)**	**p-value**	**Average of the A3 scores for technology**
JavaScript Frameworks	AMP (26,473)	<0.001	0.616
	Angular JS (21,709)		0.730
	Backbone.js (38,627)		0.703
	GSAP (109,116)		0.648
	Handlebars (36,006)		0.719
	MooTools (48,986)		0.583
	Mustache JS (51,118)		0.809
	Prototype (31,768)		0.725
	React (167,677)		0.613
	Require JS (36,818)		0.679
	Stimulus (16,394)		0.635
	Vue JS (55,722)		0.636
JavaScript Graphics	ChartJS (48,296)	<0.001	0.612
	D3 (4,909)		0.675
	Highcharts (2,303)		0.455
	MathJax (2,892)		0.543
	particlesJS (9,336)		0.562
	Raphael (10,358)		0.777
	Supersized (1,973)		0.718
	threeJS (2,328)		0.810
JavaScript Libraries	Hammer.js (71,599)	<0.001	0.645
	Isotope (208,397)		0.328
	jQuery Migrate (1,031,833)		0.766
	jQuery UI (635,742)		0.628
	jQuery (2,265,370)		0.472
	LightBox (162,919)		0.689
	Lodash (424,009)		0.743
	Modernizr (555,621)		0.646
	Moment.js (103,047)		0.738
	Polyfill (88,426)		0.769
	prettyPhoto (124,110)		0.708
	Slick (154,677)		0.378
Maps	Google Maps (128,993)	<0.001	0.712
	Leaflet (32,459)		0.636
	Mapbox GL JS (7,130)		0.707

web pages use LiveFyre (μ = 0.241) with a significant difference from the remaining web technologies accessibility scores. Disqus (μ = 0.806) and Facebook Comments (μ = 0.752) present closer and more inaccessible results. Nevertheless, Disqus reports the highest A3 metric average.

From all the JavaScript Frameworks represented in table D.5, all the pairs of technologies are statistically different, with the exception of AMP and Stimulus pair. In terms of the average of the accessibility scores, MooTools (μ = 0.583) represents the web technology that was identified in web pages with better accessibility scores. On the other hand, MustacheJS (μ = 0.809) is identified in more inaccessible content.

In JavaScript Graphics category represented in table D.6, Highcharts (μ = 0.455) leads to significantly more accessible web content compared to Chart.hs, D3, MathJax, Raphael, particles.js and three.js. In contrast to this, three.js (μ = 0.810) proved to be the web technology that was present in more inaccessible

Table 4.8: Continuation of Kruskal-Wallis test to analyze the impact of the web technologies in their categories

Categories	Technologies (Number of pages with technology)	p-value	Average of the A3 scores for technology
Programming Languages	Java (67,395)	<0.001	0.778
	Lua (28,849)		0.763
	NodeJS (25,117)		0.547
	PHP (1,780,370)		0.650
	Python (43,765)		0.816
	Ruby (26,616)		0.630
UI Frameworks	animate.css (168,733)	<0.001	0.669
	Bootstrap (953,689)		0.654
	ZURB Foundation (58,886)		0.641
Video Players	JW Player (8,042)	<0.001	0.730
	MediaElement (139,861)		0.693
	Plyr (8,787)		0.714
	VideoJS (33,948)		0.741
	Vimeo (56,347)		0.618
	YouTube (200,636)		0.689
Web Frameworks	CodeIgniter (10,015)	<0.001	0.709
	Django (9,614)		0.711
	Express (6,787)		0.677
	Laravel (11,407)		0.688
	Microsoft ASP.NET (158,078)		0.657
	Ruby On Rails (30,275)		0.607
	Symfony (10,960)		0.699
Wikis	Atlassian Confluence (830)	<0.001	0.805
	DokuWiki (916)		0.377
	Foswiki (255)		0.622
	MediaWiki (3,996)		0.180
	MoinMoin (664)		0.876

web pages.

With regards to Dunn's test results of JavaScript Libraries category, represented in table D.7, it is possible to assert that jQuery Migrate is not statistically different from any of the remaining web technologies, as the ρ-value is always higher than 0.05, except for Isotope and Slick. Isotope (μ = 0.328) is identified in more accessible web pages, compared to Hammer.js, jQueryUI, jQuery, LightBox, Lodash, Modernizr, Moment.js, Polyfill and prettyPhoto. In contrast, web pages with worse accessibility scores, include Polyfill (μ = 0.769) in their development, which means this web technology may cause more inaccessible content compared to Hammer.js, Isotope, jQueryUI, jQuery, LightBox, Lodash, Modernizr and prettyPhoto.

Table D.8 represents the Maps category. In this category, there is only one pair of technologies that is not statistically different. From the three web technologies of this category, Leaflet (μ = 0.636) is the most accessible technology compared to Google Maps (μ = 0.712) and Mapbox GL JS (μ = 0.707).

In the Programming Languages category pairs of technologies (table D.9), NodeJS (μ = 0.547) represents better accessibility scores compared to the remaining programming languages, whereas Python (μ = 0.816) presents the worst A3 metric average.

All web technologies from UI Frameworks category are statistically different, as represented in table D.10. Although they all have similar A3 metric averages, ZURB Foundation (μ = 0.641) presents more

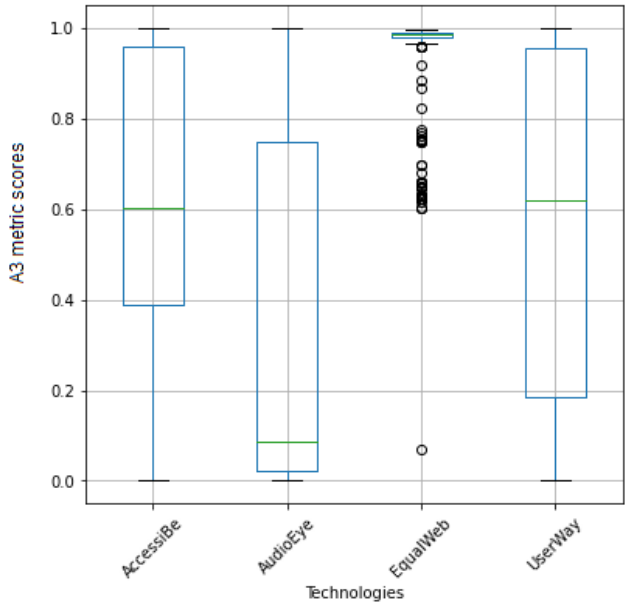

Figure 4.4: Boxplot of the A3 metric scores for each technology of Accessibility category

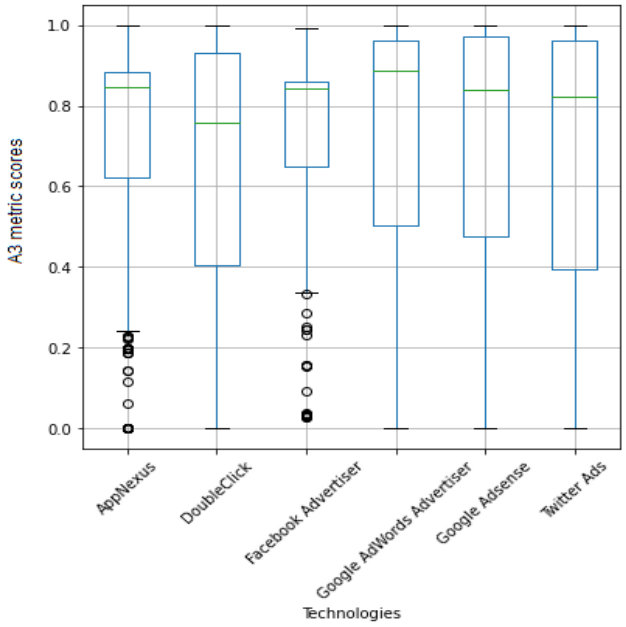

Figure 4.5: Boxplot of the A3 metric scores for each technology of Advertising category

accessible content.

Table D.11 represents all the pairwise comparisons for Video Players category, where VideoJS (μ = 0.741) and JW Player (μ = 0.730) are the two main technologies that lead to worse accessibility scores, compared to MediaElement.js (μ = 0.693), Plyr (μ = 0.714), Vimeo (μ = 0.618) and YouTube (μ = 0.689). Web pages that use Vimeo (μ = 0.618) tend to have more accessible results.

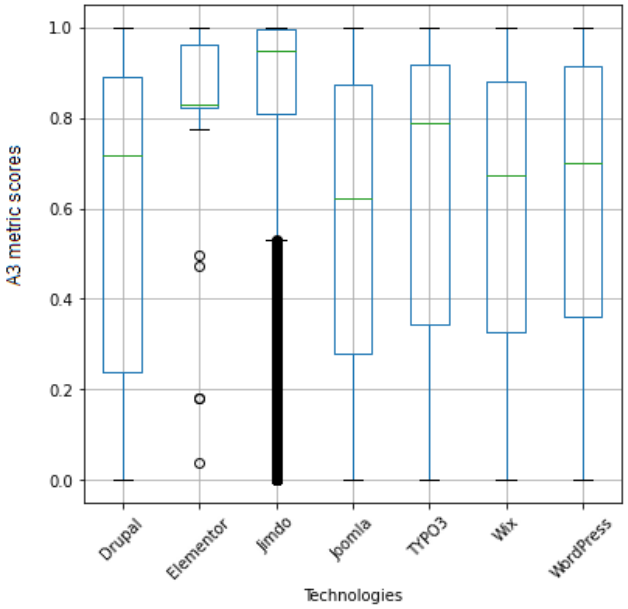

Figure 4.6: Boxplot of the A3 metric scores for each technology of Content Management Systems category

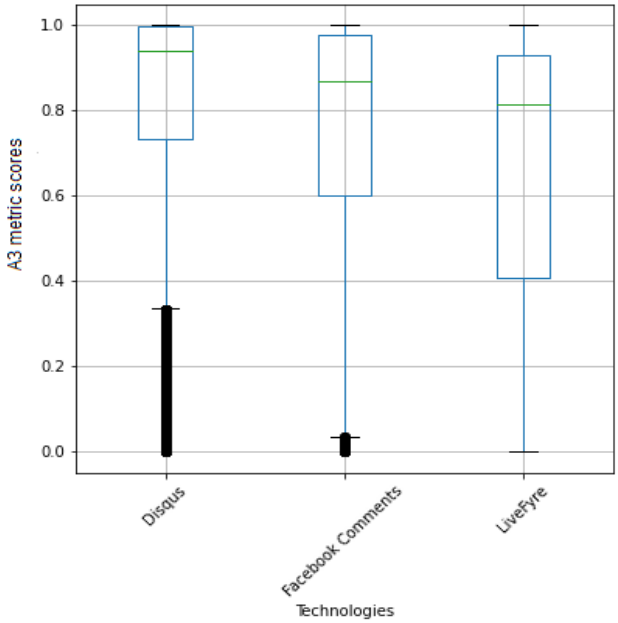

Figure 4.7: Boxplot of the A3 metric scores for each technology of Comment Systems category

According to table D.12, not all pairs of Web Frameworks technologies were considered, due to the fact that the ρ-value is higher than the significance level (0.05). For instance, the pairs of technologies CodeIgniter with Django, Ruby On Rails and Symfony; Django with Laravel, Microsoft ASP .NET and Ruby On Rails; Express with Laravel and Ruby On Rails; Laravel with Microsoft ASP .NET, Ruby On Rails and Symfony; Microsoft ASP .NET with Ruby On Rails and Symfony; and Ruby On Rails with

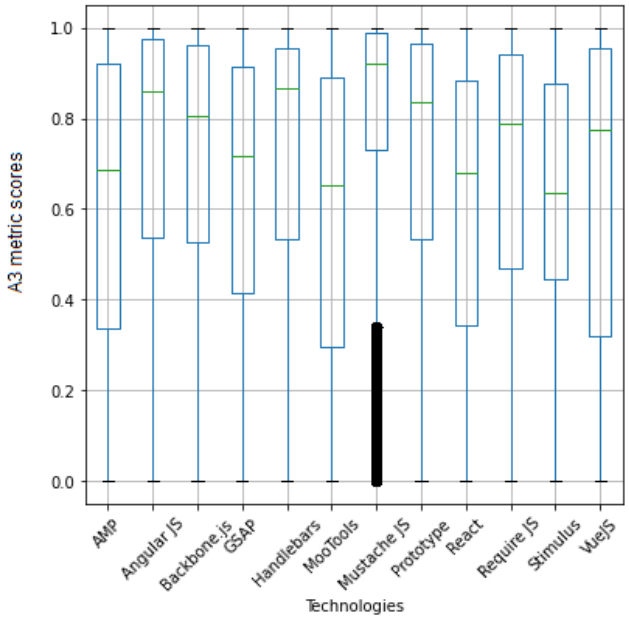

Figure 4.8: Boxplot of the A3 metric scores for each technology of JavaScript Frameworks category

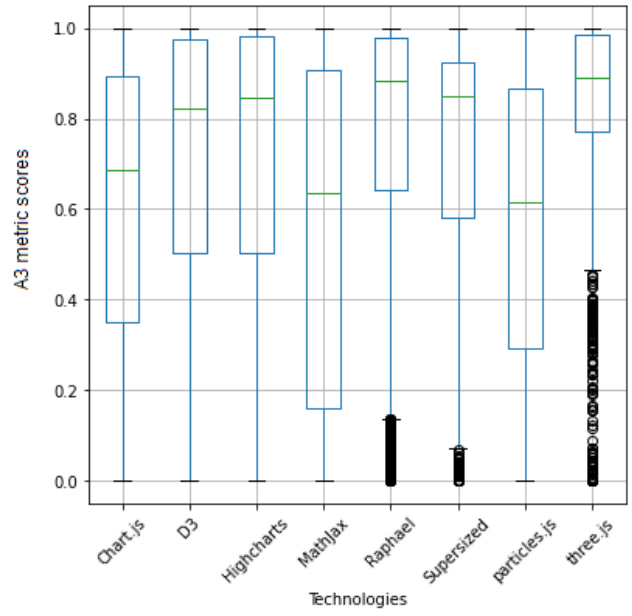

Figure 4.9: Boxplot of the A3 metric scores for each technology of JavaScript Graphics category

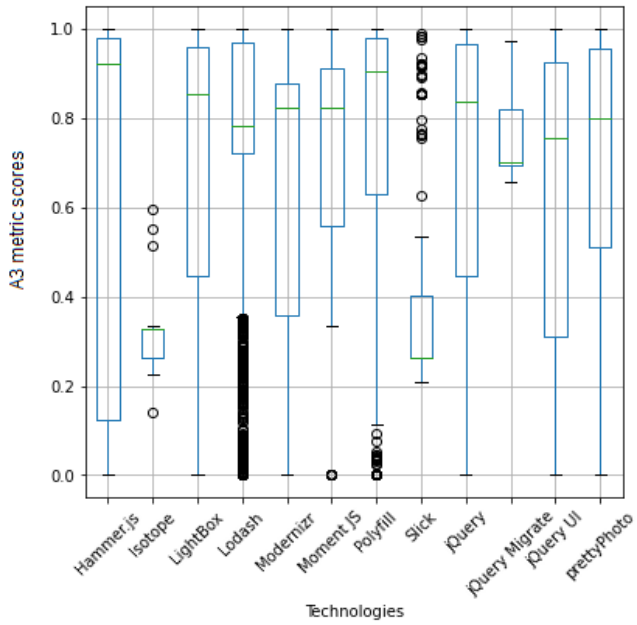

Figure 4.10: Boxplot of the A3 metric scores for each technology of JavaScript Libraries category

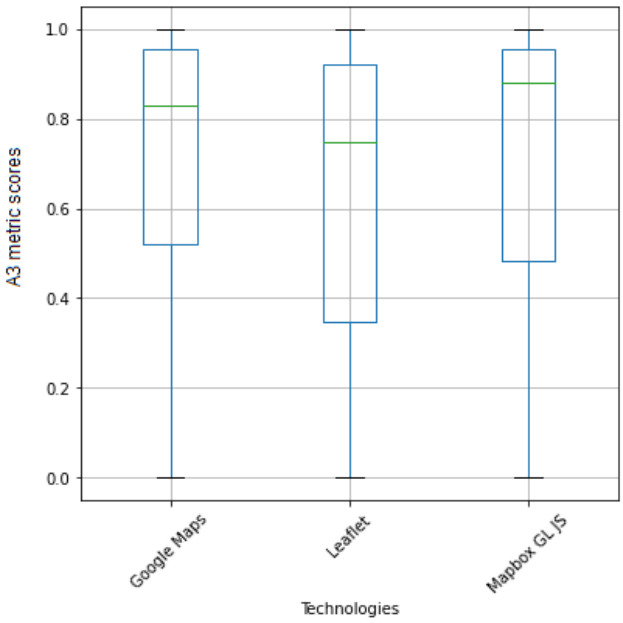

Figure 4.11: Boxplot of the A3 metric scores for each technology of Maps category

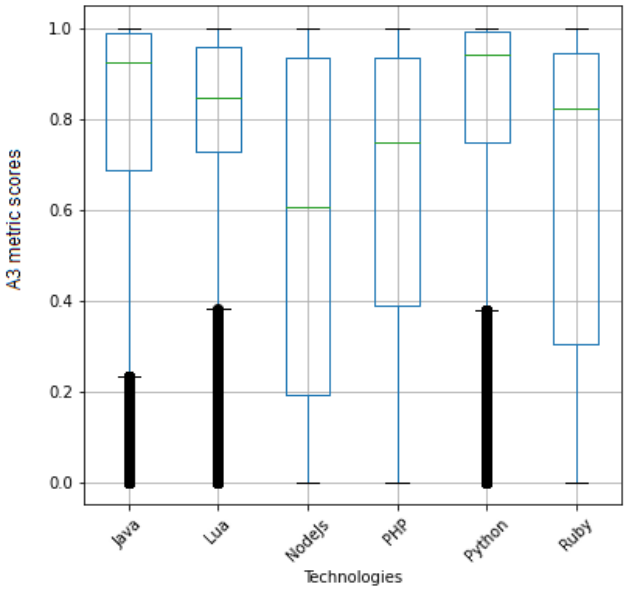

Figure 4.12: Boxplot of the A3 metric scores for each technology of Programming Languages category

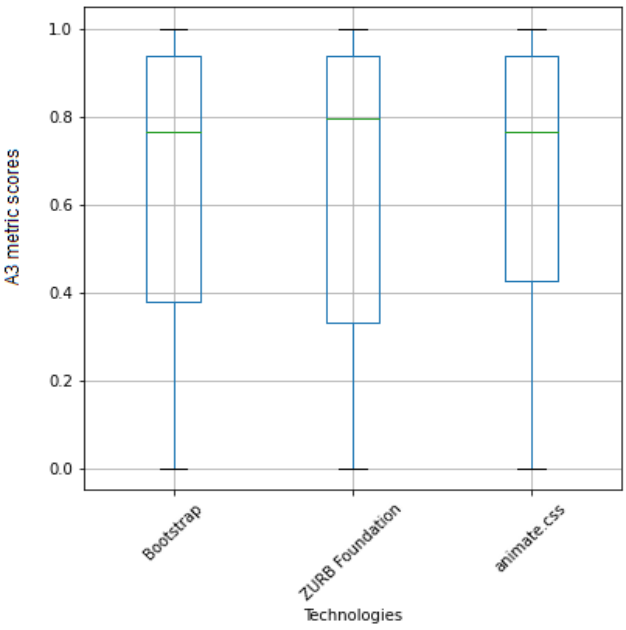

Figure 4.13: Boxplot of the A3 metric scores for each technology of UI Frameworks category

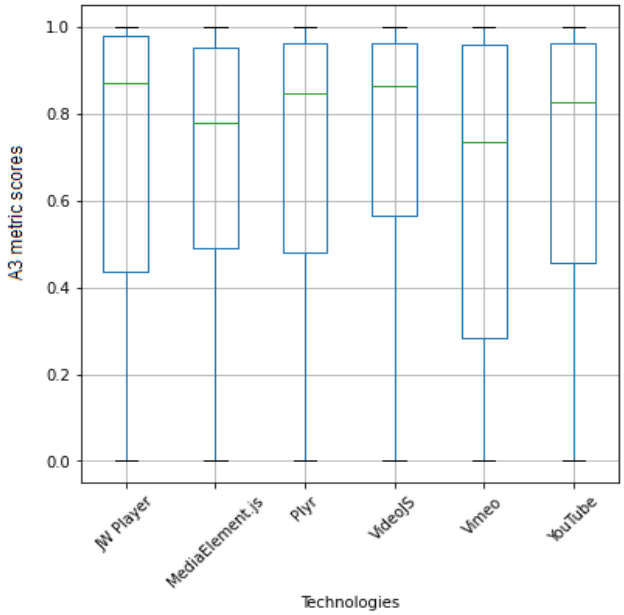

Figure 4.14: Boxplot of the A3 metric scores for each technology of Video Players category

Symfony are statistically different. Ruby on Rails (μ = 0.607) presents the lowest A3 scores' average, compared to any other technology. It shares a difference of 0.050 with Microsoft ASP.NET (μ = 0.657), which presents the second lowest A3 scores' average compared to Django (μ = 0.711), Laravel (μ = 0.688) and Symfony (μ = 0.699). Django claims to have worse accessibility scores, in web pages where this technology was identified in, compared to CodeIgniter, Laravel, Microsoft ASP .NET and Ruby on Rails.

From all web technologies from the Wikis category, represented in table D.13, MediaWiki (μ = 0.180) is, by far, the technology that leads to more accessible web content. Its average represents accessible content, as the closer the A3 metric score is to 0, the more accessible the web content is. By contrast, MoinMoin (μ = 0.876) represents the technology with worse accessibility scores of the evaluated web pages, having a difference of 0.696 from MediaWiki.

To better visualize some of the obtained results of each technology in the second experiment, figure 4.17 represents a diagram that helps comparing the technologies of all categories that were identified in more than 1 million pages. For each category, it is possible to state the technologies that had the best and the worst A3 metric scores in both ends of the straight orange line, and compare them with the metric's average for all pages (0.6657). The blue rhombuses represent the other technologies of the corresponding category, ordered by their A3 metric scores.

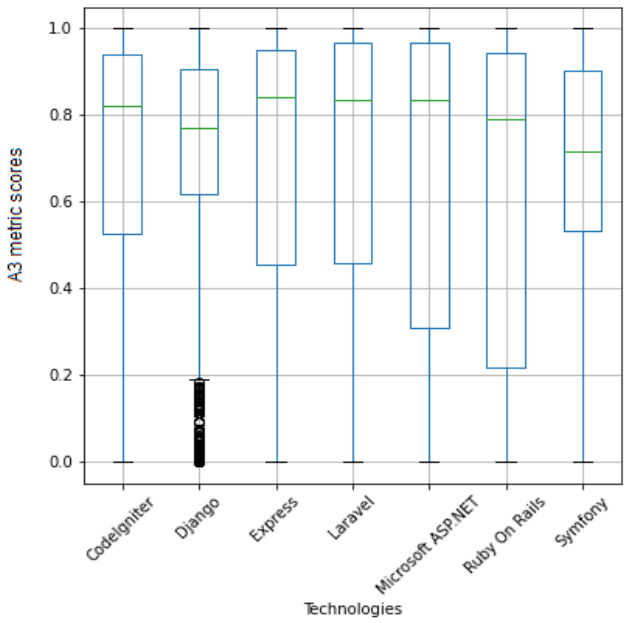

Figure 4.15: Boxplot of the A3 metric scores for each technology of Web Frameworks category

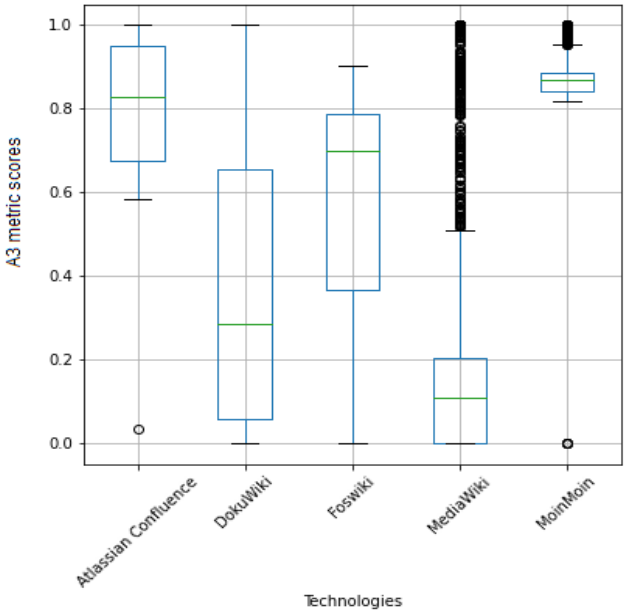

Figure 4.16: Boxplot of the A3 metric scores for each technology of Wikis category

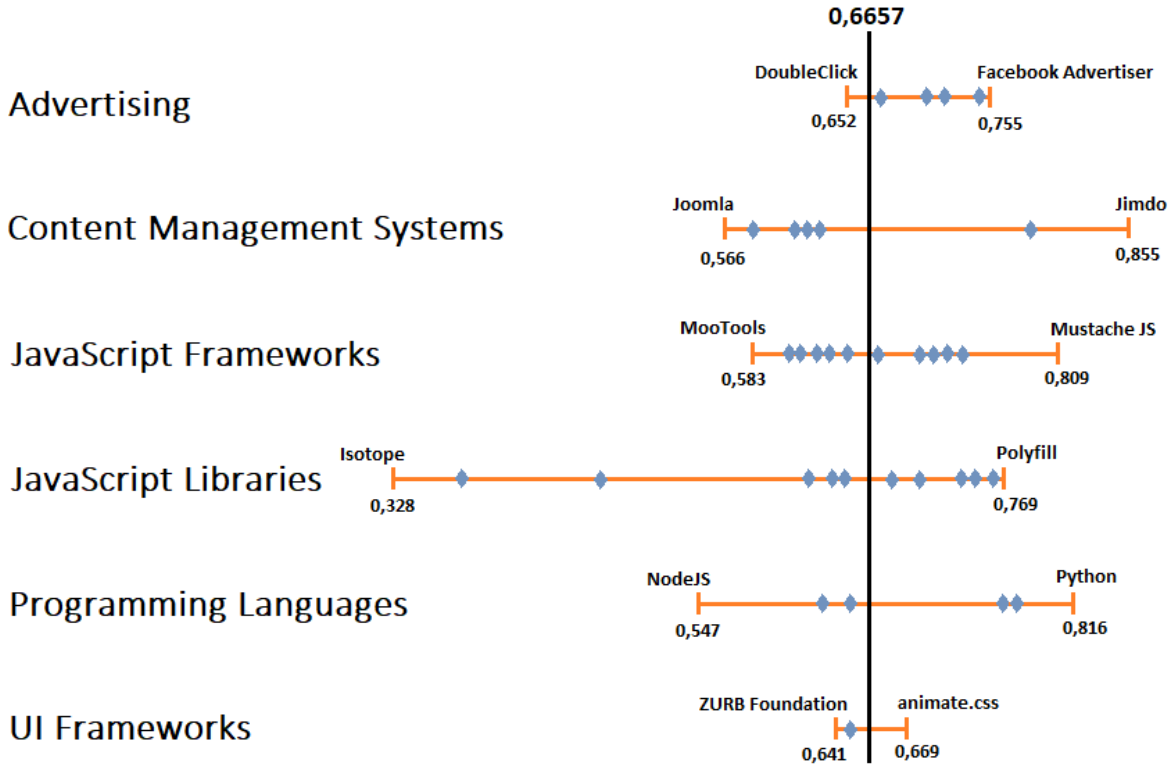

Figure 4.17: A3 metric scores of pages that use each technology of categories that were identified in more than 1 million pages

4.2 Discussion

According to the data obtained from this section's analysis, there are some interesting conclusions about web accessibility and web technologies.

Concerning the automated tools used to run all the evaluations and identifications, an important aspect is their accuracy. We only used automated tools whether to evaluate the web accessibility or to identify the web technologies. To be certain about the web technologies used in the development of our domain sample, and so achieve more corrected results, we used two web technologies identification tools. All the three tools successfully performed the evaluations and identifications in most of the web pages and domains.

When assessing the accessibility of the Web, it was possible to conclude some aspects regarding the Top-level Domains (TLD) of the evaluated web pages. We could create a data set composed of a considerable variety of web pages. According to the presented results and figure 4.18, Brazil web pages tend to have more errors compared to the other European countries like Portugal, France, Italy and the United Kingdom. Nevertheless, Portugal web pages have approximately 35 errors by page, which has a difference of only 5 errors from the TLD with more accessibility errors by page. The average number of accessibility errors by page vary between 16 and 40 depending on the TLD.

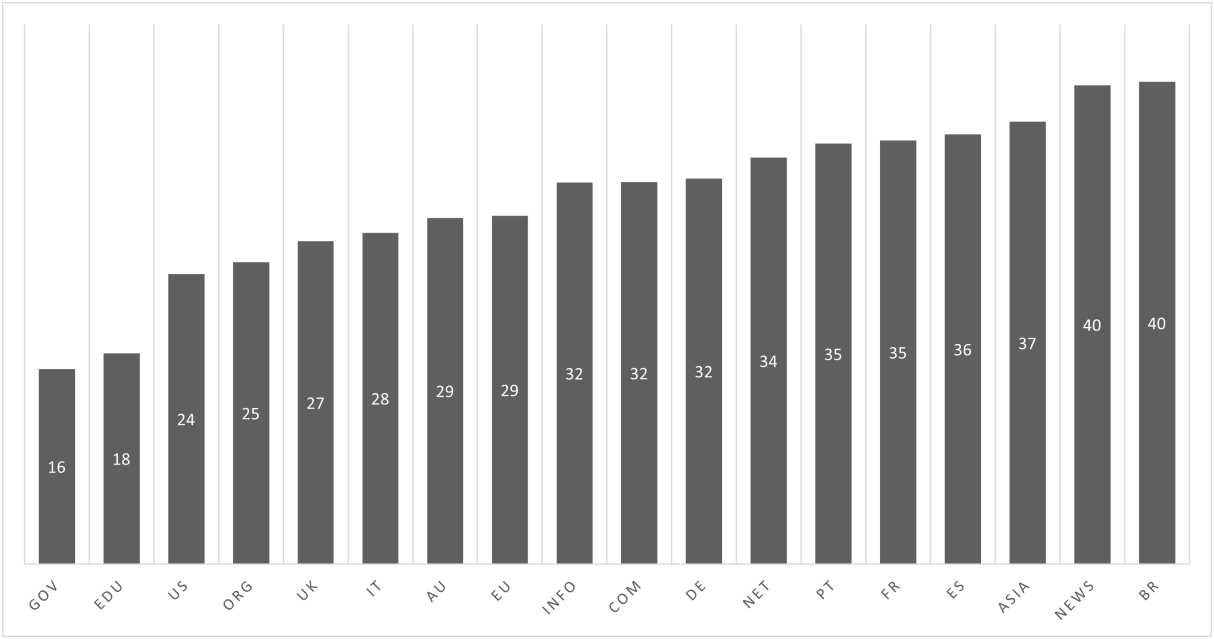

Figure 4.18: Proportion of accessibility errors by page by each analyzed Top-level Domain

The majority of the evaluated web pages presented accessibility barriers regarding the contrast of their web content. This suggests that developers are not often concerned with the way people with vision impairments deal with the colors' organization of the web content. Similarly to the WebAIM [2021] study, the contrast failures occur in the majority of the web pages, as they are the most commonly identified web accessibility issue. In WebAIM [2021], the low contrast text failure was detected in 86.4% of the web pages, while in our study, the test that assesses the minimum text contrast (ACT-Rule R37), fails in around 66% of the web pages. Accessibility failures regarding images, are present in 30.12% of the web pages in our study, while in WebAIM [2021], they found this type of errors, in

particular regarding the missing alternative text for images, in 60.6% or the web pages. These images' accessibility issues were found in twice as many web pages. Nevertheless, it is important to state that the scope of WebAIM [2021] study, resides in the assessment of home pages, which may be the reason why these percentages vary. The same significant difference between the percentage of the home pages happens when considering the form input labels and buttons. In WebAIM [2021], there are more web pages having problems associated with form labels, which means that disabled users would have more difficulty knowing details about the form field. This may prevent users from filling out the forms. There are also more pages having problems regarding buttons elements and their non-empty accessible names, which means that the assistive technologies could not give the user a context about the behavior of a given empty button. On the other hand, accessibility problems concerning links that should have non-empty accessible names, are verified in an average of 51.5% of the web pages in both studies. The *Missing document language* accessibility issue reported in WebAIM [2021] is not verified in any ACT-Rule by QualWeb tool.

This difference in the percentages of web pages between our study and WebAIM [2021] study can be explained through the fact that the home pages might contain different HTML elements that affect the accessibility evaluation results. The fact that the authors of WebAIM [2021] performed evaluations to home pages, and the fact that their results tend to differ from our results in some aspects, made us question if their procedure could be a correct approach to represent the accessibility of the Web. Hence, we compared the accessibility of the home pages with the accessibility of the remaining web pages of a set of websites, in section 6.

Besides evaluating the web accessibility, we also identified which web technologies are present in our domains' sample. The most popular technology was jQuery. It is the most frequently used technology when developing websites, since it was identified almost 188 thousand times by Wappalyzer and SimilarTech. PHP, WordPress and Bootstrap are some examples of the most frequently used technologies that have close occurrences' number, that vary between 63 and 108.

All technologies can be grouped according to categories. We verified that the category containing the Advertising technologies has the highest proportion of accessibility errors per number of pages (36.1). Accessibility category, however, presents lower proportion of errors by each page (22.4). Technologies of this category, like UserWay or AudioEye, might be more conscious about the accessibility problems and provide tools for the development of web content that are more aware about this matter. The JavaScript Libraries has a low error proportion and has the highest number of errors, compared to the remaining analyzed categories, since it is identified in the majority of the web pages.

By performing two experiments to explain the impact web technologies have in the accessibility of the web content, we could analyze the technologies of some categories that indicated better accessibility levels: Accessibility, Programming Languages, Content Management Systems, Web Frameworks, JavaScript Libraries, JavaScript Frameworks, JavaScript Graphics, LMS, PaaS, Page Builders, Rich Text Editors, Static Site Generator, Wikis, Multilingual, Online Forms and UI Frameworks categories. Nevertheless, all these categories have technologies that lead to better or worse accessibility scores. For instance, in Programming Languages category, Python (μ = 0.816) and Java (μ = 0.778) have the worst accessibility A3 metric scores, while NodeJS presents the best A3 metric scores average (μ = 0.547). Besides, PHP (μ = 0.650) and Ruby (μ = 0.630) demonstrate to have the second best A3 metric scores

compared to the remaining programming languages. For Content Management Systems category, Drupal (μ = 0.586) and Joomla (μ = 0.566) lead to the lowest accessibility scores, with a minimal difference of 0.02. In both Web Frameworks and UI Frameworks categories, a considerable difference between each technology score was not evident. Concerning JavaScript Libraries and Frameworks, the same situation is not verified as it compares a wider variety of technologies. It is possible to highlight the Isotope (μ = 0.328), Slick (μ = 0.378) and jQuery (μ = 0.472) technologies to be three best JavaScript Libraries technologies having the lowest accessibility scores. In contrast, Boomerang (μ = 0.849) reveals to be the worst technology, leading to higher accessibility scores. In terms of JavaScript Frameworks, it is interesting to notice that MooTools (μ = 0.583) tends to be identified in more accessible web pages, and, conversely, Mustache JS (μ = 0.809) behaves in the opposite way.

Chapter 5

Comparing accessibility metrics

Once the accessibility evaluation is performed, the process of obtaining a concrete answer regarding the accessibility of the web content from the assessment results is a complicated mission, due to the amount of obtained information. Thus, it is important to synthesize the results of the evaluations into one specific value, in order to obtain an overview about the accessibility level of the web content. This accessibility overview can be represented as a quantitative or qualitative value as a result of a measurement. Web accessibility metrics are responsible for measuring the web content and provide a result.

In this chapter, we gathered a set of accessibility metrics and applied them over our sample of 2.8 million web pages. First, we explain the details about 11 accessibility metrics and how we are going to compare and analyze them, regarding their similarities and validity. Finally, we present the obtained results as well as their discussion.

5.1 Methodology

In this section we present the methodology followed to compare a subset of the metrics presented in section 2.3. We introduce a description of which metrics we were able to compare, based on the constraints of running a large-scale study, which prevented us to compare metrics that rely on human judgement. We also describe how the metrics were implemented, based on the results provided by the used tool. We conclude this section with a description of how we analyzed the data before introducing a small set of web pages that we used to assess the validity of the different metrics studied.

5.1.1 Applicable metrics

The analysis of the accessibility metrics shows that not all metrics can be studied with this data set. For instance, metrics that require human judgement cannot be considered, since it is not viable to produce expert judgement over such a large set of pages. Therefore, we needed to identify the ones that could be computed with our data. From the 19 presented metrics, we found that 11 metrics could be computed with our dataset comprised by ACT-Rules evaluation results. Most of the referred metrics use WCAG 1.0 which considers checkpoints rather than success criteria. Since we are using WCAG 2.1 when computing the accessibility metrics, we will refer to the checkpoints as success criteria. Each ACT-Rule has corresponding success criteria. Each success criterion has an associated principle and conformance level. Through these success criteria, it was possible to define which principle(s) and conformance level(s)

Table 5.1: Constants used to compute the WAQM metric

Constants	**Description**	**Value**
Nall	Number of all tests	72
N	Number of applicable tests	51
Np	Number of Perceivable tests	28
No	Number of Operable tests	11
Nu	Number of understandable tests	7
Nr	Number of Robust tests	11
Npe	Number of automatic perceivable tests	26
Noe	Number of automatic operable tests	11
Nue	Number of automatic understandable tests	6
Nre	Number of automatic robust tests	10
Npw	Number of manual perceivable tests	5
Now	Number of manual operable tests	4
Nuw	Number of manual understandable tests	1
Nrw	Number of manual robust tests	1
a	Constant	20
b	Constant	0.3

characterize each test. As one test can have more than one success criterion, it also can have more than one principle or priority level. This information is required by some of the metrics.

The FR metric is the simplest metric to compute. It requires the number of potential and actual problems. For each page, the sum of all elements that failed a test and the sum of all elements applicable to the test are computed. Having both totals for all the tests, it is possible to calculate the failure rate of the page. It is important to highlight that some tests might evaluate the same elements of the page. However, they evaluate different aspects and so they cannot be counted only once, since we are considering the total number of failures and not the total number of elements that failed. For the remaining accessibility metrics that utilize the number of potential and actual points of failure for success criteria, the same logic was applied.

The WAQM metric is the most complex to compute. It considers the priority level and its weight, the type of the test, i.e. if it is manual or automatic, and the principle(s) of each test. WAQM is computed for each test and its computation relies on a number of parameters. Table 5.1 presents the parameters we used. Parameters a and b are constants once "the tuning was not necessary because WAQM proved to be independent of the tool when conducting large-scale evaluations (approx. 1400 pages)" [Vigo and Brajnik, 2011]. The other parameters were tuned to the QualWeb tool and the ACT-Rules it tests.

It was also possible to compute the UWEM and A3 metrics for each web page, since they both rely on the FR of each checkpoint of that page. Since both metrics are computed using a weight that is obtained from user feedback, we had to determine this weight according to the priority levels, due to time and resources constraints. UWEM already calculates a score for each website, by calculating its web pages' average score. Besides applying the UWEM metric to websites as the authors define [Vigo et al., 2009], we decided to use other procedure to convert this metric into a website metric, as will be described further on.

WIE, Conservative, Optimistic and Strict are four simple metrics that can be easily applied with our data, as they only require the number of applied success criteria, the number of elements, the number of

warnings, the number of fails and the number of passes.

In what concerns metrics that are applicable to websites instead of web pages, we considered WAB by Parmanto and Zeng and WAB by Hackett. The two WAB formulas were applied as one requires the priority level and the other one the weight of the priority level. Both formulas also calculate the failure rate and consider the total number of web pages a website contains.

Other metrics like SAMBA or eXaminator were not considered, either because of the lack of information in our data or the fact that the metric is semi-automated, which means that it needs manual intervention. For instance, the indexes that are computed in SAMBA metric concern the disability type; and eXaminator considers information about HTML elements that are evaluated in each page. Yet, we could partially use the WAEM/RA-WAEM metric. Since both WAEM and RA-WAEM require users' intervention as they evaluate pairs of websites, i.e. PUEXO pairs, to be compared to the results of the weighted accessibility score computation (equation 2.16), we could only consider the weighted accessibility score (equation 2.15) that can be automatically computed. This score is used in the WAEM and RA-WAEM metrics' process to classify a website and to compare the results with user classifications, and it considers the number of pages a success criterion passes in that website.

We did not consider the OAM nor the Page Metric metrics since they both consider the number of HTML attributes in their formulas. We do not have that information from the QualWeb reports. Also, we did not consider the two Fukuda et al. [2005] metrics since we do not have information regarding the aspects both formulas take into consideration (alt attributes, reaching time of a given element, page size).

We could not apply BIF for the reason that it needs a table that relates the errors that were identified by the accessibility evaluation tool with the assistive technologies that are affected by these errors. For this reason, it is not viable to attend to all the errors of our 2.8 million web pages sample and identify which assistive technologies are affected by them.

WABS was not considered since it classifies the accessibility barriers based on their severity, which means that it refers to the severity of a barrier that was identified in a set of websites and their respective web pages. Thus, this metric is focused on a specific problem that hinders the user's interaction. The final result of this metric would be a list of barriers that were found in our web pages data set and their respective severity scores. For this reason, it is not possible to correlate metrics that evaluate the accessibility of web pages with metrics that evaluate accessibility barriers.

The following list summarizes what metrics were analyzed in our study:

- Web page metrics:
  - Failure-rate (FR)
  - Unified Web Evaluation Metric (UWEM)
  - A3
  - Web Accessibility Quantitative Metric (WAQM)
  - Web Interaction Environments (WIE)
  - Conservative
  - Optimistic
  - Strict

- Website metrics:
  - Web Accessibility Barriers (WAB) by Hackett
  - Web Accessibility Barriers (WAB) by Parmanto & Zeng
  - Web Accessibility Evaluation Metric (WAEM)

5.1.2 Metrics comparison and analysis

With our goal being to understand the similarities of the different accessibility metrics, we computed their correlation pairs. We tested the normality of the data using the Shapiro-Wilk and Kolmogorov-Smirnov tests. We found that our data did not follow a normal distribution. Therefore, we used the Spearman correlation in our analysis. Following the recommendations in Statstutor [2021], absolute correlation values above 0.4 represent moderate or stronger correlation. In our analysis we considered two metrics to have similar results when they are at least moderately correlated.

It is important to take into consideration the fact that some metrics are applicable to websites whereas others are applicable to web pages. To be able to compare all metrics, the web page metrics were converted to web site metrics via two different approaches:

1. Computing the metric score based on the sum of the evaluation results of all the pages of the website and

2. Calculating the average of the metric score for all web pages of the web site.

Besides analyzing the pairwise similarity obtained from the correlation, we used this information to cluster the correlation scores and find if groups of metrics present similar behaviors in our dataset. For this analysis we used hierarchical clustering.

5.1.3 Metric validity

Our analysis allows detecting what metrics produce similar outcomes, but it does not reflect the validity of the outcomes. This is a result of the fact that the metrics were computed from a set of automated evaluation results. Automated evaluation tools are only capable of identifying a subset of the real accessibility problems in a web page. Therefore, a page that gets a good outcome on an automated accessibility evaluation might have uncaught accessibility problems. This means that a metric computed on that evaluation might indicate an accessibility level that is better than the reality.

To investigate what metrics might better reflect the actual accessibility level of web pages, we conducted a further analysis. Since it was not feasible to conduct manual assessments of the accessibility of the large data set, we compiled a small data set composed of web pages that are published online with the purpose of demonstrating good and bad accessibility practices. Table 5.2 presents the web pages we considered in our analysis. We computed the metrics outcomes for all the pages in table 5.2, and analyzed the accessibility level they reported the pages to have.

Table 5.2: Accessible and inaccessible pages

Web pages	
Accessible web pages	**Inaccessible web pages**
https://wsnet2.colostate.edu/cwis24/acns/web-accessibility/Example	https://wsnet2.colostate.edu/cwis24/acns/web-accessibility/Example/index-inaccessible.html
https://www.w3.org/WAI/demos/bad/after/home.html	https://www.w3.org/WAI/demos/bad/before/home.html
https://www.washington.edu/accesscomputing/AU/after.html	https://www.washington.edu/accesscomputing/AU/before.html
https://www.w3.org/WAI/demos/bad/after/template.html	https://www.w3.org/WAI/demos/bad/before/template.html

Table 5.3: Descriptive statistics for web page metrics

Metric	Average	Standard deviation	Best score	Worst score	First quartile	Third quartile
FR	0.0673	0.0780	0	1	0.0201	0.0856
A3	0.6657	0.3203	0	1	0.4077	0.9443
UWEM	0.3842	0.3212	0	0.9997	0.1010	0.800
WAQM	82.8626	19.1529	100	0	76.3289	95.6360
WIE	0.5545	0.1561	1	0	0.4375	0.6667
Conservative	0.3936	0.2316	1	0	0.2018	0.5556
Optimistic	0.6015	0.2314	1	0	0.4453	0.7852
Strict	0.4973	0.2692	1	0	0.2658	0.7209

5.2 Results

This section presents the results of the metrics comparison study. We begin by presenting an overview of the outcomes of each metric in the full set of web pages evaluated. We then examine the similarity between metrics across different contexts: metrics over web pages, and metrics over web sites, exploring both ways previously introduced to compute a website metric from web page metrics. We finalize with a presentation of the metrics scores over the set of pages with known high and low accessibility.

5.2.1 Descriptive results

Table 5.3 presents descriptive statistics of the scores for all metrics that are applicable at page level.

Regarding the descriptive statistics for web page metrics, we could conclude some interesting points. The FR metric average indicates that the accessibility of the evaluated web content is very optimistic. Besides, the standard deviation is also very small, indicating that the web pages scores do not vary much from the average. Also, WAQM presents a more positive perspective about accessibility of the Web, as the average is approximately 83, which is close to the maximum score that expresses the highest accessibility level. The UWEM metric is slightly positive concerning the accessibility, having an average of 0.38 and given the fact that the lower the score, the more accessible the web page is. The WIE, Optimistic and Strict metrics present an intermediate accessibility level average. In contrast, A3 and

Table 5.4: Descriptive statistics for website metrics, adding the evaluation results of all website pages

Metric	Average	Standard deviation	Best score	Worst score	First quartile	Third quartile
FR	0.0832	0.0836	0	1	0.0296	0.1080
A3	0.8390	0.2713	0	1	0.8301	1.0000
UWEM	0.4728	0.3294	0	0.9997	0.1715	0.8131
WAQM	79.4362	21.7414	100	0	71.7592	94.8497
WIE	0.5176	0.1662	1	0.04	0.400	0.6250
Conservative	0.4327	0.2191	1	0.0006	0.2640	0.5799
Optimistic	0.6366	0.1994	1	0.0006	0.5065	0.7857
Strict	0.5390	0.2410	1	0.0006	0.3515	0.7273
WAB-H	0.4742	0.6927	0	5.8333	0.0263	0.6875
WAB-PZ	0.3053	0.4799	0	4.2	0.0133	0.400
WAEM	4.1765	1.3273	8.68	0.0072	3.3353	5.1446

Table 5.5: Descriptive statistics for website metrics, considering the average of the website pages' metric scores

Metric	Average	Standard deviation	Best score	Worst score	First quartile	Third quartile
FR	0.0859	0.0849	0	1	0.0312	0.1111
A3	0.6744	0.3007	0	1	0.4569	0.9255
UWEM	0.4433	0.3208	0	0.9997	0.1562	0.800
WAQM	77.9130	21.3167	100	0	70.1056	93.1619
WIE	0.5715	0.1562	1	0.0588	0.4645	0.6797
Conservative	0.4457	0.2182	1	0.00055	0.2777	0.5955
Optimistic	0.6470	0.1970	1	0.00055	0.5178	0.7968
Strict	0.5527	0.2384	1	0.00055	0.3688	0.7402
WAB-H	0.4742	0.6927	0	5.8333	0.0263	0.6875
WAB-PZ	0.3053	0.4799	0	4.2	0.0133	0.400
WAEM	4.1765	1.3273	8.68	0.0072	3.3353	5.1446

Conservative metrics report a more negative perspective about the web accessibility of the evaluated web pages, as their values are closer to the inaccessible reference.

Tables 5.4 and 5.5 present descriptive statistics of the scores of all metrics at the website level. Table 5.4 shows results where scores for page level metrics were calculated by adding the evaluation results for all pages of the same website. Table 5.5 results were calculated averaging the page metric results for all pages of the same website.

In relation to the descriptive statistics for website metrics, there are some differences that could be observed. The FR average indicates that the accessibility of the evaluated web content is very optimistic, as it was observed in this metric's web page version. Also, WAQM still presents a more positive perspective about the accessibility of the Web, as it was stated in the descriptive analysis of the web page metrics. The UWEM metric has slightly increased its average when comparing with this metric applied to web pages. Nevertheless, it still provides a positive perspective about the accessibility. The Optimistic and Strict website metrics' average also increased, yet it does not show an evident difference compared to the web pages metrics. The average of the website scores using the WIE metric decreased when the evaluation results for all pages of the same website was considered, compared to this metric's web page

Table 5.6: Spearman correlation scores for web page metrics

	FR	A3	UWEM	WAQM	WIE	Conservative	Optimistic	Strict
FR	1							
A3	0.0008	1						
UWEM	0.0008	**0.8375**	1					
WAQM	0.0002	-0.0173	-0.0175	1				
WIE	-0.0006	**-0.6342**	**-0.4963**	0.0099	1			
Conservative	0.0001	-0.0178	0.0226	0.0061	0.0403	1		
Optimistic	0.0004	-0.0209	0.0240	0.0070	0.0456	**0.9042**	1	
Strict	0.0003	-0.0193	0.0231	0.0068	0.0427	**0.9706**	**0.9733**	1

version. The same did not happen when considering the average of the page metric results for all pages of the same website, as it shows an increase on its average. Conservative, as a website metric, also has a similar negative perspective about web accessibility compared to the same metric applied to web pages. The A3 metric for websites, in particular, considering the average of the website pages' scores, did not show a noticeable difference in the average result, compared to the A3 metric for web pages (around 0.67). Nevertheless, a considerable difference between these last two approaches for A3 metric was detected in the website level, concerning a website as a web page, having an average of approximately 0.84. Interestingly, the WAB-H and WAB-PZ metrics reveal differences in their averages that might be justified from the worst scores. WAB-H evaluated a website that had a score of 5.8333, which represents the most inaccessible website. Yet, the WAB-PZ worst score was 4.2, which indicates that the accessibility issues are less weighted compared to WAB-H. Since WAB-PZ, WAB-H and WAEM do not provide a limited range of values, it is more complicated to define an accessibility level by their results' scores.

5.2.2 Web page metrics

Table 5.6 presents the Spearman correlation coefficients between every pair of page metrics. By comparing the web page metrics (FR, A3, UWEM, WAQM, WIE, Conservative, Optimistic and Strict) results, it was possible to find some interesting correlation scores.

The highest correlation score obtained was between the Strict and Optimistic metrics (ρ = 0.9733), followed by Conservative and Strict metrics (ρ = 0.9706) and Optimistic and Conservative metrics (ρ = 0.9042). They seem to have a very strong positive correlation, since they are all based on the ratio of passed tests over applicable tests, differing only on how warnings are considered. Possibly, the number of warnings classified by QualWeb was not high enough to ensure evident differences between the results of each of these three metrics applied to each web page.

A3 appears to have a strong positive correlation with UWEM (ρ = 0.8375), which is expectable since these two metrics are similar. Also, A3 has a strong negative correlation with WIE (ρ = -0.6342). Since WIE considers the number of elements that pass, the higher the WIE score, the higher the accessibility level of the web page. The A3 shows an opposite behavior. Similarly to A3, UWEM shows a moderate negative correlation with WIE (ρ = -0.4963). This behavior might be explained with the fact that both UWEM and A3 consider the number of elements that failed while WIE considers the number of success criteria that passed in a page. If a page that fails all the success criteria that are tested, also fails one element per test, the number of failed elements will be similar to the number of failed success criteria.

All the other pairs of metrics are not correlated.

Table 5.7: Spearman correlation scores for website metrics

	WAEM	WAB-H	WAB-PZ
WAEM	1		
WAB-H	-0.1183	1	
WAB-PZ	-0.1850	**0.9858**	1

5.2.3 Website metrics

In this study we considered 3 accessibility metrics that are exclusively applied at website level: WAEM, WAB-H and WAB-PZ. Table 5.7 presents the Spearman correlation coefficients between all pairs of metrics.

The WAB metrics show a very strong positive correlation as expected (ρ = 0.9858), since they share the same formula, with just one little difference: WAB by Hackett considers the priority level (1, 2 or 3) of the checkpoint (success criterion, in our case), whereas WAB by Parmanto and Zeng considers the weight of the checkpoint priority level (P1=0.8, P2=0.16 and P3=0.04). Both WAB metrics are not correlated with WAEM.

Domain as a web page

Table 5.8 presents the Spearman correlation coefficients for all metrics, with the website metrics being computed from page level metrics considering the sum of the evaluation results for each website.

Table 5.8: Spearman correlation scores for website metrics, considering a domain as a web page

	FR	A3	UWEM	WAQM	WIE	Conservative	Optimistic	Strict	WAB-H	WAB-PZ	WAEM
FR	1										
A3	0.1779	1									
UWEM	**0.4442**	**0.4131**	1								
WAQM	**-0.5423**	-0.2140	**-0.7285**	1							
WIE	-0.1154	**-0.5942**	-0.3649	0.2485	1						
Conservative	-0.0188	-0.3528	-0.0967	0.0333	**0.4838**	1					
Optimistic	-0.1336	-0.3519	-0.1329	0.0775	0.4523	**0.8759**	1				
Strict	-0.0984	-0.3730	-0.1283	0.0673	**0.4898**	**0.9573**	**0.9704**	1			
WAB-H	**0.4217**	-0.2910	**0.5698**	**-0.5675**	-0.0233	-0.0024	-0.0556	-0.0378	1		
WAB-PZ	**0.4071**	-0.2222	**0.6457**	**-0.6249**	-0.0525	-0.0098	-0.0591	-0.0434	**0.9858**	1	
WAEM	-0.3125	**-0.5252**	**-0.5057**	**0.4902**	**0.6566**	0.2870	0.2764	0.3044	-0.1183	-0.1850	1

As it would be expected, and matching to the web pages correlation scores, the Conservative metric has a very strong and positive correlation with Optimistic and Strict: ρ=0.8759 and ρ=0.9573, respectively. Also, as observed in the web pages scores, Strict and Optimistic metrics still have the same strong positive correlation (ρ=0.9704). WAEM appears to have a strong positive correlation with WIE (ρ = 0.6566). A3 has a moderate positive correlation with UWEM (ρ = 0.4131) and it is negatively correlated with WAEM (ρ = -0.5252) and WIE (ρ = -0.5942). UWEM has a moderate negative correlation with WAEM (ρ = -0.5057), while WAQM has a moderate positive correlation with WAEM (ρ = 0.4902), and it has a strong negative correlation with WAQM (ρ = -0.7285). UWEM also has positive correlation with both WAB-PZ (ρ = 0.6457) and WAB-H (ρ=0.5698). In contrast to UWEM, WAQM has negative correlations with WAB-PZ (ρ=-0.6249) and WAB-H (ρ = -0.5675). FR shows a positive moderate correlation with UWEM (ρ = 0.4442), with WAB by Hackett (ρ = 0.4217) and with WAB-PZ (ρ = 0.4071). It presents a moderate negative correlation with WAQM (ρ = -0.5423).

Table 5.9: Spearman correlation scores for website metrics, considering the average of the web pages' scores

	FR	A3	UWEM	WAQM	WIE	Conservative	Optimistic	Strict	WAB-H	WAB-PZ	WAEM
FR	1										
A3	**0.4568**	1									
UWEM	**0.4914**	**0.8612**	1								
WAQM	**-0.5606**	**-0.7167**	**-0.7917**	1							
WIE	-0.2018	**-0.6053**	**-0.4411**	**0.3914**	1						
Conservative	0.0014	-0.3129	-0.0916	0.0177	**0.5604**	1					
Optimistic	-0.1283	-0.3499	-0.1392	0.0748	**0.5253**	**0.8740**	1				
Strict	-0.0910	-0.3567	-0.1332	0.0630	**0.5685**	**0.9536**	**0.9718**	1			
WAB-H	0.4083	**0.5310**	**0.6161**	**-0.5036**	-0.2607	-0.0262	-0.0813	-0.0646	1		
WAB-PZ	0.3992	**0.5957**	**0.6846**	**-0.5665**	-0.2798	-0.0291	-0.0816	-0.0666	**0.9858**	1	
WAEM	-0.3390	**-0.5563**	**-0.5077**	**0.5369**	**0.5968**	0.2577	0.2583	0.2839	-0.1183	-0.1850	1

Average of the web pages' scores

Table 5.9 presents the Spearman correlation coefficients for all metrics, with the website metrics being computed from page level metrics considering the average of the web pages' scores for each website.

As it would be expected, and matching to web pages correlation scores, the Conservative metric has a very strong and positive correlation with Optimistic and Strict: $\rho = 0.8740$ and $\rho = 0.9536$, respectively. These values are very similar to the ones obtained when considering a domain as a web page. Also, as observed in the web pages scores, Strict and Optimistic metrics have the same strong positive correlation ($\rho = 0.9718$).

WAEM appears to have a moderate positive correlation with WIE ($\rho = 0.5968$).

Unlike results obtained from considering a domain as a web page, A3 has a very strong positive correlation with UWEM ($\rho = 0.8612$). Since A3 and UWEM are very similar, having a strong correlation, WAQM also shares similar correlations with both metrics: ρ = -0.7167 and ρ = -0.7917, respectively. The same happens with WAB-PZ and WAB-H. Since these two metrics are strongly correlated, their correlations with WAQM are also very similar: ρ = -0.5665 and ρ = -0.5036, respectively.

UWEM also has a strong positive correlation with WAB-PZ ($\rho = 0.6846$) and WAB-H ($\rho = 0.6161$). A3 has a similar correlation score with WAB-H ($\rho = 0.5310$) and WAB-PZ ($\rho = 0.5957$). However, it has a negative correlation with the remaining metrics: Moderate with WAEM (ρ = -0.5563), and strong with WIE (ρ = -0.6053) and WAQM (ρ = -0.7167).

UWEM and WAQM have moderate correlation with WAEM: ρ = -0.5077 and ρ = 0.5369, respectively. They both share a moderate to almost moderate correlation with WIE: $\rho = -0.4411$ and $\rho = 0.3914$.

WIE correlation scores with Conservative, Optimistic and Strict metrics vary between 0,52 and 0,57.

FR shows a positive correlation, although moderate, with UWEM ($\rho = 0.4914$) and A3 ($\rho = 0.4568$). However, FR presents a negative correlation with WAQM (ρ = -0.5606).

5.2.4 Metric validity

Table 5.10 presents the scores of each page level metric for each of the pages used to assess the validity of the metrics. Table 5.11 presents the scores for website metrics. To avoid the canceling effect in these metrics of having a website with the same number of accessible and inaccessible pages we split each website in two websites: a good website with the accessible pages and a bad website with the inaccessible pages.

Table 5.10: Web page metrics scores for assessing the metrics' validity

	FR	A3	UWEM	WIE	Conservative	Optimistic	Strict	WAQM
Accessible								
https://wsnet2.colostate.edu/cwis24/acns/web-accessibility/Example	0.00218	0	0	0.875	0.8297	0.9454	0.93827	98.823
https://www.w3.org/WAI/demos/bad/after/home.html	0.01954	0.02192	0.0057	0.6	**0.3909**	0.70684	0.5714	89.005
https://www.washington.edu/accesscomputing/AU/after.html	0.00998	0	0	0.7333	0.525	0.6367	0.5910	90.667
https://www.w3.org/WAI/demos/bad/after/template.html	0.0185	0.0192	0.0057	0.6	**0.384**	0.7159	0.5746	89.005
Inaccessible								
https://wsnet2.colostate.edu/cwis24/acns/web-accessibility/Example/index-inaccessible.html	**0.271**	0.997	0.967	0.308	0.1050	0.2514	0.1230	33.2218
https://www.w3.org/WAI/demos/bad/before/home.html	**0.1453**	0.9998	0.939	0.267	0.1738	0.444	0.238	38.6422
https://www.washington.edu/accesscomputing/AU/before.html	**0.0704**	0.995	0.915	**0.6154**	**0.7605**	**0.901**	**0.885**	51.7320
https://www.w3.org/WAI/demos/bad/before/template.html	**0.1448**	0.999	0.9518	0.2667	0.1785	0.468	0.251	35.420

Table 5.11: Website metrics scores for assessing the metrics' validity

	WAEM	WAB-PZ	WAB-H
Accessible Domains			
wsnet2.colostate.edu	11.06	0.0008	0.007
www.w3.org	**4.430**	0.001	0.006
www.washington.edu	8.860	0.001	0.011
Inaccessible Domains			
wsnet2.colostate.edu	**5.440**	0.83	1.10
www.w3.org	1.920	0.406	0.521
www.washington.edu	3.840	0.811	1.042

Web page metrics

The results of this experiment show that the FR metric produces similar scores when evaluating accessible and inaccessible web pages. Since 1 means that the web page is completely inaccessible and 0 means otherwise, we were expecting to have values close to 1 for the inaccessible web pages and close to 0 for the accessible web pages. The FR scores for all the accessible web pages seem to be coherent and close to 0. However, all the inaccessible web pages also have low values, indicating a positive accessibility level.

WIE, Conservative, Optimistic and Strict metrics exhibit a score close to 1 for the same inaccessible web page, which means that this page is close to be completely accessible. Also, these metrics' scores for this particular inaccessible page are higher than some accessible pages' scores. This means that the inaccessible page is more accessible than some accessible pages, according to WIE, Conservative, Optimistic and Strict metrics' results. The remaining scores for these metrics seem to be coherent, except for Conservative metric that classifies two accessible web pages as inaccessible, by showing scores close to some of the inaccessible pages' scores.

A3, UWEM and WAQM are the only three metrics that demonstrated coherent scores for all accessible and inaccessible web pages. WAQM metric shows values close to 100 for all the accessible web pages. For the inaccessible pages, this metric varies from around 33 to 51, which is not close to 0. A3 and UWEM proved to have a correct behavior as all the accessible pages scores are close to 0 and the inaccessible pages scores are close to 1. Nevertheless, A3 metric scores for inaccessible web pages are more close to 1 compared to UWEM metric scores for the same web pages.

Website metrics

Since the three website accessibility metrics do not have a range of scores limited by two values, the level of accessibility of a certain website becomes uncertain. The main conclusion we can take from the WAB metrics is that the higher the score, the more inaccessible the website is. Nevertheless, it is also possible to detect that the accessible domains' scores are really close to 0, which indicates the domains are more accessible. In addition, and in contrast to the accessible domains, the inaccessible domains' scores are higher and close to 1. By observing the table 5.4 or 5.5, the accessible scores are in the first quartile, while the inaccessible scores belong to the third quartile, indicating that WAB-PZ and WAB-H may have an appropriate representation of the accessibility.

As for the WAEM, the higher this metric's score, the more accessible the website is. Consequently, we cannot define whether a website is accessible or inaccessible. This metric seems to be the only one with incoherent results as the www.w3.org accessible domain presents a lower score compared to the wsnet2.colostate.edu inaccessible domain.

5.3 Discussion

To identify the groups of metrics, we used hierarchical clustering that groups similar metrics into clusters according to the correlation matrices. To define the number of clusters, we had to cut the dendrograms in order to define the different clusters. To obtain an interesting number of clusters, we analyzed the dendrograms to determine the best cluster distance where we would cut the dendrogram. With respect to web page clusters, we decided to cut the dendrogram where the cluster distance is 1. Concerning the website metrics clusters, we cut the dendrogram where the cluster distance is approximately 0.7. The dendrograms' cuts are represented by the yellow and red lines.

With regards to the web accessibility metrics' results, and concerning the web page metrics, we could find four groups: Conservative, Strict and Optimistic; WIE, A3 and UWEM; FR; and WAQM. The clusters are illustrated in figure 5.1.

The Conservative, Strict and Optimistic metrics have similar formulas based on the number of passed tests over applicable tests, differing on whether warnings are considered passes, fails or not applicable. For this reason, it is not unexpected that they produce similar outcomes and are clustered together.

A3 and UWEM also have very similar formulas, which justifies the high correlation between their outcomes. While A3 and UWEM are based on the ratio of actual and potential barriers, WIE considers pass rates of checkpoints. Even though those perspectives of measuring the accessibility of web pages are different, the fact is that they seem to be correlated. This might be relevant information when deciding on using one of these metrics, since WIE requires information that is easier to obtain and less resources

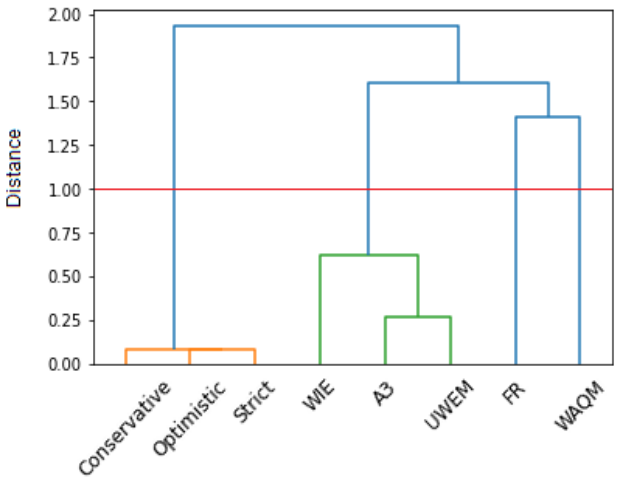

Figure 5.1: Clusters of the web page metrics

to compute than A3 or UWEM.

WAQM did not form a group with FR, as their distance is higher than 1. The distance between these two metrics may be significant, once their correlation is almost null.

With respect to website metrics, in particular interpreting a website as a web page, the following five clusters were identified, as represented in figure 5.2:

- Conservative, Strict and Optimistic
- A3, WIE and WAEM
- FR
- WAB-H, WAB-PZ
- UWEM and WAQM

From these groups, only one cluster is similar to the web page metrics' groups. In fact, the web pages cluster WIE, A3 e UWEM is similar to one cluster of the website metrics (A3, WIE and WAEM), except the fact that UWEM metric is not included into A3 and WIE group. The main difference is the fact that it now includes the correlations with the website metrics (WAB-H, WAB-PZ and WAEM). Besides this main inclusion, the WAQM seems to be closer to UWEM, compared to the web page metrics' results. This could be happening because all the web pages data are grouped together to provide the website final score, modifying the metrics' behavior. The FR is still distant from the remaining metrics. WAQM and UWEM calculate the failure rate, which might justify the cluster they are grouped in. WAB-PZ and WAB-H had a very strong correlation, so it was expected they would be part of the same cluster. They share similar formulas that only differ in the way they consider the success criterion weight: as the priority level or the weight of the priority level. WIE may relate with WAEM, since they both consider when a checkpoint passes: WIE increments 1 every time a certain checkpoint passes on a website, while WAEM counts the number of pages where a checkpoint passed. For instance, if a certain checkpoint

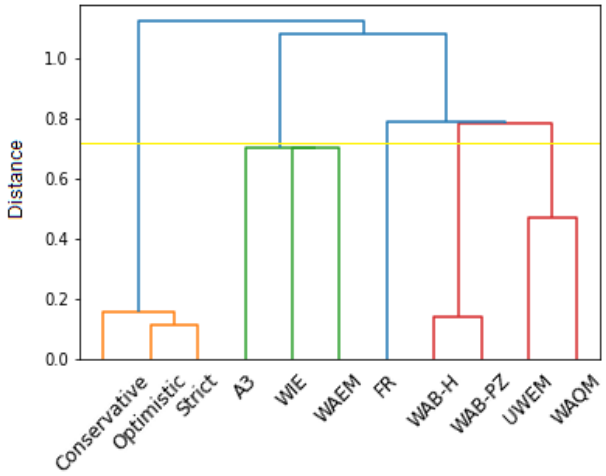

Figure 5.2: Clusters of the website metrics, interpreting a website as a web page

passes on a website that only has one web page, it will count as 1 for WIE and also for WAEM. The main difference between them is that WIE considers the total number of checkpoints, while WAEM not only considers the number of website pages but also the weight of the checkpoint.

Regarding the average of the web pages' scores, figure 5.3 represents the following six groups:

- Conservative, Strict and Optimistic
- FR
- A3, UWEM and WAQM
- WIE
- WAEM
- WAB-H and WAB-PZ

The above groups show that Conservative, Strict and Optimistic metrics belong to same cluster, as seen before. However, there are two main differences when comparing with the other website metrics approach: (1) A3 is now part of the UWEM and WAQM cluster; (2) WAEM and WIE do not belong to the same cluster, as their cluster distance is higher than the previously defined threshold.

Another interesting aspect is the fact that Conservative, Optimistic and Strict are always in the same independent group, in all the three approaches we have mentioned. Perhaps because the number of warnings of the considered domains and web pages is not that significant to the point of changing these metrics' results, since the only difference between these three metrics' formulas is the way the warnings are considered.

In all metrics' clusters, the FR metric does not form a group with any other metric, even though some of them incorporate the failure rate in their formulas, indicating that their results do not correlate with the FR metric results. Thus, we can recognize that the metrics that integrate the FR, consider other important information in their scope that makes them different from the FR. For instance, WAQM is more complex

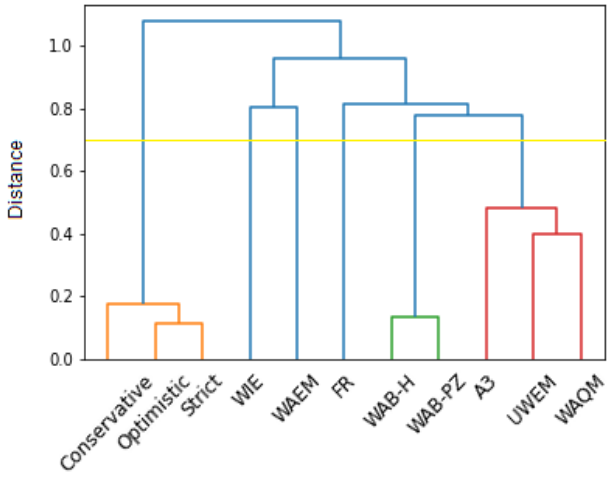

Figure 5.3: Clusters of the website metrics, calculating the average of the web pages' scores

than FR, taking into account the principles, the type and the priority levels of success criteria. For this reason, when opting for one of these metrics, FR seems a more straightforward and easy choice, but it should be kept in mind that the other metrics may give more relevant information.

5.3.1 Metric validity

To define which metric is the most suitable option, it is important to analyze their results regarding the accessible and inaccessible web pages' and domains' evaluations.

FR seems to have coherent scores for all the accessible web pages. This means that all the accessible web pages have expected results. However, all the inaccessible web pages have low scores, indicating that these web pages are accessible when they are not.

WIE, Conservative, Strict and the Optimistic metrics always fail to assess the accessibility level of the inaccessible web page *https://www.washington.edu/accesscomputing/AU/before.html*, showing high scores that indicate the web page is accessible. Also, Conservative assigns a score that is close to some inaccessible pages' scores to the *https://www.w3.org/WAI/demos/bad/after/home.html* accessible web page, which means that this page is not accessible.

A3, UWEM and WAQM seem to have the expected behavior. Still, the A3 metric shows to be more discriminative, as its scores are closer to what is supposed to be accessible and inaccessible compared to the UWEM and WAQM metrics.

Regarding the website metrics, it was possible to state that WAB-PZ and WAB-H seem to have a correct behavior, although it is not possible to define an accessibility level as these metrics do not provide a limited range of values.

Chapter 6

Comparing the accessibility of the home pages with the remaining website's accessibility

Up to now, it is possible to state that several studies do solely rely on the evaluation of home pages instead of all web pages, or a larger sample of pages, of a given website. The authors of these studies draw conclusions about the accessibility of websites, believing in the assumption that the home pages are a representation of all the website. As a consequence, the reliability of the study might be compromised. Therefore, we went further and investigated whether this belief can be incorporated, in order to produce reliable results concerning a web accessibility overall analysis. In this analysis, we will consider the remaining web pages of a website, excluding the home page, as the interior pages.

This chapter compares and analyzes the accessibility of the interior web pages and the home pages of a set of websites, so it is possible to state whether assessing the accessibility of only home pages is representative enough to conclude valid accessibility aspects. To perform this analysis, we investigated the A3 metric scores of all web pages and then we compared the average of the interior web pages of the analyzed websites with all home pages' scores. We have also investigated the reported issues in both approaches.

The methodology will be further explained in more detail. The results and their discussion will follow this section.

6.1 Methodology

In this section, we present the methodology used to implement the analysis about the accessibility of the home pages.

First, we selected all web pages from a website whose home page is the concatenation between the HTTP protocol and the domain url. For instance, *https://www.google.com/* is an example of a home page that is a result of the concatenation of the HTTP protocol *https://* with the domain name *www.google.com*. As another example, the domain name of the home page *http://qualweb.di.fc.ul.pt/evaluator/* is *qualweb.di.fc.ul.pt*. In this example, the concatenation of the HTTP protocol *http://* with the domain name *qualweb.di.fc.ul.pt* does not correspond to the actual home page *http://qualweb.di.fc.ul.pt/evaluator/*.

Thus, we only considered the domains for which we found a page that represents the concatenation of the HTTP protocol with the domain name, because, otherwise, we could not identify, from the set of pages belonging to a website, which one was the home page.

To measure the accessibility of the pages we continued to use the A3 metric, as it was possible to see that it is the most discriminative metric, as described in 5.1.3 section.

We used the Spearman correlation in this analysis to correlate the home pages' scores with the average of the remaining pages' scores. Following the recommendations in Statstutor [2021], absolute correlation values above 0.6 represent strong to very strong correlations. In our analysis we considered that the accessibility of the home pages is similar to the accessibility of the rest of the website pages if they are at least strongly correlated (> 0.6). Therefore, if the home pages scores are strongly correlated with the remaining website pages scores, it is an indication that the accessibility level of home pages is similar to the accessibility level of the website and they are good proxies to perform accessibility studies by evaluating only the home pages.

Besides the analysis of the correlation scores, we also verified the percentage of home pages and the percentage of interior pages that violate each ACT-Rule.

6.2 Results

Before analyzing the correlation between the interior web pages and the home pages, it is important to observe the distribution of the A3 metric scores of both types of web pages, that is presented in figure 6.1, as well as its descriptive analysis presented in table 6.1. The values of the A3 metric vary between 0 and 1 in both interior and home pages scores. It is also possible to identify that the average of the interior web pages scores is quite close to the average of the home pages. Nevertheless, the interior web pages appear to be more accessible than the home pages.

We calculated the Spearman correlation, since the samples did not follow a normal distribution. We found a correlation of $\rho = 0.6957$ that indicates a strong relationship between the A3 metric (equation

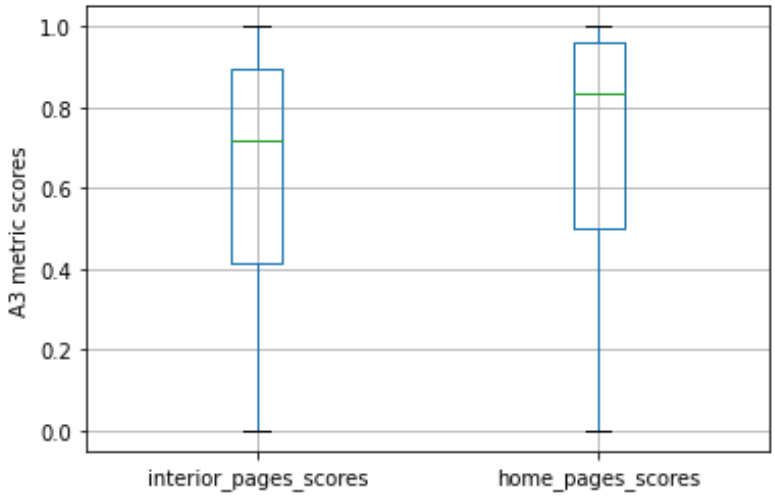

Figure 6.1: A3 metric scores of interior web pages and home pages

Table 6.1: Descriptive results of A3 metric scores

	Average	Best score	Worst score	First quartile	Median	Third quartile
Interior web pages	0.640	0	1	0.416	0.717	0.893
Home pages	0.704	0	1	0.501	0.833	0.961

2.3) scores calculated from home pages and the average of this metric's scores for the remaining web pages of the website.

Nevertheless, there are evident differences between interior web pages and home pages as stated in figure 6.1 and table 6.1. Besides, the calculated correlation states that the accessibility levels are similar, according to the A3 metric results, and does not give information about the accessibility problems that were detected in both approaches. Hence, the differences stated in figure 6.1 and table 6.1 led us to a further investigation about the ACT-Rules that are violated in both approaches. For all the interior and home web pages, the ACT-Rules were identified, as well as the percentage of web pages that violated each rule. Table 6.2 represents the results of the ACT-Rules which have a corresponding percentage of web pages that violate them higher than 10%. The complete tables' results are represented in appendixes section. Results indicate that, in general, there is a slightly higher percentage of home pages that fail each ACT-Rule compared to the percentage of interior pages, except for *Form control has accessible name*, *HTML has lang attribute*, *svg element with explicit role has accessible name*, *HTML Page has a title*, *All table header cells have assigned data cells*, *Element with role attribute has required states and properties*, *This rule checks that each aria- attribute specified is defined in ARIA 1.1.*, *Menuitem has non-empty accessible name*, *ARIA state or property has valid value* and *HTML lang and xml:lang match* ACT-Rules. Despite this difference between the percentage of interior and home pages, we can see that there is no accessibility issue that was reported on only one approach. This suggests that all the home pages detected the same problems as all the interior pages. Regardless, the fact that the A3 metric scores' average is higher for home pages, it can be concluded that home pages are less accessible. Considering table 6.2, the more violated issue in home and interior pages regards the contrast and there are more home pages reporting accessibility problems in this matter compared to interior web pages. Apart from this type of accessibility issues, other different problems are more frequent in home pages, which may be justified with the fact that they have more information and higher complexity in the main web page.

6.3 Discussion

A lot of studies usually discuss accessibility problems regarding different contexts by performing accessibility assessments only to home pages due to possible constraints. By means of this experiment, it was possible to prove the viability of these assessments and if they are enough to draw conclusions about the accessibility of the Web. After this analysis, and given the Spearman correlation result and the ACT-Rules that were violated in both home and interior pages, the conclusion about whether the home pages can be representative of the whole website seemed to be straightforward: home pages have similar accessibility levels compared to the remaining pages of a website, which conveys that the accessibility

Table 6.2: Number of failures and percentage of web pages by each ACT-Rule for home and interior pages

ACT-Rule	**Description**	**Percentage of home pages**	**Percentage of interior pages**
ACT-Rule R76	Text has enhanced contrast	78.630%	74.770%
ACT-Rule R37	Text has minimum contrast	65.121%	60.733%
ACT-Rule R12	Link has accessible name	56.690%	47.923%
ACT-Rule R17	Image has accessible name	30.978%	26.670%
ACT-Rule R18	id attribute value is unique	30.104%	26.628%
ACT-Rule R14	meta viewport does not prevent zoom	22.814%	22.015%
ACT-Rule R19	iframe element has accessible name	17.147%	14.760%
ACT-Rule R16	Form control has accessible name	17.021%	17.829%
ACT-Rule R2	HTML has lang attribute	15.932%	18.234%
ACT-Rule R35	Heading has accessible name	11.906%	8.072%

of the home page may be representative of the overall website accessibility.

Notwithstanding, and according to the boxplots that represent the interior and the home pages scores, the previous conclusion felt incomplete. For this reason, we investigated the accessibility problems that were detected in both home and interior pages, as well as their respective percentage of pages that violated each accessibility issue. We performed the same analysis twice: one for the home pages and another one for the interior web pages, in order to compare the type of accessibility failures that occur in both approaches. After analyzing the results, we have verified that home pages are less accessible than interior pages, and so they provide worse accessibility scores and analysis' results. They have a more negative perspective about the web accessibility. As the main web page needs to be appealing and complete regarding the information it provides, the quantity of the web content may need to increase when compared to the remaining web pages of the website. The majority of the home pages certainly have more images, links, text, and other HTML elements. Consequently, the probability of containing accessibility issues increases. Nevertheless, the percentage of home pages that violated each rule is quite similar to the percentage of interior pages. This analysis denotes that, although home pages might be less accessible than the other website pages, their accessibility scores are strongly correlated. As a result, we can say the accessibility of the home page is representative of the overall website accessibility.

Chapter 7

Conclusions

Web accessibility is responsible for making the Web open to disabled users. It aims to achieve high quality in the websites and web tools creation and development, making products more accessible to any user and avoiding the occurrence of barriers during the interaction.

Web accessibility can be evaluated through different types of tools and procedures that normally verify the conformance with a set of guidelines. Although this verification can assure the validation of certain HTML elements, developers should always try to follow good practices, in order to guarantee more accessible content. The tools used in the assessment of the accessibility of the Web can be automatic, semi-automatic, or manual. The choice of what procedure should be adopted depends on the context of the study in question. For instance, in large-scale contexts, an automated evaluation process is preferred, once it is not cost-effective to perform hundreds, thousands, or millions of evaluations by human experts.

One of the biggest challenges in website development and maintenance is the inclusion of good practices that improve the accessibility to all types of users. For that reason, there are tools and web technologies that are specialized to guarantee comfort and convenience in building websites. Through an analysis of the ACT-Rules that are violated in a web pages sample, an overview about the current accessibility status can be concluded. In particular, accessibility problems that are still committed. A specific problem that occurs in the majority of the web pages, is related to the contrast between the text and the background and foreground. Something that was already proven to be the most commonly detected problem in other study [WebAIM, 2021], in a 3-year analysis that started in 2019. The fact that 80% of the web pages have contrast problems, indicates that several developers do not give the necessary importance to avoid this basic mistake. Other problems related to the lack of accessible names of certain HTML elements like links and images are frequently detected. Simple aspects that could be easily addressed and, consequently, improve the accessibility of the web content.

With the objective of concluding what the impact the web technologies used in the web development have in web accessibility is, we analyze the results obtained from the accessibility assessment of 2,884,498 web pages as well as the web technologies identification of their respective domains. The main goal was to understand if web technologies influence web accessibility, in such a way that software development enterprises start concerning the accessibility of their contents and the application of certain tools and technologies that lead to better accessibility levels.

After performing the large-scale accessibility evaluation and technologies' identifications, we could

reveal that technologies of certain categories, such as Accessibility, Content Management Systems, JavaScript Frameworks, JavaScript Graphics, JavaScript Libraries, PaaS, Page Builders, Programming Languages, Rich Text Editors, LMS, Static Site Generator, UI Frameworks, Wikis, Web Frameworks, Multilingual and Online Forms may lead to a positive difference in the web accessibility, while others, like Forum Software or Comment Systems, lead to worse accessibility levels. These conclusions derived from the categories of technologies that are present in web pages whose accessibility score, obtained from the computation of the A3 metric, indicates an improvement in the accessibility level. When a website is being developed, there are categories of web technologies that are always needed in the first place. For instance, developers need to use programming languages, however, a technology from LMS category is not always needed, depending on the context of the website. If the website is not about online learning it does not need any platform like Moodle. Thus, the fact that a category of technologies proved to be present in web pages with better accessibility levels compared to others, does not make its usage mandatory.

When opting for a certain web category, it is important to know which technology leads to more accessible content. In this regard, and considering the web technologies of some of the categories that were identified in pages with more positive accessibility scores, it is possible to conclude the following for each category:

- Accessibility: this category employs web technologies that provide developers accessibility compliance with guidelines in their websites. AudioEye is the web technology that is present in more accessible web pages.
- Content Management Systems: in this category that helps developers create, edit, manage and publish web content, Joomla seems to have a more positive impact in accessibility.
- JavaScript Frameworks: MooTools is the web technology that showed less negative impact in the accessibility.
- JavaScript Graphics: this category that enables the creation of graphics has the Highcharts as the technology that was present in web pages that are more accessible.
- JavaScript Libraries: Isotope represents a JavaScript library that is used in more accessible content.
- Programming Languages: NodeJS and its supported languages, are considered to be the best programming languages used to lead to a more positive impact regarding web accessibility.
- UI Frameworks: ZURB Foundation framework has the best A3 metric scores, and so, it is identified in web pages that are more accessible.
- Web Frameworks: in this category, Ruby On Rails is present in more accessible content compared to the remaining web frameworks.
- Wikis: this category gathers several platforms that help in the creation of web content that can be edited collaboratively by any user. MediaWiki is considered to be the best when leading to more accessible pages.

Likewise, there are categories of technologies that tend to show a negative impact in the web accessibility, as they are present in web pages that demonstrated worse accessibility scores, according to A3 metric. Yet, if the scope of the web content needs some of these technologies, the following list presents the ones that were identified in those pages that had better accessibility levels:

- Advertising: although all ad systems are not recommended, DoubleClick is identified in less inaccessible content than the others.
- Comment Systems: the ability that users have to comment in a website is provided by the technologies of this category. In general, all of them were identified in several inaccessible web pages. Still, LiveFyre represents the best of the three considered web technologies.
- Maps: in this category, the Leaflet technology presented more positive accessibility scores in those pages where this technology was identified in.

According to these conclusions, developers should opt for certain web technologies to improve the accessibility of their web contents, even if the context of the web content that is being developed requires technologies that proved to be identified in more inaccessible content. Hence, the main intention should focus on choosing the web technologies or web categories that lead to better accessibility levels as well as consider good practices in the scope of the web content, in order to improve the interaction and access for disabled users.

While investigating other authors' works about the evaluation of the accessibility, we came across studies that relied only on home pages in their assessment scope. Since we did perform a large-scale accessibility evaluation of millions of web pages, the idea of comparing our results with other studies' results, in particular with WebAIM [2021] results, emerged. Therefore, it would be possible to verify the reliability of evaluating the home pages and not considering the remaining web pages. After the analysis, we could conclude that home pages can be a reference of the website, and so, the accessibility assessment of only home pages may conclude representative results. Nevertheless, home pages usually represent a more negative perspective about web accessibility.

After any web accessibility evaluation, its obtained results can be gathered and synthesized into a single value that represents the accessibility level of the assessed web resource. Thus, an overview of the web accessibility of a specific web page or website is achieved. To obtain the accessibility level of web resources, the web accessibility metrics are applied to the accessibility results. In this thesis, we compared eleven existing metrics by computing them over a dataset of more than two million web pages evaluated by the QualWeb automated accessibility evaluation tool. The main goal was to understand if these metrics correlate with each other. The examined web accessibility metrics included FR, A3, UWEM, WAQM, WIE, Conservative, Optimistic, Strict, WAB-PZ, WAB-H and WAEM. By analyzing the pairwise correlations we were able to identify groups of metrics. When considering the subset of metrics that are applicable at page level, we identified four clusters of distinct metrics. By looking at the full set of metrics applicable at site level, we identified a larger number of groups. This information is relevant when a decision between using one over other metrics is needed. By knowing that the outcomes of two metrics are similar, it becomes possible to choose the one that is less resource intensive, or from which it is easier to obtain the data required to compute the metric, for instance.

Additionally, we ran an experiment with a small number of web pages with known levels of accessibility to assess the validity of the different metrics. Even though the set of pages was small, and the metrics were computed from the outcomes of an automated tool (i.e., not all accessibility problems were caught), we were able to identify which metrics were consistent with the expected levels of accessibility of the pages, and which were not. This information can be also relevant to assist in choosing which metrics to employ.

This thesis focused predominantly on automatic evaluation tools, as, to perform the accessibility assessments and the web technologies identifications, we used QualWeb and Wappalyzer and SimilarTech, respectively. The fact that we did not resort to manual interventions implies that the detected accessibility issues may not be sufficient to report and explore all different problems the Web can have. Nevertheless, we performed a large-scale analysis, with 2.8 million web pages. As such, manual methodologies would be an unfeasible choice. The web accessibility metrics analysis is the only one for which this limitation did not impact much, since its main focus is to compare all metrics, regardless of if the assessment results do not cover all the accessibility aspects of the Web content.

In the future, it would be interesting to include accessibility evaluations performed by experts on specific websites, in order to mitigate the limitation resulting from the exclusive use of automated procedures. In this way, we can improve the amount of accessibility problems that are reported and, thus, draw more valid conclusions about the accessibility of the Web. It is also interesting to further investigate the web accessibility concerning the detected issues and the state of the accessibility.

References

Aasif. *Most Popular Web Development Frameworks for 2020*, 2020. URL https://www.appypie.com/top-web-development-frameworks?usource=lc&lctid=748278&lcid=1602782997_3_5498607. Accessed in 25 of October of 2021.

Julio Abascal, Myriam Arrue, and Xabier Valencia. Tools for web accessibility evaluation. In Yeliz Yesilada and Simon Harper, editors, *Web Accessibility: A Foundation for Research*, pages 479–503. Springer, London, 2019. ISBN 978-1-4471-7440-0. doi: 10.1007/978-1-4471-7440-0_26. URL https://doi.org/10.1007/978-1-4471-7440-0_26.

Hayfa Yousef Abuaddous, Mohd Zalisham Jali, and Nurlida Basir. Quantitative metric for ranking web accessibility barriers based on their severity. *Journal of Information and Communication Technology*, 16(1):81–102, 2017.

Simran Arora. *10 Best JavaScript Frameworks to Use in 2020*, 2020. URL https://hackr.io/blog/best-javascript-frameworks. Accessed in 25 of October of 2021.

J. Bailey and E. Burd. Tree-map visualisation for web accessibility. In *29th Annual International Computer Software and Applications Conference (COMPSAC'05)*, volume 1, pages 275–280 Vol. 2, 2005. doi: 10.1109/COMPSAC.2005.161.

John Bailey and Elizabeth Burd. Towards more mature web maintenance practices for accessibility. In *2007 9th IEEE International Workshop on Web Site Evolution*, pages 81–87, 2007. doi: 10.1109/WSE.2007.4380248.

Matteo Battistelli, Silvia Mirri, Ludovico Antonio Muratori, and Paola Salomoni. *Measuring accessibility barriers on large scale sets of pages*, 2011. URL https://www.w3.org/WAI/RD/2011/metrics/paper2/. Accessed in 25 of October of 2021.

Macy Bayern. *The 10 most popular container tools for businesses*, 2019. URL https://www.techrepublic.com/article/the-10-most-popular-container-tools-for-businesses/. Accessed in 25 of October of 2021.

Carlos Benavidez. *Libro blanco de eXaminator*. 2012.

Garenne Bigby. *WCAG: Web Content Accessibility Guidelines Explained*, January 2018. URL https://dynomapper.com/blog/27-accessibility-testing/273-wcag-the-web-content-accessibility-guidelines-explained. Accessed in 25 of October of 2021.

Giorgio Brajnik. Comparing accessibility evaluation tools: a method for tool effectiveness. *Universal access in the information society*, 3(3), August 2004. doi: 10.1007/s10209-004-0105-y. URL https://doi.org/10.1007/s10209-004-0105-y.

Giorgio Brajnik. *Barrier Walkthrough*, 2009. URL https://users.dimi.uniud.it/~giorgio.brajnik/projects/bw/bw.html. Accessed in 25 of October of 2021.

Giorgio Brajnik and Raffaella Lomuscio. Samba: A semi-automatic method for measuring barriers of accessibility. In *Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility*, Assets '07, page 43–50, New York, NY, USA, October 2007. Association for Computing Machinery. ISBN 9781595935731. doi: 10.1145/1296843.1296853. URL https://doi.org/10.1145/1296843.1296853.

Giorgio Brajnik and Markel Vigo. Automatic web accessibility metrics. In Yeliz Yesilada and Simon Harper, editors, *Web Accessibility: A Foundation for Research*, pages 505–521. Springer London, London, June 2019a. ISBN 978-1-4471-7440-0. doi: 10.1007/978-1-4471-7440-0_27. URL https://doi.org/10.1007/978-1-4471-7440-0_27.

Giorgio Brajnik and Markel Vigo. Automatic web accessibility metrics: Where we were and where we went. In *Web Accessibility*, pages 505–521, 06 2019b. ISBN 978-1-4471-7439-4. doi: 10.1007/978-1-4471-7440-0_27.

Giorgio Brajnik, Markel Vigo, and Joshue O Connor. *Research Report on Web Accessibility Metrics*, 2014. URL https://www.w3.org/WAI/RD/2011/metrics/note/ED-metrics. Accessed in 25 of October of 2021.

Christian Bühler, Helmut Heck, Olaf Perlick, Annika Nietzio, and Nils Ulltveit-Moe. Interpreting results from large scale automatic evaluation of web accessibility. In Klaus Miesenberger, Joachim Klaus, Wolfgang L. Zagler, and Arthur I. Karshmer, editors, *Computers Helping People with Special Needs*, pages 184–191, Berlin, Heidelberg, 2006. Springer. ISBN 978-3-540-36021-6. doi: 10.1007/11788713_28.

Docker. *Use containers to Build, Share and Run your applications*, 2021. URL https://www.docker.com/resources/what-container. Accessed in 25 of October of 2021.

Carlos Duarte, Inês Matos, João Vicente, Ana Salvado, Carlos M. Duarte, and Luís Carriço. Development technologies impact in web accessibility. In *Proceedings of the 13th Web for All Conference*, W4A '16, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450341387. doi: 10.1145/2899475.2899498. URL https://doi.org/10.1145/2899475.2899498.

DI FCUL. *Qualweb - Web Accessibility Evaluator*. URL http://qualweb.di.fc.ul.pt/evaluator/. Accessed in 25 of October of 2021.

Colin Flynn. *14 Technologies Every Web Developer Should Be Able to Explain*, January 2015. URL https://www.w3.org/WAI/fundamentals/accessibility-intro/. Accessed in 25 of October of 2021.

Tânia Frazão and Carlos Duarte. Comparing accessibility evaluation plug-ins. In *Proceedings of the 17th International Web for All Conference*, W4A '20, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450370561. doi: 10.1145/3371300.3383346. URL https://doi.org/10.1145/3371300.3383346.

André P. Freire, Thiago J. Bittar, and Renata P. M. Fortes. An approach based on metrics for monitoring web accessibility in brazilian municipalities web sites. In *Proceedings of the 2008 ACM Symposium on Applied Computing*, SAC '08, page 2421–2425, New York, NY, USA, 2008a. Association for Computing Machinery. ISBN 9781595937537. doi: 10.1145/1363686.1364259. URL https://doi.org/10.1145/1363686.1364259.

André P. Freire, Renata P. M. Fortes, Marcelo A. S. Turine, and Debora M. B. Paiva. An evaluation of web accessibility metrics based on their attributes. In *Proceedings of the 26th Annual ACM International Conference on Design of Communication*, SIGDOC '08, page 73–80, New York, NY, USA, 2008b. Association for Computing Machinery. ISBN 9781605580838. doi: 10.1145/1456536.1456551. URL https://doi.org/10.1145/1456536.1456551.

Andre P Freire, Christopher Power, Helen Petrie, Eduardo H Tanaka, Heloisa V Rocha, and Renata PM Fortes. Web accessibility metrics: Effects of different computational approaches. In *International Conference on Universal Access in Human-Computer Interaction*, pages 664–673. Springer, 2009. doi: 10.1007/978-3-642-02713-0_70.

Kentarou Fukuda, Shin Saito, Hironobu Takagi, and Chieko Asakawa. Proposing new metrics to evaluate web usability for the blind. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, page 1387–1390, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 1595930027. URL https://doi.org/10.1145/1056808.1056923.

Joshua Hardwick. *Find Out How Much Traffic a Website Gets: 3 Ways Compared*. URL https://ahrefs.com/blog/website-traffic/. Accessed in 25 of October of 2021.

Simon Harper and Yeliz Yesilada. *Web accessibility: a foundation for research*. Springer, 2008.

Shawn Lawton Henry. *Web Content Accessibility Guidelines (WCAG) Overview*, April 2021. URL https://www.w3.org/WAI/standards-guidelines/wcag/. Accessed in 25 of October of 2021.

Jhansi Karee. *How to choose a data model?*, 2020. URL https://medium.com/@jhansiredy007/how-to-choose-a-data-model-c32cbfc1d1a7. Accessed in 25 of October of 2021.

Royce Kimmons. Open to all? nationwide evaluation of high-priority web accessibility considerations among higher education websites. *Journal of Computing in Higher Education*, May 2017. doi: 10.1007/s12528-017-9151-3. URL https://doi.org/10.1007/s12528-017-9151-3.

Chandan Kumar. *How to find What Technology Website using?*, July 2020. URL https://geekflare.com/what-technology-website-using/. Accessed in 25 of October of 2021.

Zui Young Lim, Jia Min Chua, Kaiting Yang, Wei Shin Tan, and Yinn Chai. Web accessibility testing for singapore government e-services. In *Proceedings of the 17th International Web for All Conference*, W4A '20, New York, NY, USA, April 2020. Association for Computing Machinery. ISBN 9781450370561. doi: 10.1145/3371300.3383353. URL https://doi.org/10.1145/3371300.3383353.

Rui Lopes and Luís Carriço. The impact of accessibility assessment in macro scale universal usability studies of the web. In *Proceedings of the 2008 International Cross-Disciplinary Conference on Web Accessibility (W4A)*, page 5–14, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605581538. URL https://doi.org/10.1145/1368044.1368048.

Rui Lopes, Daniel Gomes, and Luís 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, New York, NY, USA, April 2010. Association for Computing Machinery. ISBN 9781450300452. doi: 10.1145/1805986.1806001. URL https://doi.org/10.1145/1805986.1806001.

Ana Belén Martínez, Aquilino A. Juan, Darío Álvarez, and Ma del Carmen Suárez. Wab*: A quantitative metric based on wab. In Martin Gaedke, Michael Grossniklaus, and Oscar Díaz, editors, *Web Engineering*, pages 485–488, Berlin, Heidelberg, 2009. Springer. ISBN 978-3-642-02818-2. doi: 10.1007/978-3-642-02818-2_44.

MongoDB. *When to Use a NoSQL Database*, 2021. URL https://www.mongodb.com/nosql-explained/when-to-use-nosql. Accessed in 25 of October of 2021.

Justyna Mucha, Mikael Snaprud, and Annika Nietzio. Web page clustering for more efficient website accessibility evaluations. In Klaus Miesenberger, Christian Bühler, and Petr Penaz, editors, *Computers Helping People with Special Needs*, pages 259–266, Cham, July 2016. Springer International Publishing. ISBN 978-3-319-41264-1. doi: 10.1007/978-3-319-41264-1_35.

Marian Padure and Costin Pribeanu. Exploring the differences between five accessibility evaluation tools. In Alin Moldoveanu and Alan J. Dix, editors, *16th International Conference on Human-Computer Interaction, RoCHI 2019, Bucharest, Romania, October 17-18, 2019*, pages 87–90. Matrix Rom, 2019.

Bambang Parmanto and Xiaoming Zeng. Metric for web accessibility evaluation. *Journal of the American Society for Information Science and Technology*, 56(13):1394–1404, 2005.

Kalpesh Patel. *Top 5 Online Tools That Identify technology on websites*, June 2019. URL https://www.webknowledgefree.com/top-5-online-tools-that-identify-technology-on-websites/. Accessed in 25 of October of 2021.

Pei-Luen Patrick Rau, Lianhui Zhou, Na Sun, and Runting Zhong. Evaluation of web accessibility in china: changes from 2009 to 2013. *Universal Access in the Information Society*, September 2014. doi: 10.1007/s10209-014-0385-9. URL https://doi.org/10.1007/s10209-014-0385-9.

Sasha Reeves. *Web Technologies: A Journey From HTML To Web 3.0*, 2019. URL https://www.goodcore.co.uk/blog/web-technologies/. Accessed in 25 of October of 2021.

Anne Spencer Ross. An epidemiology-inspired, large-scale analysis of mobile app accessibility. *SIGACCESS Access. Comput.*, (123), March 2020. ISSN 1558-2337. doi: 10.1145/3386402.3386408. URL https://doi.org/10.1145/3386402.3386408.

Lauren Schaefer. *NoSQL vs SQL Databases*, 2021. URL https://www.mongodb.com/nosql-explained/nosql-vs-sql. Accessed in 25 of October of 2021.

Serpstat. *How to detect which CMS a website is using: 8 easy ways*, October 2019. URL https://serpstat.com/blog/how-to-detect-which-cms-a-website-is-using-8-easy-ways/. Accessed in 25 of October of 2021.

Pornpat Sirithumgul, Atiwong Suchato, and Proadpran Punyabukkana. Quantitative evaluation for web accessibility with respect to disabled groups. In *Proceedings of the 2009 International Cross-Disciplinary Conference on Web Accessibililty (W4A)*, W4A '09, page 136–141, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585611. doi: 10.1145/1535654.1535687. URL https://doi.org/10.1145/1535654.1535687.

Mikael Snaprud and Agata Sawicka. Large scale web accessibility evaluation - a european perspective. In Constantine Stephanidis, editor, *Universal Access in Human-Computer Interaction. Applications and Services*, pages 150–159, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-73283-9. doi: 10.1007/978-3-540-73283-9_18.

Mikael Holmesland Snaprud, Nils Ulltveit-Moe, Anand Balachandran Pillai, and Morten Goodwin Olsen. A proposed architecture for large scale web accessibility assessment. In Klaus Miesenberger, Joachim Klaus, Wolfgang L. Zagler, and Arthur I. Karshmer, editors, *Computers Helping People with Special Needs*, pages 234–241, Berlin, Heidelberg, 2006. Springer. ISBN 978-3-540-36021-6. doi: 10.1007/11788713_35.

Shuyi Song, Can Wang, Liangcheng Li, Zhi Yu, Xiao Lin, and Jiajun Bu. Waem: A web accessibility evaluation metric based on partial user experience order. In *Proceedings of the 14th Web for All Conference on The Future of Accessible Work*, W4A '17, New York, NY, USA, April 2017. Association for Computing Machinery. ISBN 9781450349000. doi: 10.1145/3058555.3058576. URL https://doi.org/10.1145/3058555.3058576.

Shuyi Song, Jiajun Bu, Chengchao Shen, Andreas Artmeier, Zhi Yu, and Qin Zhou. Reliability aware web accessibility experience metric. In *Proceedings of the 15th International Web for All Conference*, W4A '18, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356510. doi: 10.1145/3192714.3192836. URL https://doi.org/10.1145/3192714.3192836.

Statstutor. *Spearman's correlation*, 2021. URL https://www.statstutor.ac.uk/resources/uploaded/spearmans.pdf. Accessed in 25 of October of 2021.

Terry Sullivan and Rebecca Matson. Barriers to use: Usability and content accessibility on the web's most popular sites. In *Proceedings on the 2000 Conference on Universal Usability*, CUU '00, page 139–144, New York, NY, USA, 2000. Association for Computing Machinery. ISBN 1581133146. doi: 10.1145/355460.355549. URL https://doi.org/10.1145/355460.355549.

Milos Timotic. *9 Web Technologies Every Web Developer Must Know in 2020*, October 2018. URL https://tms-outsource.com/blog/posts/web-technologies/. Accessed in 25 of October of 2021.

Markel Vigo and Giorgio Brajnik. Automatic web accessibility metrics: Where we are and where we can go. *Interacting with Computers*, 23(2):137–155, 01 2011. ISSN 0953-5438. doi: 10.1016/j.intcom.2011.01.001. URL https://doi.org/10.1016/j.intcom.2011.01.001.

Markel Vigo, Myriam Arrue, Giorgio Brajnik, Raffaella Lomuscio, and Julio Abascal. Quantitative metrics for measuring web accessibility. In *Proceedings of the 2007 International Cross-Disciplinary Conference on Web Accessibility (W4A)*, page 99–107, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 1595935908. URL https://doi.org/10.1145/1243441.1243465.

Markel Vigo, Giorgio Brajnik, Myriam Arrue, and Julio Abascal. Tool independence for the web accessibility quantitative metric. *Disability and Rehabilitation: Assistive Technology*, 4(4):248–263, October 2009. doi: 10.1080/17483100902903291. URL https://doi.org/10.1080/17483100902903291.

Markel Vigo, Julio Abascal, Amaia Aizpurua, and Myriam Arrue. *Attaining metric validity and reliability with the web accessibility quantitative metric*, 2011. URL https://www.w3.org/WAI/RD/2011/metrics/paper6/. Accessed in 25 of October of 2021.

Markel Vigo, Giorgio Brajnik, and Joshue O Connor. Research report on web accessibility metrics. In Markel Vigo, Giorgio Brajnik, and Joshue O Connor eds., editors, *W3C WAI Symposium on Website Accessibility Metrics*, W3C WAI Research and Development Working Group (RDWG) Notes. W3C Web Accessibility Initiative (WAI), first public working draft edition, August 2012. URL http://www.w3.org/TR/accessibility-metrics-report.

Markel Vigo, Justin Brown, and Vivienne 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*, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450318440. URL https://doi.org/10.1145/2461121.2461124.

Vishwasraj. *How to detect which CMS a website is using: 8 easy ways*, July 2018. URL https://www.quora.com/Is-it-possible-to-find-which-front-end-framework-a-website-uses. Accessed in 25 of October of 2021.

W3C. *ARIA Landmarks Example*, a. URL https://www.w3.org/TR/wai-aria-practices-1.1/examples/landmarks/main.html. Accessed in 25 of October of 2021.

W3C. *Evaluating Web Accessibility Overview*, b. URL https://www.w3.org/WAI/test-evaluate/. Accessed in 25 of October of 2021.

W3C. *Introduction to Web Accessibility*, February 2005. URL https://www.w3.org/WAI/fundamentals/accessibility-intro/. Accessed in 25 of October of 2021.

W3C. *Web Content Accessibility Guidelines (WCAG) 2.0*, June 2010. URL https://www.w3.org/WAI/GL/WCAG20/. Accessed in 25 of October of 2021.

W3C. *Understanding SC 4.1.1*, 2016. URL https://www.w3.org/TR/UNDERSTANDING-WCAG20/ensure-compat-parses.html. Accessed in 25 of October of 2021.

W3C. *Accessibility Conformance Testing (ACT) Rules Format 1.0*, October 2019. URL https://www.w3.org/TR/act-rules-format/. Accessed in 25 of October of 2021.

W3Techs. *Usage statistics of content management systems*. URL https://w3techs.com/technologies/overview/content_management. Accessed in 25 of October of 2021.

WebAIM. *WAVE Web Accessibility Evaluation Tool*. URL https://wave.webaim.org/. Accessed in 25 of October of 2021.

WebAIM. *The WebAIM Million*, 2021. URL https://webaim.org/projects/million/. Accessed in 25 of October of 2021.

Wikipedia. *Spider trap*, a. URL https://en.wikipedia.org/wiki/Spider_trap. Accessed in 25 of October of 2021.

Wikipedia. *WebAIM*, b. URL https://pt.wikipedia.org/wiki/WebAIM. Accessed in 25 of October of 2021.

W&L. *Evaluation Tools*. URL https://my.wlu.edu/disability-accommodations/web-accessibility/evaluation-tools. Accessed in 25 of October of 2021.

Appendix A

Number of errors by each ACT-Rule and respective percentage of pages that failed each ACT-Rule

Table A.1: Number of errors and percentage of web pages that failed each ACT-Rule

ACT-Rule	**Description**	**Number of errors**	**Percentage of web pages**
ACT-Rule R76	Text has enhanced contrast	33109298	79.33%
ACT-Rule R37	Text has minimum contrast	18322594	65.60%
ACT-Rule R12	Link has accessible name	9026352	51.73%
ACT-Rule R18	id attribute value is unique	6569778	30.53%
ACT-Rule R17	Image has accessible name	6274421	30.12%
ACT-Rule R16	Form control has accessible name	1878778	22.26%
ACT-Rule R19	iframe element has accessible name	1207887	18.80%
ACT-Rule R48	Element marked as decorative is not exposed	792267	6.69%
ACT-Rule R14	meta viewport does not prevent zoom	658061	21.95%
ACT-Rule R20	role attribute has valid value	630281	5.57%
ACT-Rule R21	svg element with explicit role has accessible name	616250	3.91%
ACT-Rule R13	Element with aria-hidden has no focusable content	582953	6.22%
ACT-Rule R11	Button has accessible name	540438	7.39%

Table A.2: Number of errors and percentage of web pages that failed each ACT-Rule - continuation

ACT-Rule	Description	Number of errors	Percentage of web pages
ACT-Rule R2	HTML has lang attribute	518204	17.97%
ACT-Rule R30	Visible label is part of accessible name	451589	3.80%
ACT-Rule R35	Heading has accessible name	447579	8.48%
ACT-Rule R25	ARIA state or property is permitted	444805	5.63%
ACT-Rule R33	ARIA required context role	350098	2.03%
ACT-Rule R65	Element with presentational children has no focusable content	211593	2.20%
ACT-Rule R38	ARIA required owned elements	199775	4.11%
ACT-Rule R68	Line height in style attributes is not !important	102049	0.69%
ACT-Rule R39	All table header cells have assigned data cells	78378	0.73%
ACT-Rule R1	HTML Page has a title	76691	2.66%
ACT-Rule R28	Element with role attribute has required states and properties	71186	1.09%
ACT-Rule R71	meta element has no refresh delay (no exception)	67166	2.33%
ACT-Rule R4	Meta-refresh no delay	66899	2.32%
ACT-Rule R70	frame with negative tabindex has no interactive elements	46276	1.31%
ACT-Rule R27	This rule checks that each aria- attribute specified is defined in ARIA 1.1.	41084	0.36%

Table A.3: Number of errors and percentage of web pages that failed each ACT-Rule - continuation

ACT-Rule	Description	Number of errors	Percentage of web pages
ACT-Rule R43	Scrollable element is keyboard accessible	38337	0.98%
ACT-Rule R6	Image button has accessible name	34804	0.92%
ACT-Rule R66	Menuitem has non-empty accessible name	31531	0.40%
ACT-Rule R36	Headers attribute specified on a cell refers to cells in the same table element	28476	0.14%
ACT-Rule R24	autocomplete attribute has valid value	20489	0.43%
ACT-Rule R5	Validity of HTML Lang attribute	16836	0.55%
ACT-Rule R34	ARIA state or property has valid value	15830	0.27%
ACT-Rule R22	Element within body has valid lang attribute	14163	0.08%
ACT-Rule R42	Object element has non-empty accessible name	9175	0.20%
ACT-Rule R67	Letter spacing in style attributes is not !important	3494	0.05%
ACT-Rule R69	Word spacing in style attributes is not !important	582	0.01%
ACT-Rule R3	HTML lang and xml:lang match	576	0.02%
ACT-Rule R7	Orientation of the page is not restricted using CSS transform property	262	0.01%

Appendix B

Number of errors by each ACT-Rule and respective percentage of pages that failed each ACT-Rule for home pages

Table B.1: Number of failures and percentage of web pages by each ACT-Rule for home pages

ACT-Rule	**Description**	**Number of failures**	**Percentage of web pages**
ACT-Rule R76	Text has enhanced contrast	610719	78.630%
ACT-Rule R37	Text has minimum contrast	349389	65.121%
ACT-Rule R12	Link has accessible name	216861	56.690%
ACT-Rule R17	Image has accessible name	126109	30.978%
ACT-Rule R18	id attribute value is unique	103772	30.104%
ACT-Rule R16	Form control has accessible name	21664	17.021%
ACT-Rule R48	Element marked as decorative is not exposed	18965	7.619%
ACT-Rule R13	Element with aria-hidden has no focusable content	18815	7.882%
ACT-Rule R19	iframe element has accessible name	17140	17.147%
ACT-Rule R20	role attribute has valid value	13743	5.624%
ACT-Rule R35	Heading has accessible name	12956	11.906%
ACT-Rule R14	meta viewport does not prevent zoom	12297	22.814%
ACT-Rule R11	Button has accessible name	11939	6.130%
ACT-Rule R2	HTML has lang attribute	8369	15.932%
ACT-Rule R21	svg element with explicit role has accessible name	6988	2.692%
ACT-Rule R33	ARIA required context role	6549	1.748%
ACT-Rule R30	Visible label is part of accessible name	6484	3.351%
ACT-Rule R25	ARIA state or property is permitted	5969	5.054%
ACT-Rule R65	Element with presentational children has no focusable content	5440	2.454%
ACT-Rule R38	ARIA required owned elements	4636	5.039%
ACT-Rule R71	meta element has no refresh delay (no exception)	2431	4.628%
ACT-Rule R4	Meta-refresh no delay	2426	4.618%
ACT-Rule R68	Line height in style attributes is not !important	1259	0.674%
ACT-Rule R70	frame with negative tabindex has no interactive elements	1214	1.911%
ACT-Rule R39	All table header cells have assigned data cells	1110	0.362%
ACT-Rule R1	HTML Page has a title	1100	2.094%
ACT-Rule R43	Scrollable element is keyboard accessible	711	0.944%
ACT-Rule R28	Element with role attribute has required states and properties	634	0.607%
ACT-Rule R24	autocomplete attribute has valid value	464	0.590%
ACT-Rule R6	Image button has accessible name	452	0.636%
ACT-Rule R27	This rule checks that each aria- attribute specified is defined in ARIA 1.1.	330	0.228%
ACT-Rule R66	Menuitem has non-empty accessible name	312	0.232%
ACT-Rule R5	Validity of HTML Lang attribute	256	0.451%
ACT-Rule R42	Object element has non-empty accessible name	245	0.209%
ACT-Rule R22	Element within body has valid lang attribute	231	0.053%
ACT-Rule R67	Letter spacing in style attributes is not !important	130	0.086%
ACT-Rule R34	ARIA state or property has valid value	124	0.110%
ACT-Rule R69	Word spacing in style attributes is not !important	41	0.013%
ACT-Rule R36	Headers attribute specified on a cell refers to cells in the same table element	14	0.002%
ACT-Rule R7	Orientation of the page is not restricted using CSS transform property	9	0.013%
ACT-Rule R3	HTML lang and xml:lang match	4	0.008%

Appendix C

Number of errors by each ACT-Rule and respective percentage of pages that failed each ACT-Rule for interior pages

Table C.1: Number of failures and percentage of web pages by each ACT-Rule for interior pages

ACT-Rule	**Description**	**Number of failures**	**Percentage of web pages**
ACT-Rule R76	Text has enhanced contrast	515969	74.770%
ACT-Rule R37	Text has minimum contrast	287621	60.733%
ACT-Rule R12	Link has accessible name	135404	47.923%
ACT-Rule R18	id attribute value is unique	99171	26.628%
ACT-Rule R17	Image has accessible name	89635	26.670%
ACT-Rule R16	Form control has accessible name	27154	17.829%
ACT-Rule R19	iframe element has accessible name	15188	14.760%
ACT-Rule R14	meta viewport does not prevent zoom	11911	22.015%
ACT-Rule R48	Element marked as decorative is not exposed	11375	6.427%
ACT-Rule R13	Element with aria-hidden has no focusable content	10152	5.384%
ACT-Rule R2	HTML has lang attribute	9578	18.234%
ACT-Rule R20	role attribute has valid value	8926	5.494%
ACT-Rule R11	Button has accessible name	8924	5.738%
ACT-Rule R21	svg element with explicit role has accessible name	8703	3.206%
ACT-Rule R35	Heading has accessible name	8177	8.072%
ACT-Rule R30	Visible label is part of accessible name	6042	3.246%
ACT-Rule R33	ARIA required context role	5907	1.628%
ACT-Rule R25	ARIA state or property is permitted	5296	4.434%
ACT-Rule R65	Element with presentational children has no focusable content	3120	1.748%
ACT-Rule R38	ARIA required owned elements	3022	3.330%
ACT-Rule R71	meta element has no refresh delay (no exception)	1824	3.472%
ACT-Rule R4	Meta-refresh no delay	1819	3.463%
ACT-Rule R1	HTML Page has a title	1410	2.684%
ACT-Rule R39	All table header cells have assigned data cells	1346	0.476%
ACT-Rule R68	Line height in style attributes is not !important	1137	0.619%
ACT-Rule R70	frame with negative tabindex has no interactive elements	930	1.472%
ACT-Rule R28	Element with role attribute has required states and properties	837	0.699%
ACT-Rule R43	Scrollable element is keyboard accessible	665	0.860%
ACT-Rule R27	This rule checks that each aria- attribute specified is defined in ARIA 1.1.	650	0.247%
ACT-Rule R66	Menuitem has non-empty accessible name	617	0.307%
ACT-Rule R6	Image button has accessible name	440	0.676%
ACT-Rule R24	autocomplete attribute has valid value	433	0.504%
ACT-Rule R5	Validity of HTML Lang attribute	250	0.459%
ACT-Rule R22	Element within body has valid lang attribute	200	0.057%
ACT-Rule R34	ARIA state or property has valid value	199	0.175%
ACT-Rule R42	Object element has non-empty accessible name	132	0.158%
ACT-Rule R67	Letter spacing in style attributes is not !important	53	0.038%
ACT-Rule R3	HTML lang and xml:lang match	9	0.017%
ACT-Rule R36	Headers attribute specified on a cell refers to cells in the same table element	5	0.002%
ACT-Rule R7	Orientation of the page is not restricted using CSS transform property	3	0.006%
ACT-Rule R69	Word spacing in style attributes is not !important	1	0.002%

Appendix D

Dunn's test results

Table D.1: Dunn-test ρ-value scores with Bonferroni adjustment for Accessibility category

	AccessiBe	**AudioEye**	**EqualWeb**	**UserWay**
AccessiBe	1.00			
AudioEye	<0.001	1.00		
EqualWeb	<0.001	<0.001	1.00	
UserWay	<0.001	<0.001	<0.001	1.00

Table D.2: Dunn-test ρ-value scores with Bonferroni adjustment for Advertising category

	AppNexus	**DoubleClick**	**Facebook Advertiser**	**Google AdWords Advertiser**	**Google Adsense**	**Twitter Ads**
AppNexus	1.00					
DoubleClick	0.0013	1.00				
Facebook Advertiser	1.00	0.12	1.00			
Google AdWords Advertiser	<0.001	<0.001	0.034	1.00		
Google Adsense	0.0064	<0.001	1.00	<0.001	1.00	
Twitter Ads	1.00	<0.001	1.00	<0.001	<0.001	1.00

Table D.3: Dunn-test ρ-value scores with Bonferroni adjustment for Content Management Systems category

	Drupal	**Elementor**	**Jimdo**	**Joomla**	**TYPO3**	**Wix**	**WordPress**
Drupal	1.00						
Elementor	0.037	1.00					
Jimdo	<0.001	1.00	1.00				
Joomla	<0.001	0.006	<0.001	1.00			
TYPO3	<0.001	0.27	<0.001	<0.001	1.00		
Wix	1.00	0.036	<0.001	<0.001	<0.001	1.00	
WordPress	<0.001	0.17	<0.001	<0.001	<0.001	<0.001	1.00

Table D.4: Dunn-test ρ-value scores with Bonferroni adjustment for Comment Systems category

	Disqus	**Facebook Comments**	**LiveFyre**
Disqus	1.00		
Facebook Comments	<0.001	1.00	
LiveFyre	<0.001	<0.001	1.00

Table D.5: Dunn-test ρ-value scores with Bonferroni adjustment for JavaScript Frameworks category

	AMP	**Angular JS**	**Backbone.js**	**GSAP**	**Handlebars**	**MooTools**	**Mustache JS**	**Prototype**	**React**	**Require JS**	**Stimulus**	**VueJS**
AMP	1.00											
Angular JS	<0.001	1.00										
Backbone.js	<0.001	<0.001	1.00									
GSAP	<0.001	<0.001	<0.001	1.00								
Handlebars	<0.001	<0.001	<0.001	<0.001	1.00							
MooTools	<0.001	<0.001	<0.001	<0.001	<0.001	1.00						
Mustache JS	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	1.00					
Prototype	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	1.00				
React	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	1.00			
Require JS	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	1.00		
Stimulus	1.00	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	1.00	
VueJS	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	1.00

Table D.6: Dunn-test ρ-value scores with Bonferroni adjustment for JavaScript Graphics category

	Chart.js	**D3**	**Highcharts**	**MathJax**	**Raphael**	**Supersized**	**particles.js**	**three.js**
Chart.js	1.00							
D3	<0.001	1.00						
Highcharts	<0.001	<0.001	1.00					
MathJax	<0.001	<0.001	<0.001	1.00				
Raphael	<0.001	<0.001	<0.001	<0.001	1.00			
Supersized	<0.001	1.00	0.67	<0.001	<0.001	1.00		
particles.js	<0.001	<0.001	<0.001	1.00	<0.001	<0.001	1.00	
three.js	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	1.00

Table D.7: Dunn-test ρ-value scores with Bonferroni adjustment for JavaScript Libraries category

	Hammer.js	**Isotope**	**LightBox**	**Lodash**	**Modernizr**	**Moment JS**	**Polyfill**	**Slick**	**jQuery**	**jQuery Migrate**	**jQuery UI**	**prettyPhoto**
Hammer.js	1.00											
Isotope	<0.001	1.00										
LightBox	1.00	<0.001	1.00									
Lodash	0.97	<0.001	0.013	1.00								
Modernizr	<0.001	0.008	<0.001	<0.001	1.00							
Moment JS	1.00	<0.001	1.00	1.00	0.24	1.00						
Polyfill	<0.001	<0.001	<0.001	<0.001	<0.001	0.177	1.00					
Slick	<0.001	1.00	<0.001	<0.001	<0.001	<0.001	<0.001	1.00				
jQuery	1.00	<0.001	1.00	<0.001	<0.001	1.00	<0.001	<0.001	1.00			
jQuery Migrate	1.00	<0.001	1.00	1.00	1.00	1.00	1.00	<0.001	1.00	1.00		
jQuery UI	<0.001	0.002	<0.001	<0.001	0.1	1.00	<0.001	<0.001	<0.001	1.00	1.00	
prettyPhoto	1.00	<0.001	1.00	<0.001	<0.001	1.00	<0.001	<0.001	1.00	1.00	<0.001	1.00

Table D.8: Dunn-test ρ-value scores with Bonferroni adjustment for Maps category

	Google Maps	**Leaflet**	**Mapbox GL JS**
Google Maps	1.00		
Leaflet	<0.001	1.00	
Mapbox GL JS	1.00	<0.001	1.00

Table D.9: Dunn-test ρ-value scores with Bonferroni adjustment for Programming Languages category

	Java	**Lua**	**NodeJs**	**PHP**	**Python**	**Ruby**
Java	1.00					
Lua	<0.001	1.00				
NodeJs	<0.001	<0.001	1.00			
PHP	<0.001	<0.001	<0.001	1.00		
Python	<0.001	<0.001	<0.001	<0.001	1.00	
Ruby	<0.001	<0.001	<0.001	0.69	<0.001	1.00

Table D.10: Dunn-test ρ-value scores with Bonferroni adjustment for UI Frameworks category

	Bootstrap	**ZURB Foundation**	**animate.css**
Bootstrap	1.00		
ZURB Foundation	<0.001	1.00	
animate.css	<0.001	<0.001	1.00

Table D.11: Dunn-test ρ-value scores with Bonferroni adjustment for Video Players category

	JW Player	**MediaElement.js**	**Plyr**	**VideoJS**	**Vimeo**	**YouTube**
JW Player	1.00					
MediaElement.js	<0.001	1.00				
Plyr	<0.001	<0.001	1.00			
VideoJS	0.21	<0.001	<0.001	1.00		
Vimeo	<0.001	<0.001	<0.001	<0.001	1.00	
YouTube	<0.001	<0.001	<0.001	<0.001	<0.001	1.00

Table D.12: Dunn-test ρ-value scores with Bonferroni adjustment for Web Frameworks category

	CodeIgniter	**Django**	**Express**	**Laravel**	**Microsoft ASP.NET**	**Ruby On Rails**	**Symfony**
CodeIgniter	1.00						
Django	0.0012	1.00					
Express	0.24	1.00	1.00				
Laravel	1.00	<0.001	0.0077	1.00			
Microsoft ASP.NET	1.00	0.0077	1.00	0.0049	1.00		
Ruby On Rails	<0.001	<0.001	<0.001	<0.001	<0.001	1.00	
Symfony	<0.001	1.00	0.42	<0.001	<0.001	<0.001	1.00

Table D.13: Dunn-test ρ-value scores with Bonferroni adjustment for Wikis category

	Atlassian Confluence	**DokuWiki**	**Foswiki**	**MediaWiki**	**MoinMoin**
Atlassian Confluence	1.00				
DokuWiki	<0.001	1.00			
Foswiki	<0.001	<0.001	1.00		
MediaWiki	<0.001	<0.001	<0.001	1.00	
MoinMoin	<0.001	<0.001	<0.001	<0.001	1.00