REAL-TIME ACCESSIBILITY OPTIMIZATION: ENHANCING WEB ACCESSIBILITY THROUGH ON-DEVICE MACHINE LEARNING AND AUTOMATED ADAPTATION

Sivaramarajalu Ramadurai Venkataraajalu
Amazon, New York, United States

Abstract: This paper presents a novel approach to enhance web accessibility through real-time optimization using on-device machine learning and automated adaptation. The proposed system utilizes a browser extension that integrates machine learning models to analyze web pages in real-time, identify accessibility issues, and generate appropriate adaptations to improve the user experience. By performing the analysis and adaptation directly on the user’s device, the system ensures a seamless and personalized accessibility experience. The paper details the architecture of the proposed system, including the accessibility analysis module, adaptation engine, on-device machine learning models, and user interface. The accessibility analysis process employs techniques such as computer vision, natural language processing, and pattern recognition to identify issues related to alt text, color contrast, heading structure, and keyboard accessibility. The adaptation engine generates dynamic modifications to the web page to address the identified issues and enhance accessibility. The evaluation framework, consisting of user studies, performance metrics, and comparative analyses, assesses the effectiveness and usability of the proposed approach. This research contributes to the advancement of web accessibility solutions and promotes a more inclusive digital environment for individuals with disabilities.

Index Terms – On-Device Machine Learning, Web Accessibility, Frontend Development, Browser Extension, Real-time Optimization, W3C Standards

I. INTRODUCTION

The rapid growth of the internet and digital technologies has revolutionized the way people access information, communicate, and participate in various aspects of life. However, this digital transformation has also highlighted the challenges faced by individuals with disabilities in accessing and utilizing web content effectively. Web accessibility refers to the practice of designing and developing websites, tools, and technologies that can be used by people with a wide range of abilities and disabilities, including visual, auditory, motor, and cognitive impairments [1].

The World Health Organization estimates that over one billion people worldwide live with some form of disability [2]. Failing to provide accessible web content not only excludes a significant portion of the population but also violates their fundamental rights to access information and participate in the digital society [3]. Recognizing the importance of web accessibility, various guidelines and standards have been developed to assist web designers and developers in creating accessible content. The most widely recognized and adopted guidelines are the Web Content Accessibility Guidelines (WCAG), developed by the World Wide Web Consortium (W3C) [4]. WCAG provides a comprehensive set of principles, guidelines, and success criteria for ensuring that web content is perceivable, operable, understandable, and robust for users with disabilities.

Despite the existence of accessibility guidelines and the increasing awareness of their importance, many websites still fail to meet the necessary accessibility standards [5]. This can be attributed to various factors,
such as lack of awareness, limited resources, complexity of implementation, and the ever-evolving nature of web technologies [6]. Consequently, users with disabilities often encounter barriers and challenges when attempting to access and interact with web content, leading to frustration, exclusion, and unequal opportunities.

To address these challenges, there is a growing need for innovative solutions that can automate and optimize web accessibility in real-time. Recent advancements in machine learning and artificial intelligence have opened up new possibilities for developing intelligent systems that can analyze web content, identify accessibility issues, and provide automated adaptations to enhance the user experience for individuals with disabilities [7].

In this paper, I propose a novel approach to enhancing web accessibility through real-time optimization using on-device machine learning and automated adaptation. My proposed system leverages the power of machine learning models deployed directly within a browser extension to analyze web pages in real-time, identify accessibility issues, and generate appropriate adaptations to improve the user experience. By bringing the intelligence and processing capabilities directly to the user's device, my system aims to provide a seamless and personalized accessibility experience, tailored to the specific needs and preferences of each individual user.

II. LITERATURE REVIEW

Frontend Web accessibility has been a topic of extensive research and development in recent years, with efforts focused on understanding accessibility barriers, developing guidelines and standards, and creating tools and technologies to support accessible design and evaluation.

2.1 Web Accessibility Guidelines and Standards

The Web Content Accessibility Guidelines (WCAG) are the most widely recognized and adopted guidelines for web accessibility [4]. Developed by the World Wide Web Consortium (W3C), WCAG provides a comprehensive set of principles, guidelines, and success criteria for ensuring that web content is perceivable, operable, understandable, and robust for users with disabilities. The latest version, WCAG 2.1, covers a wide range of accessibility considerations, including text alternatives, keyboard accessibility, color contrast, and compatibility with assistive technologies [8].

Several studies have investigated the adoption and conformance of websites to WCAG guidelines. Lazar et al. [9] conducted a study to understand webmasters' perceptions and practices regarding web accessibility. They found that while there was a general awareness of accessibility guidelines, many websites still failed to meet the necessary standards due to various challenges, such as lack of resources, time constraints, and the complexity of implementation.

Aizpurua et al. [10] explored the factors influencing the experienced accessibility of websites by users with disabilities. Their study highlighted the role of user expectations, memories, and confidence in shaping the perceived accessibility of web content. The findings emphasized the importance of considering user perspectives and experiences in accessibility evaluation and design.

2.2 Automated Accessibility Evaluation

Tools To support the evaluation and improvement of web accessibility, various automated tools and frameworks have been developed. These tools analyze web pages and identify potential accessibility issues based on predefined rules and heuristics derived from accessibility guidelines.

WAVE [11] is a widely used web accessibility evaluation tool that provides visual feedback on accessibility issues within a web page. It highlights elements with accessibility problems, such as missing alternative text, poor color contrast, and incorrect heading structures. Similarly, aXe [12] is an open-source accessibility testing engine that can be integrated into web development workflows to identify and report accessibility violations.

Lighthouse [13] is another popular tool that combines accessibility evaluation with performance and best practice audits. It provides a comprehensive report on the accessibility status of a web page, along with suggestions for improvement. These automated tools have been valuable in identifying common accessibility issues and guiding developers in creating more accessible websites.

However, automated accessibility evaluation tools have limitations. Vigo et al. [14] conducted a benchmarking study to assess the effectiveness of automated accessibility testing tools. They found that while these tools can identify certain accessibility issues, they may miss complex or context-specific barriers that require human judgment. The study emphasized the need for a combination of automated and manual evaluation methods to ensure comprehensive accessibility coverage.

2.3 Machine Learning Approaches to Web Accessibility

Machine learning techniques have been explored to enhance accessibility evaluation and adaptation in recent years. These approaches aim to leverage the power of data-driven models to automatically identify accessibility issues and provide personalized adaptations.

Ferreira et al. [15] proposed a machine learning approach to predict the accessibility of web pages based on a set of features extracted from the HTML structure and content. They trained a classifier using a dataset of manually annotated web pages and achieved promising results in identifying accessibility issues and suggesting improvements. The study demonstrated the potential of machine learning in automating accessibility evaluation and providing insights for remediation.

Akpinar et al. [16] developed a vision-based page segmentation algorithm to detect and localize accessibility issues within web pages. Their approach utilized visual and structural features to identify elements with accessibility problems, such as low contrast text and small font sizes. The algorithm achieved high accuracy in detecting and localizing accessibility issues, showing the effectiveness of machine learning in analyzing visual aspects of web pages.

2.4 User-Centered Approaches to Web Accessibility

User-centered approaches have been employed to understand and address the accessibility needs of individuals with disabilities. These approaches involve directly engaging users with disabilities in the design, evaluation, and improvement of web accessibility.
Breslin et al. [17] conducted a study involving users with disabilities to identify common accessibility barriers and gather insights for designing more inclusive websites. They emphasized the importance of involving users with disabilities throughout the design and evaluation process to ensure that their needs and preferences are adequately addressed. The study highlighted the value of user feedback and participatory design in creating accessible and usable web experiences.

Bigham et al. [18] developed a system called WebInSight, which leverages crowdsourcing and computer vision techniques to provide alt text descriptions for images on websites. The system relies on human workers to generate high-quality alt text, which is then used to train machine learning models for automated alt text generation. This approach combines the strengths of human expertise and machine learning to enhance the accessibility of visual content on the web.

2.5 Real-time Accessibility Adaptation

While the aforementioned approaches have made significant contributions to web accessibility, there is still a need for real-time, automated solutions that can optimize accessibility on-the-fly. Real-time accessibility adaptation involves dynamically adjusting the presentation and behavior of web content based on the user's specific accessibility needs and preferences.

Garrido et al. [19] proposed a framework for real-time adaptation of web pages to improve accessibility. Their approach involved analyzing the structure and content of web pages and applying predefined adaptation rules to modify the presentation and behavior of the page. The framework demonstrated the feasibility of real-time accessibility adaptation and its potential benefits for users with disabilities.

Fernandes et al. [20] developed a browser extension called AccessiBrowser, which provides personalized accessibility adaptations based on user preferences. The extension allows users to customize various aspects of the web page, such as font size, color contrast, and keyboard navigation, to suit their individual needs. The study highlighted the importance of user control and personalization in real-time accessibility adaptation.

My proposed system builds upon the existing research and advances the state-of-the-art by leveraging on-device machine learning and automated adaptation to provide real-time accessibility optimization. By bringing the intelligence and processing capabilities directly to the user's device, my approach aims to offer a seamless and personalized accessibility experience, tailored to the specific needs and preferences of each individual user.

III. PROPOSED SOLUTION

The proposed real-time accessibility optimization system consists of a browser extension that integrates on-device machine learning models for accessibility analysis and adaptation. The browser extension operates on the client-side, directly interacting with the web pages loaded in the user's browser. Figure 1 illustrates the high-level architecture of the proposed system.

The key components of the system architecture are as follows:

3.1 Accessibility Analysis Module
The accessibility analysis module is responsible for analyzing the structure, content, and visual aspects of web pages to identify potential accessibility issues. It comprises a set of machine learning models trained to detect specific accessibility barriers, such as missing alt text, low contrast text, improper heading structures, and keyboard navigation issues. The module employs various techniques, including computer vision, natural language processing, and pattern recognition, to extract relevant features from the web page. These features are then fed into the trained machine learning models, which classify and prioritize the identified accessibility issues based on their severity and impact on user experience.

```javascript
// Pseudo-code for generating accessibility adaptations

function generateAdaptations(accessibilityIssues, userPreferences) {
  const adaptations = [];

  // Generate adaptations based on the identified accessibility issues
  for (const issue of accessibilityIssues) {
    const adaptationRules = getAdaptationRules(issue, userPreferences);
    const generatedAdaptations = applyAdaptationRules(issue, adaptationRules);
    adaptations.push(...generatedAdaptations);
  }

  return adaptations;
}

function applyAdaptationRules(issue, adaptationRules) {
  const adaptations = [];

  // Apply each adaptation rule to the web page
  for (const rule of adaptationRules) {
    const elements = selectElements(issue.selector);
    const adaptedElements = applyRule(elements, rule);
    adaptations.push(adaptedElements);
  }

  return adaptations;
}
```

### 3.2 Adaptation Engine

The adaptation engine takes the output of the accessibility analysis module and generates appropriate adaptations to enhance the accessibility of the web page. It consists of a rule-based system that maps the identified accessibility issues to specific adaptation strategies.

The adaptation strategies encompass a range of techniques, such as modifying the visual presentation (e.g., adjusting font sizes, colors, and contrast), restructuring the page layout (e.g., reordering content, adding skip navigation links), and enhancing the interaction mechanisms (e.g., providing keyboard shortcuts, improving focus management).

The adaptation engine applies these strategies dynamically to the web page, modifying the HTML, CSS, and JavaScript code in real-time. The adaptations are tailored to the user's specific accessibility needs and preferences, which can be configured through the browser extension's user interface. The adaptation engine also incorporates a conflict resolution mechanism to handle potential conflicts between multiple adaptation strategies. For example, if one adaptation suggests increasing the font size for better readability while another adaptation proposes adjusting the color contrast, the conflict resolution mechanism determines the optimal combination of adaptations based on predefined priority rules and user preferences. This ensures that the generated adaptations are consistent, compatible, and aligned with the user's needs. Additionally, the adaptation engine includes a fallback mechanism to gracefully handle cases where certain adaptations cannot be applied due to technical limitations or potential disruptions to the web page's functionality. By considering these aspects, the adaptation engine generates reliable and effective adaptations that enhance the accessibility of the web page while maintaining its usability and integrity.
On-Device Machine Learning Models

The proposed system leverages on-device machine learning models to enable real-time accessibility analysis and adaptation without the need for server-side processing. The machine learning models are deployed directly within the browser extension, allowing for fast and efficient inference on the client-side.

The on-device models are trained offline using large datasets of web pages annotated with accessibility labels. Transfer learning techniques are employed to fine-tune pre-trained models, such as convolutional neural networks (CNNs) for image-based tasks and transformer-based models for text-based tasks, to the specific domain of web accessibility.

The use of on-device machine learning offers several advantages, including reduced latency, improved privacy, and the ability to operate offline. By performing the accessibility analysis and adaptation locally on the user's device, the system ensures a seamless and responsive user experience, without the need for constant communication with external servers.

User Interface and Personalization

The browser extension provides a user-friendly interface that allows users to configure their accessibility preferences and interact with the system. The user interface enables users to customize various aspects of the accessibility adaptations, such as the desired font size, color scheme, and keyboard navigation settings. Moreover, the system incorporates a personalization component that learns from user interactions and feedback over time. As users interact with the adapted web pages and provide explicit or implicit feedback (e.g., through user settings, browsing behavior), the system refines its adaptation strategies to better align with the user's individual needs and preferences.

The personalization component employs machine learning techniques, such as reinforcement learning and collaborative filtering, to continuously improve the accuracy and relevance of the accessibility adaptations. This ensures that the system evolves and adapts to the changing needs and preferences of each individual user.

---

3.3 On-Device Machine Learning Models

```javascript
// Pseudo-code for generating accessibility adaptations
function generateAdaptations(accessibilityIssues, userPreferences) {
const adaptations = [];

// Generate adaptations based on the identified accessibility issues
for (const issue of accessibilityIssues) {
const adaptationRules = getAdaptationRules(issue, userPreferences);
const generatedAdaptations = applyAdaptationRules(issue, adaptationRules);
adaptations.push(…generatedAdaptations);
}

return adaptations;
}

function applyAdaptationRules(issue, adaptationRules) {
const adaptations = [];

// Apply each adaptation rule to the web page
for (const rule of adaptationRules) {
const elements = selectElements(issue.selector);
const adaptedElements = applyRule(elements, rule);
adaptations.push(adaptedElements);
}

return adaptations;
}
```

3.3 User Interface and Personalization

```javascript
// Pseudo-code for loading and using on-device machine learning models
async function loadModel(modelPath) {
const model = await tf.loadLayersModel(modelPath);
return model;
}

async function predictAccessibilityIssues(model, features) {
const predictions = await model.predict(features);
const accessibilityIssues = extractIssues(predictions);
return accessibilityIssues;
}

// Example usage
const altTextModel = await loadModel('models/alt_text_model.json');
const altTextIssues = await predictAccessibilityIssues(altTextModel, visualFeatures);
```
IV. ACCESIBILITY ANALYSIS AND ADAPTATION PROCESS

The accessibility analysis and adaptation process is a key component of the proposed system, responsible for identifying accessibility issues in web pages and generating appropriate adaptations to enhance the user experience. Figure 2 illustrates the workflow of the accessibility analysis and adaptation process.

4.1 Web Page Parsing and Feature Extraction

The process begins with the parsing of the web page loaded in the user’s browser. The system traverses the Document Object Model (DOM) tree of the page, extracting relevant features for accessibility analysis. These features include:

1. Textual content: The system extracts the text from various elements, such as headings, paragraphs, links, and buttons, to analyze the linguistic characteristics and semantic structure of the page.
2. Visual attributes: The system captures visual properties, such as font sizes, colors, contrast ratios, and layout information, to assess the visual accessibility of the page.
3. Structural information: The system analyzes the hierarchical structure of the page, including the use of semantic HTML elements (e.g., <nav>, <header>, <main>), to evaluate the logical organization and navigability of the content.

The extracted features are preprocessed and transformed into suitable representations for input to the machine learning models.

4.2 Machine Learning-based Accessibility Analysis

The preprocessed features are fed into a suite of machine learning models trained to detect specific accessibility issues. Each model focuses on a particular aspect of accessibility, such as:

1. Alt text analysis: A model trained to identify missing or inadequate alt text descriptions for images and other non-text content.
2. Color contrast assessment: A model that evaluates the color contrast ratios of text and background elements to ensure sufficient visibility for users with visual impairments.
3. Heading structure analysis: A model that examines the use and hierarchy of heading elements (<h1> to <h6>) to assess the logical structure and navigability of the page.
4. Keyboard accessibility evaluation: A model that checks for the presence and functionality of keyboard navigation mechanisms, such as proper focus management and keyboard shortcuts.

The machine learning models employ various techniques, such as deep learning, decision trees, and support vector machines, to classify and prioritize the identified accessibility issues based on their severity and impact on user experience.

4.3 Adaptation Generation

Based on the output of the accessibility analysis models, the adaptation engine generates appropriate adaptations to enhance the accessibility of the web page. The adaptation generation process involves the following steps:

1. Issue prioritization: The identified accessibility issues are prioritized based on their severity and the user’s specific needs and preferences. For example, issues related to keyboard accessibility may be given higher priority for users who rely on keyboard navigation.
2. Adaptation rule selection: The adaptation engine selects relevant adaptation rules from its knowledge base based on the prioritized issues. These rules define specific modifications to be applied to the web page to address the identified accessibility barriers.
3. Dynamic page modification: The selected adaptation rules are applied dynamically to the web page, modifying the HTML, CSS, and JavaScript code in real-time. The modifications may include adjusting font sizes, colors, and contrast ratios, restructuring the page layout, adding skip navigation links, or enhancing keyboard navigation functionality.
4. User feedback incorporation: The system incorporates user feedback and preferences into the adaptation generation process. If the user has previously provided explicit feedback or customized settings, these are taken into account while generating the adaptations.

The generated adaptations are seamlessly applied to the web page, providing an enhanced and personalized accessibility experience for the user.

![Accessibility Analysis and Adaptation Process](image)

**4.4 Continuous Learning and Refinement**

The proposed system incorporates a continuous learning and refinement mechanism to improve the accuracy and effectiveness of the accessibility analysis and adaptation process over time. As users interact with the adapted web pages and provide feedback, the system collects and analyzes this data to update and refine its machine learning models and adaptation strategies.

The collected user feedback includes explicit ratings, preferences, and settings provided through the browser extension’s user interface, as well as implicit feedback inferred from user interactions and browsing behavior. This feedback is used to retrain and fine-tune the machine learning models, improving their ability to identify accessibility issues and generate more accurate and relevant adaptations.

Moreover, the system employs online learning techniques, such as incremental learning and active learning, to continuously update its knowledge base of adaptation rules based on real-world usage patterns and emerging accessibility best practices. This ensures that the system remains up-to-date and responsive to the evolving needs and expectations of users with disabilities.

**V. EVALUATION FRAMEWORK AND EXPECTED RESULTS**

To assess the effectiveness and usability of the proposed real-time accessibility optimization system, I have designed a comprehensive evaluation framework. The evaluation will involve both quantitative and qualitative methods, including user studies, performance metrics, and comparative analyses.

**5.1 User Studies**

User studies should include individuals with diverse disabilities, including visual, auditory, motor, and cognitive impairments. The user studies will consist of the following components:
1. Usability testing: Participants will be asked to perform a set of tasks on a range of websites using the proposed browser extension. I will measure task completion rates, time on task, and user satisfaction using standardized questionnaires such as the System Usability Scale (SUS) [21] and the Web Accessibility Barrier (WAB) questionnaire [22].

2. Accessibility evaluation: Accessibility experts will assess the adapted web pages generated by the system using established accessibility evaluation methods, such as manual audits and automated testing tools. They will evaluate the system's ability to identify and address accessibility issues effectively.

3. User feedback and interviews: I will gather qualitative feedback from participants through semi-structured interviews and open-ended questionnaires. This feedback will provide insights into the users' experiences, preferences, and suggestions for improvement.

The user studies will involve a diverse sample of participants, ensuring representation from different age groups, genders, and levels of web experience. The studies will be conducted in controlled laboratory settings as well as remote sessions to assess the system's performance in various real-world scenarios.

5.2 Performance Metrics

In addition to user studies, we should evaluate the system's performance using objective metrics. These metrics will measure the accuracy, efficiency, and scalability of the accessibility analysis and adaptation process. Key performance metrics include:

1. Accessibility issue detection accuracy: I will assess the precision and recall of the machine learning models in detecting accessibility issues across different categories, such as alt text, color contrast, and keyboard accessibility. Ground truth data will be established through manual annotations by accessibility experts.

2. Adaptation quality: I will evaluate the quality and appropriateness of the generated adaptations using metrics such as the Web Content Accessibility Guidelines (WCAG) conformance level [23] and the Accessible User Experience (AUX) metric [24]. These metrics will assess the system's ability to produce adaptations that enhance accessibility while preserving the original content and functionality.

3. Performance overhead: I will measure the computational overhead introduced by the on-device machine learning models and the adaptation engine. Metrics such as processing time, memory usage, and battery consumption will be monitored to ensure that the system operates efficiently without degrading the overall user experience.

4. Personalization effectiveness: I will evaluate the effectiveness of the personalization component by measuring the improvement in user satisfaction and task completion rates over time as the system adapts to individual user preferences and behaviors.

5.3 Comparative Analysis

To demonstrate the advantages of the proposed system over existing approaches, I will conduct comparative analyses with state-of-the-art accessibility tools and frameworks. This will involve benchmarking the system's performance against popular automated accessibility evaluation tools and manually adapted web pages. I will compare the system's ability to identify and address accessibility issues, the quality and relevance of the generated adaptations, and the overall user experience. The comparative analysis will highlight the benefits of real-time optimization and on-device machine learning in enhancing web accessibility.

5.3 Expected Results Based on the proposed evaluation framework

I expect the following results:

1. Improved accessibility: The system will significantly improve the accessibility of web pages for users with disabilities, as demonstrated by higher task completion rates, reduced time on task, and increased user satisfaction in the user studies.

2. High accuracy and efficiency: The machine learning models will achieve high accuracy in detecting accessibility issues, while the adaptation engine will generate high-quality adaptations efficiently, without introducing significant performance overhead.

3. Personalization benefits: The personalization component will enhance the user experience over time, as evidenced by increased user satisfaction and task completion rates as the system adapts to individual user preferences and behaviors.

4. Comparative advantages: The proposed system will outperform existing accessibility tools and frameworks in terms of accuracy, efficiency, and user experience, highlighting the benefits of real-time optimization and on-device machine learning.

The evaluation results will provide valuable insights into the effectiveness and usability of the proposed real-time accessibility optimization system. The findings will inform further refinements and improvements to the system, contributing to the advancement of web accessibility research and practice.
VI. CONCLUSION

I have proposed a novel approach to enhancing web accessibility through real-time optimization using on-device machine learning and automated adaptation. The proposed system leverages the power of machine learning models deployed directly within a browser extension to analyze web pages in real-time, identify accessibility issues, and generate appropriate adaptations to improve the user experience for individuals with disabilities.

The proposed system architecture, consisting of the accessibility analysis module, adaptation engine, on-device machine learning models, and user interface, enables a seamless and personalized accessibility experience. By performing the accessibility analysis and adaptation locally on the user's device, the system ensures fast and efficient processing while preserving user privacy. The evaluation framework, encompassing user studies, performance metrics, and comparative analyses, will provide a comprehensive assessment of the system's effectiveness and usabability. The expected results highlight the potential of the proposed approach in significantly improving web accessibility, offering high accuracy and efficiency, and delivering personalized user experiences.

However, there are several challenges and limitations that need to be addressed in future work. These include:

1. Scalability: Ensuring that the system can handle a wide range of websites and accommodate the growing complexity and diversity of web technologies.
2. User adoption: Exploring strategies to promote user adoption of the browser extension and gather a large-scale dataset of user interactions and feedback for continuous improvement of the machine learning models.
3. Integration with assistive technologies: Investigating ways to seamlessly integrate the proposed system with existing assistive technologies, such as screen readers and voice recognition tools, to provide a comprehensive accessibility solution.
4. Collaborative accessibility: Exploring mechanisms for users to share their personalized adaptations and contribute to a global knowledge base of accessibility best practices, fostering a collaborative approach to web accessibility.
5. Ethical considerations: Addressing potential ethical concerns related to user privacy, data security, and the transparent communication of the system’s capabilities and limitations to users.

Future research will focus on addressing these challenges and expanding the scope of the proposed system. This may involve exploring advanced machine learning techniques, such as federated learning and transfer learning, to enhance the scalability and adaptability of the system. Collaborations with accessibility communities, web developers, and policymakers will be crucial in driving the adoption and standardization of real-time accessibility optimization approaches.

In conclusion, the proposed real-time accessibility optimization system represents a significant step forward in enhancing web accessibility through the application of on-device machine learning and automated adaptation. By empowering users with disabilities to access and interact with web content more effectively, this research contributes to the creation of a more inclusive and equitable digital society. The insights gained from this work will pave the way for future innovations in web accessibility, ultimately benefitting millions of individuals worldwide.

References


